Stability Regularized Cross-Validation

Ryan Cory-Wright
Department of Analytics, Marketing and Operations
Imperial College Business School
r.cory-wright@imperial.ac.uk

Andrés Gómez
Department of Industrial and Systems Engineering
Viterbi School of Engineering, University of Southern California, CA
gomezand@usc.edu

Abstract

We revisit the problem of ensuring strong test-set performance via cross-validation. Motivated by the generalization theory literature, we propose a nested k-fold cross-validation scheme that selects hyperparameters by minimizing a weighted sum of the usual cross-validation metric and an empirical model-stability measure. The weight on the stability term is itself chosen via a nested cross-validation procedure. This reduces the risk of strong validation set performance and poor test set performance due to instability. We benchmark our procedure on a suite of 13 real-world UCI datasets, and find that, compared to k-fold cross-validation over the same hyperparameters, it improves the out-of-sample MSE for sparse ridge regression and CART by 4% on average, but has no impact on XGBoost. This suggests that for interpretable and unstable models, such as sparse regression and CART, our approach is a viable and computationally affordable method for improving test-set performance.

1 Introduction

A central problem in machine learning and data-driven optimization involves constructing models that reliably generalize well to unseen data. One of the most popular approaches is cross-validation as introduced by [53, 18], which selects hyperparameters that perform well on a cross-validation set as a proxy for strong test-set performance. Indeed, this model selection pipeline is advocated by most machine learning textbooks [e.g. 28, 25]. Moreover, in the statistical learning literature, there is a broad set of conditions under which the cross-validation loss (possibly with mild corrections) is a good estimator of the test-set error [41, 32, 29, 40], especially when the sample size is large [34, 11].

At the same time, both theory [e.g. 48, 22] and experiments [e.g. 45, 44, 42, 52, 3] have documented an *adaptivity gap* between validation and test-set performance (although not always observed [46, 3]), especially in settings with limited data. Concretely, a positive adaptivity gap arises when the validation error is systematically lower than the test set error of a machine learning model.

One explanation for adaptivity gaps is as follows: validation sets give approximately unbiased estimators of test set performance for a *fixed* combination of hyperparameters. However, validation scores are random variables and subject to some variance. Therefore, the act of *optimizing* the validation set error risks selecting hyperparameter combinations that disappoint out of sample. In the extreme case with many hyperparameters relative to the number of samples, the act of optimizing the (cross) validation error can be viewed as training on the (cross) validation set. This phenomenon is well documented in different parts of the statistics and optimization literature, where it is variously

Preprint. Under review.

called "post-decision surprise", "out-of-sample disappointment", "researcher degree of freedom" or "the optimizer's curse" [27, 38, 39, 51, 50, 56]. This raises the question: *is it possible to improve the performance of (cross) validation by combating the adaptivity gap*, possibly by selecting models with a higher validation error and a lower score according to a second metric.

In this work, we propose a strategy for mitigating out-of-sample disappointment and show that it sometimes improves cross-validation's performance. Specifically, we propose selecting models according to a weighted sum of their cross-validation error and their empirical *hypothesis stability* (see Section 2.2). This is motivated by the observation that both the cross-validation error and the hypothesis stability appear in generalization bounds on the test-set error; thus, minimizing the cross-validation error alone may be vulnerable to selecting high-variance models that perform poorly out-of-sample.

Our main contributions are threefold. First, we extend a generalization bound on the test set error due to [11] from leave-one-out to k-fold cross-validation. This generalization bound takes the form of the cross-validation error plus a term related to a model's hypothesis stability. Second, motivated by this (often conservative) bound, we propose *regularizing* cross-validation by selecting models that minimize a weighted sum of a validation metric and the hypothesis stability, rather than the validation score alone, to mitigate out-of-sample disappointment without being overly conservative. Indeed, models with a low cross-validation error *that are stable* generalize better than models with a low cross-validation error that are unstable. Moreover, to select the weight in this scheme, we embed the entire scheme within a nested cross-validation procedure. Finally, we empirically investigate our proposal using sparse ridge regression, CART, and XGBoost models, and find that it improves the out-of-sample performance of sparse ridge regression and CART by 4% on average but has no impact on XGBoost, likely because XGBoost is a more stable learner.

1.1 Motivating Example: Poor Performance of Cross-Validation for Sparse Linear Regression

We illustrate the pitfalls of cross-validation in a sparse ridge regression setting, as studied by [7, 31, 35]. Suppose we wish to recover a $\tau_{\text{true}} = 5$ sparse regressor which is generated from a stochastic process according to the following setup [cf. 8]: we fix the number of features p, number of datapoints n, correlation parameter $\rho = 0.3$ and signal to noise parameter $\nu = 1$, and generate $\boldsymbol{X}, \boldsymbol{y}$ according to a data generation procedure standard to the literature and stated in Appendix A for brevity.

Following the standard cross-validation paradigm, we then evaluate the cross-validation error for each τ and 20 values of γ log-uniformly distributed on $[10^{-3}, 10^3]$, using the Generalized Benders Decomposition scheme developed by [7] to solve each MIO to optimality, which is of the form

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \quad \frac{\gamma}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \text{ s.t. } \|\boldsymbol{\beta}\|_0 \leq \tau, \tag{1}$$

and selecting the hyperparameter combination with the lowest cross-validation error, for both leave-one-out and five-fold cross-validation.

Figure 1 depicts each hyperparameter combination's leave-one-out (left) and test (right) error, in an overdetermined setting where $n = 50, p = 10$ (top) and an underdetermined setting where $n = 10, p = 50$ (bottom). We generate equivalent plots for five-fold cross-validation in Figure 2 (Appendix B) and obtain similar results. In the overdetermined setting, cross-validation performs well: a model trained by minimizing the LOOCV (resp. five-fold) cross-validation error attains a test error within 0.6% (resp. 1.1%) of the (unknowable) test minimum. However, in the underdetermined setting, cross-validation performs poorly: a model trained by minimizing the LOOCV (resp. five-fold) error attains a test set error 16.4% (resp. 31.7%) larger than the test set minimum and is seven orders of magnitude larger (resp. one order of magnitude larger) than its LOOCV estimator. This highlights the danger of optimizing the cross-validation error alone, especially in underdetermined settings.

1.2 Literature Review

From a statistical learning perspective, there is significant literature on quantifying the out-of-sample performance of models with respect to their training and validation error, originating with the works by [57] on VC-dimension and [11] on algorithmic stability theory. As noted, for instance, by [1], algorithmic stability bounds are generally preferable: they are *a posteriori* bounds with tight constants that depend on only the problem data. In contrast, VC-dimension bounds are *a priori* bounds that

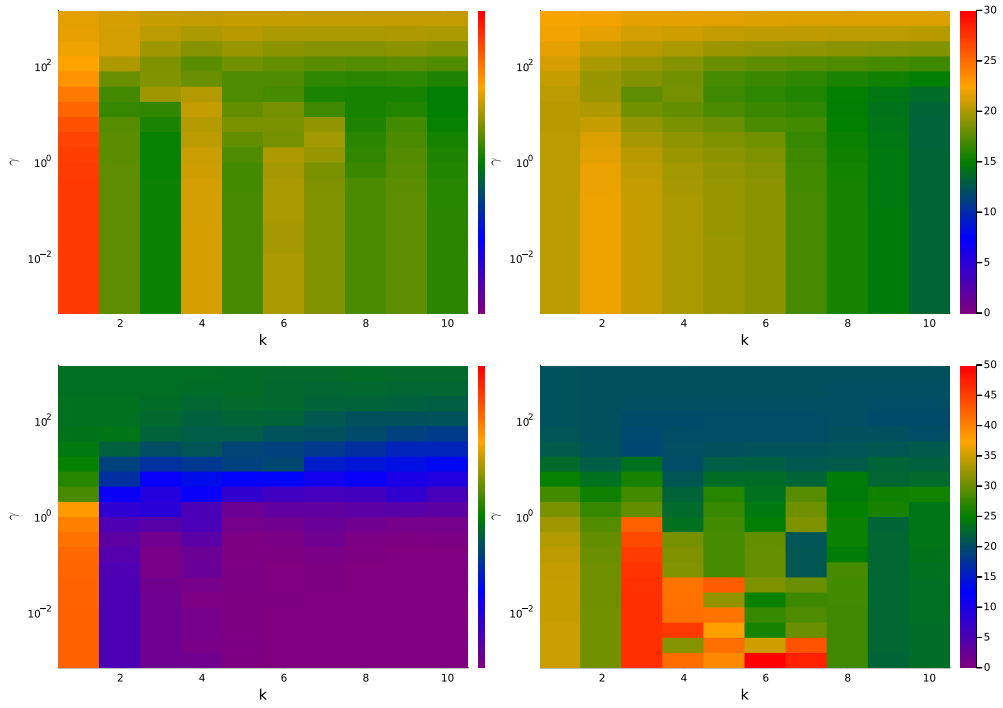

Figure 1: Leave-one-out (LOOCV, left) and test (right) error for varying τ and γ, for an overdetermined setting (top, $n = 50, p = 10$) and an underdetermined setting (bottom, $n = 10, p = 50$). In the overdetermined setting, LOOCV is a good estimate of the test error for most values of parameters (γ, τ). In contrast, in the underdetermined setting, LOOCV is a poor approximation of the test error, and estimators that minimize LOOCV ($\gamma \to 0$, $\tau = 10$) significantly disappoint out-of-sample. Our conclusions are identical when using five-fold cross-validation (Appendix B).

depend on computationally intractable constants like Rademacher averages. A key conclusion from both streams of work is that simpler and more stable models, as well as models obtained using less computational time to cross-validate, tend to disappoint less out-of-sample. We refer to [26, 52, 3] for more extensive reviews of cross-validation and algorithmic stability.

More recently, the statistical learning theory literature has been connected to the distributionally robust optimization literature by [2, 1, 22, 21, 23] among others. Indeed, [1] proposes solving newsvendor problems by designing decision rules that map features to an order quantity and obtain finite-sample guarantees on out-of-sample costs of newsvendor policies in terms of in-sample cost.

Even closer to our work, [22] proposes correcting solutions to high-dimensional problems by invoking Stein's lemma to obtain a Stein's Unbiased Risk Estimator (SURE) approximation of the out-of-sample disappointment and demonstrates that minimizing their bias-corrected training objective generates models that outperform sample-average approximation models out-of-sample. Moreover, they demonstrate that a naive implementation of leave-one-out cross-validation performs poorly in settings with limited data. Building upon this work, [23] proposes debiasing a model's in-sample performance by incorporating a variance gradient correction term derived via sensitivity analysis. Unfortunately, it is unclear how to extend their approach to the setting considered here, as it applies to problems with linear objectives over subsets of $[0, 1]^n$.

2 Stability Adjusted Cross-Validation

In this section, we propose techniques for improving the performance of cross-validation. First, we define our notation (Section 2.1) and propose a bound on the test set error of a machine learning model in terms of its k-fold error and algorithmic stability, which was originally proven for the special case of leave-one-out cross-validation by [11] (Section 2.2). By leveraging this bound, we propose a

technique for improving the performance of cross-validation, namely nested cross-validation with stability regularization (Section 2.3)

2.1 Setup and Notation

We consider a generic k-fold cross-validation setting for supervised learning problems with n datapoints, and our notation is standard to the machine learning literature. Concretely, we have features $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathcal{X}^n \subseteq \mathbb{R}^{p \times n}$ and response $(y_1, \ldots, y_n) \in \mathcal{Y}^n \subseteq \mathbb{R}^n$, and we assume that the data $(\boldsymbol{x}_i, y_i)_{i \in [n]} \in \mathcal{X} \times \mathcal{Y}$ are drawn i.i.d. from some (unknown) stochastic process. We aim to understand how abstract models $\boldsymbol{\beta} : \mathcal{X} \to \mathcal{Y}$ trained on the full dataset $(\boldsymbol{X}, \boldsymbol{y})$ generalize to other observations from the same stochastic process, by studying $\boldsymbol{\beta}$ and related models $\boldsymbol{\beta}^{(\mathcal{N}_j)} : \mathcal{X} \to \mathcal{Y}$ trained on all data apart from the points $i \in \mathcal{N}_j$. We formalize the notion of generalization with a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, e.g., $\ell(\hat{y}, y) = (\hat{y} - y)^2$. We let $\{\mathcal{N}_j\}_{j \in [k]}$ be a partition of the integers $[n]$ into k disjoint subsets, where k is frequently either $5, 10$ (five-fold and ten-fold cross-validation) or n (leave-one-out cross-validation), and the cardinality of each fold $\mathcal{N}_j$ is typically identical.

To make the dependence of our abstract models $\boldsymbol{\beta}$ on hyperparameters explicit, we let $\boldsymbol{\theta} \in \Theta$ be the vector of all hyperparameters that are used to select $\boldsymbol{\beta}$, and Θ denote the set of all possible hyperparameter combinations. We define an abstract model trained using hyperparameters $\boldsymbol{\theta}$ by $\boldsymbol{\beta}(\boldsymbol{\theta})$ for concreteness and assume that this choice is unique for simplicity[1]. Finally, at the risk of overloading notation, we let, $\boldsymbol{\beta}(\boldsymbol{\theta}, \boldsymbol{x}_i) \in \mathcal{Y}$ denote the output predicted by model $\boldsymbol{\beta}(\boldsymbol{\theta})$ with data $\boldsymbol{x}_i$.

Given the above notation, the k-fold cross-validation error with hyperparameters $\boldsymbol{\theta}$ is given by[2]:

$$h(\boldsymbol{\theta}) = \frac{1}{n} \sum_{j=1}^{k} \sum_{i \in \mathcal{N}_j} \ell\left(y_i, \beta^{(\mathcal{N}_j)}(\boldsymbol{\theta}, \boldsymbol{x}_i)\right) \tag{2}$$

Moreover, for each $j \in [k]$, we let the jth partial k-fold error be:

$$h_j(\boldsymbol{\theta}) := \sum_{i \in \mathcal{N}_j} \ell\left(y_i, \beta^{(\mathcal{N}_j)}(\boldsymbol{\theta}, \boldsymbol{x}_i)\right). \tag{3}$$

Therefore, the average k-fold error is given by $1/n \sum_{j=1}^{k} h_j(\boldsymbol{\theta}) = h(\boldsymbol{\theta})$.

We now define the stability of our models, following [11]. Specifically, let μ_h be the hypothesis stability of our learner analogously to [11, Definition 3] but where $k < n$ folds are possible:

$$\mu_h := \max_{j \in [k]} \mathbb{E}_{\boldsymbol{x}_i, y_i} \left| \ell\left(y_i, \beta^{(\mathcal{N}_j)}(\boldsymbol{\theta}, \boldsymbol{x}_i)\right) - \ell\left(y_i, \beta(\boldsymbol{\theta}, \boldsymbol{x}_i)\right) \right|, \tag{4}$$

where the expectation is taken over all $(\boldsymbol{x}_i, y_i)$ drawn i.i.d. from the underlying stochastic process that generated the training set. This quantity measures the worst-case average absolute change in the loss after omitting a fold of data.

Unfortunately, the hypothesis stability involves computing a possibly high-dimensional expectation and thus is #P-hard to compute in general (even when $\boldsymbol{\beta}(\boldsymbol{\theta})$ is computable in polynomial time) by reduction from two-stage stochastic programming with random recourse [24, 6]. Moreover, the stochastic process which $(\boldsymbol{x}_i, y_i)$ is drawn from is often unknown in practice, and thus cannot be computed even with the help of Monte-Carlo simulation or similar techniques to evaluate high-dimensional integrals. Indeed, we have the following result:

Proposition 1. *Determining the quantity μ_h in (4) is #P-hard, i.e., at least as hard as computing the number of optimal solutions to an NP-complete problem, even for the simple case of a binary loss function and i.i.d. discrete random data $(\boldsymbol{x}_i, y_i)$.*

Proof. We perform a reduction from Bertsimas and Sturt [6, Corollary 2.1], where the authors report that if $X_1, \ldots, X_n$ are i.i.d. discrete random variables with support containing at least n distinct

[1] Relaxing this assumption leads to a notion of Pareto Cross-Validation analogous to Pareto-optimal Benders cuts as described by [37] and Pareto robust optimization as described by [33]. We leave this for future work.

[2] We could also consider average over all $\binom{n}{k}$ folds of size k for accuracy. We leave this for future work.

values, then computing $\mathbb{P}(\sum_i X_i \leq \alpha)$ is #P-hard. Specifically, let the problem data and lower level training problem be such that

$$\ell(y_i, \boldsymbol{\beta}^{(\mathcal{N}_j)}(\boldsymbol{\theta}, \boldsymbol{x}_i)) = 1\left\{\sum_{i\in[n]\setminus\mathcal{N}_j}\sum_{l\in[p]} x_{i,j} \leq \alpha\right\}$$

and

$$\ell(y_i, \boldsymbol{\beta}(\boldsymbol{\theta}, \boldsymbol{x}_i)) = 1\left\{\sum_{i\in[n]}\sum_{l\in[p]} x_{i,j} \leq \alpha\right\}.$$

Thus, if μ_h were not #P-hard then $\mathbb{P}(\sum_{i=1}^n X_i \leq \alpha)$ would also not be #P-hard, a contradiction. □

Thus, to avoid the cost of computing the potentially unknowable and #P-hard quantity μ_h, in our computations we approximate (4) via the empirical hypothesis stability analogously to [11, Definition 3] but where $k < n$ folds are possible:

$$\hat{\mu}_h := \max_{j\in[k]} \frac{1}{n}\sum_{i=1}^{n}\left|\ell\left(y_i, \beta^{(\mathcal{N}_j)}(\boldsymbol{\theta}, \boldsymbol{x}_i)\right) - \ell\left(y_i, \beta(\boldsymbol{\theta}, \boldsymbol{x}_i)\right)\right|. \tag{5}$$

This can be viewed as a sample-average approximation of the hypothesis stability [see 49, for a general theory], and thus it converges almost surely to the true hypothesis stability as $n \to \infty$ under the usual assumptions of the sample-average-approximation method [49].

We remark that the above empirical hypothesis stability differs from the pointwise hypothesis stability defined by [11, Definition 4] as we average over all data points, rather than only data points omitted when training $\boldsymbol{\beta}^{(\mathcal{N}_j)}$. This is because the pointwise stability only gives generalization bounds on the training error [11], while we are interested in bounds on the kCV error.

2.2 Generalization Bound

By combining our definitions and notation, letting M represent an upper bound on the loss $\ell(\boldsymbol{\beta}(\boldsymbol{\theta}, \boldsymbol{x}_i), y_i)$ for any model $\boldsymbol{\beta}(\boldsymbol{\theta})$ and any datapoint $(\boldsymbol{x}_i, y_i)$ (e.g., if $(\boldsymbol{x}_i, y_i)$ are drawn from a bounded domain), and letting $\mathcal{S}$ denote a test set of observations drawn from the same distribution as our training set (but not seen by the model), the following result follows from Chebyshev's inequality (proof deferred to Section B.1):

Theorem 1. *Suppose the training data $(\boldsymbol{x}_i, y_i)_{i\in[n]}$ are drawn from an unknown distribution $\mathcal{D}$ such that M and μ_h are finite constants. Further, suppose n is exactly divisible by k and each $\mathcal{N}_j$ is of cardinality n/k. Then, the following bound on the test error holds with probability at least $1 - \Omega$:*

$$\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \ell\left(y_i, \beta(\boldsymbol{\theta}, \boldsymbol{x}_i)\right) \leq \frac{1}{n}\sum_{j\in[k]} h_j(\boldsymbol{\theta}) + \sqrt{\frac{M^2 + 6Mk\mu_h}{2k\Omega}}. \tag{6}$$

Remark 1 (To Train or to Validate in (6))**.** *A similar bound to* (6) *can be derived using the empirical risk instead of kCV [11, Theorem 11]. However, this bound has a larger constant (*12 *instead of* 6*) and still involves the expensive pointwise hypothesis stability defined by [11].*

Theorem 1 reveals that, if the number of folds k increases with n, M is finite, and the hypothesis stability μ_h decreases with n, then the kCV error generalizes to the test set with high probability as n becomes large. Moreover, when models have the same cross-validation error, hypothesis stability, and loss bound M, training on more folds results in a stronger generalization bound.

We remark that Theorem 1 implicitly justifies the use of regularization in machine learning, because regularization implicitly controls the hypothesis stability μ_h, leading to better generalization properties when μ_h is lower. Indeed, Bousquet and Elisseeff [11, Theorem 22] provides a result formalizing this notion in the context of supervised learning for Reproducing Kernel Hilbert Spaces.

Unfortunately, in preliminary experiments, we found that Equation (6)'s bound is often excessively conservative in practice, especially when $n \gg p$. This conservatism stems from using Chebyshev's inequality in the proof of Theorem 1, which is known to be tight for discrete measures but excessively

conservative over continuous measures [5], especially unimodal continuous measures [55]. Thus, motivated by the robust optimization literature, where probabilistic guarantees are used to motivate uncertainty sets but less stringent guarantees are used in practice to avoid excessive conservatism [see 20, Section 3, for a discussion], we leverage Theorem 1 to propose a new approach to cross-validation in the rest of this section.

2.3 Stability Regularization and Nested Cross-Validation

Motivated by the idea that more stable models are less likely to disappoint out-of-sample (Theorem 1), we propose selecting models that minimize a weighted sum of the kCV error and the hypothesis stability. This corresponds to selecting $\boldsymbol{\theta}$ through the optimization problem

$$\boldsymbol{\theta} \in \operatorname*{arg\,min}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{1}{n} \sum_{j \in [k]} h_j(\boldsymbol{\theta}) + \lambda \mu_h(\boldsymbol{\theta}). \tag{7}$$

Notably, this procedure still requires a hyperparameter λ, which the user must select. Accordingly, we invoke nested cross-validation, which has been empirically shown to be significantly less vulnerable to out-of-sample disappointment than regular cross-validation [13, 3]. We perform an outer loop over candidate λ values. For each λ, we run an inner (nested) k-fold cross-validation: on each fold, train a model with that λ (using the remaining folds with stability regularization) and measure its validation error. We then choose the λ that yields the lowest average inner validation error across the k folds.

Specifically, let $\mathcal{N}_{j,l} := \mathcal{N}_j \cup \mathcal{N}_l$ denote the data contained in the jth or lth fold of the training set. Then, we select λ by solving

$$\lambda \in \arg\min_{\lambda \in \boldsymbol{\Lambda}} \sum_{j \in [k]} \sum_{i \in \mathcal{N}_j} \ell(y_i, \boldsymbol{\beta}^{(\mathcal{N}_j)}(\boldsymbol{\theta}_j^\star, \boldsymbol{x}_j)), \tag{8}$$

where $\boldsymbol{\beta}^{(\mathcal{N}_j)}(\boldsymbol{\theta}_j^\star)$ denotes a model trained on all data but the fold $\mathcal{N}_j$ with hyperparameters $\boldsymbol{\theta}_j^\star$, and $\boldsymbol{\theta}_j^\star$ denotes an optimal solution to the following lower-level problem, which cross-validates $\boldsymbol{\theta}$ with the jth fold of the data omitted and λ fixed:

$$\boldsymbol{\theta}_j^\star \in \operatorname*{arg\,min}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{1}{n - |\mathcal{N}_j|} \sum_{l \in [k]: l \neq j} \sum_{i \in \mathcal{N}_l} \ell(y_i, \boldsymbol{\beta}^{(\mathcal{N}_{j,l})}(\boldsymbol{\theta})) + \lambda \boldsymbol{\mu}_h(\boldsymbol{\theta}),$$

where $\boldsymbol{\mu}_h(\boldsymbol{\theta})$ is calculated without reference to the jth fold of the data in this case. This corresponds to selecting λ in a manner that ensures that models $\boldsymbol{\theta}_{\boldsymbol{j}}^\star$ perform well on average, as in the standard nested cross-validation paradigm.

Finally, once λ is selected, we fix λ and select $\boldsymbol{\theta}$ by minimizing (7). This selects models that are robust to omitting one fold of the data, and tend to perform well out-of-sample as estimated by the nested cross-validation procedure, at the price of increasing the cost of hyperparameter selection. For completeness, we formalize this procedure in Algorithm 1 (see Appendix C).

We acknowledge that even nested CV errors are optimistically biased estimates (as ordinary CV errors are). However, various authors [13, 3, 4] and our experiments indicate that the nested CV error is a more accurate estimator of test performance than the standard CV error. We view our nesting approach as a practical improvement over the status quo, while leaving room for future refinements.

3 Numerical Experiments

In this section, we evaluate the numerical performance of the nested regularized cross-validation scheme proposed in Section 2. All experiments in this section were implemented in `Julia` version 1.9, using `Mosek` version 11.0 to solve all conic optimization problems, and conducted on a standard MacBook Pro laptop with a 36 GB Apple M3 CPU and 36 GB main memory.

3.1 Ridge Regularized Best Subset Selection

We now benchmark our proposed nested cross-validation scheme on sparse regression for a suite of commonly studied real-world datasets. Specifically, we benchmark a cyclic coordinate descent

scheme for ℓ_0–ℓ_2^2 sparse regression, where we repeatedly solve the following lower-level problem for different values of the sparsity parameter τ and the regularization hyperparameter γ

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p, \boldsymbol{z}\in\{0,1\}^p} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{1}{2\gamma}\sum_{j=1}^{p}\frac{\beta_j^2}{z_j} \text{ s.t. } \sum_{j=1}^{p} z_j \leq \tau,$$

using the greedy rounding algorithm described by [59, 9] to obtain near-optimal solutions in a practically tractable amount of time. In particular, we iteratively minimize the k-fold cross-validation error with respect to τ (with γ fixed) and with respect to γ (with τ fixed) until we either cycle or exceed a limit of 10 iterations. After converging, we fit a model to the full dataset with the hyperparameters $(\tau^\star, n_{\text{train}}\gamma^\star/n)$, where n_{train} is the number of observations with one fold of the data left out as in [36], to account for the extra fold when fitting a model to the full dataset. Moreover, we perform the same procedure for nested k-fold cross-validation with stability regularization. Note that for our cyclic coordinate descent schemes, we set the largest permissible value of τ such that $\tau \log \tau \leq n$, because Gamarnik and Zadik [17, Theorem 2.5] demonstrated that, up to constant terms and under certain assumptions on the data generation process, on the order of $\tau \log \tau$ observations are necessary to recover a sparse model with binary coefficients. In preliminary experiments, we relaxed this requirement to $\tau \leq p$ and found that this did not change the optimal value of τ. Moreover, we use a grid size of 20 values of γ log-uniformly distributed over $\{10^{-3}, 10^3\}$.

We compare against the following methods as a benchmark, using in-built functions to approximately minimize the cross-validation loss, and subsequently fit a regression model on the entire dataset with these cross-validated parameters. Note, however, that we are mainly interested in the performance of sparse ridge regression with and without stability regularization. For all methods, we use five folds:

- The `ElasticNet` method in the `GLMNet` package, with grid search on their parameter $\alpha \in \{0, 0.1, 0.2, \ldots, 1\}$.
- The Minimax Concave Penalty (MCP) and Smoothly Clipped Absolute Deviation Penalty (SCAD) as implemented in the `R` package `ncvreg`, using the `cv.ncvreg` function and default parameters.
- The `L0Learn.cvfit` method implemented in the `L0Learn` `R` package [cf. 30], with a grid of 10 different values of γ and default parameters otherwise.

For each dataset, we repeat the following procedure ten times to reduce the variance of our results: we randomly split the data into 90% training/validation data and 10% testing data, and report the average sparsity of the cross-validated model, the method's estimate of the MSE (kCV or nested kCV error) and the average test set MSE (using the same splits for all methods to reduce variance). We also report summary statistics in terms of the average percentage improvement of the nested procedure compared to the MSE without nesting, in order that each dataset is weighted equally. For each method, we retrain on the full training/validation dataset after cross-validation.

Table 1 depicts the dimensionality of each dataset, the average k-fold cross-validation error ("CV") or nested k-fold cross-validation error ("nCV"), the average test set error ("MSE"), and the sparsity attained by our coordinate descent scheme without any stability regularization or nesting (kCV), our cyclic coordinate descent with nested stability-adjusted cross-validation (nested-kCV), and the performance of MCP on each dataset. We also report summary statistics in terms of the average percentage improvement of the nested procedure compared to the MSE without nesting, so each dataset is weighted equally. We remark that the average (geometric mean over the average for each dataset) runtime was 14.8 seconds for sparse regression, 592 seconds for sparse regression with nesting, and under 1 second for all other methods. Table 4 (Appendix E) also shows to performance of other methods from the literature (SCAD, GLMNet, and L0Learn) in the same datasets.

We measure the improvement from nested cross-validation vs. k-fold cross-validation by computing the average MSE across each dataset, normalizing by the MSE of k-fold cross-validation on that dataset to compute a percentage improvement across each dataset, and then taking the geometric mean across all datasets (to account for the fact that percentage improvement is an asymmetric measure).

We observe that across the overdetermined datasets, nested cross-validation improves the out-of-sample MSE of sparse regression by 10.0% on average, while across the underdetermined datasets it worsens the out-of-sample MSE by 5.92% on average, for an overall improvement of 4.85% on average. Moreover, the nested kCV error is, on average, 0.9% smaller than the test set error for the

Dataset	n	p	kCV			nested-kCV			MCP		
			τ	CV	MSE	τ	nCV	MSE	τ	CV	MSE
Wine	6497	11	9.3	0.544	0.543	9.3	0.545	0.542	11	0.543	0.542
Housing	506	13	11	23.60	23.68	11	23.73	23.70	11	23.79	23.66
Auto-MPG	392	25	18.5	8.524	8.608	18	8.647	8.703	18.7	9.051	8.828
Hitters	263	19	11.7	0.076	0.082	14.7	0.080	0.080	11.5	0.077	0.082
Prostate	97	8	5.1	0.529	0.559	4.8	0.571	0.549	7.1	0.570	0.552
Servo	167	19	13.8	0.746	0.771	15.4	0.795	0.715	12	0.752	0.705
Toxicity	38	9	3.8	0.037	0.054	3.8	0.044	0.057	2.7	0.050	0.060
Steam	25	8	2.8	0.404	0.467	2.8	0.565	0.426	2.7	0.511	0.684
Alcohol2	44	21	3.6	0.210	0.472	2.7	0.266	0.229	2.1	0.232	0.273
Avg.	-	-	8.84	3.853	3.916	9.17	3.916	3.889	8.76	3.953	3.932
Std. dev	-	-	5.41	7.886	7.895	5.928	7.918	7.927	5.56	7.908	7.910
TopGear	242	373	41.3	0.037	0.050	34.9	0.055	0.062	8.1	0.057	0.066
Bardet	120	200	23.5	0.007	0.010	25.3	0.009	0.010	5.3	0.009	0.011
Vessel	180	486	28.8	0.016	0.027	29.2	0.024	0.027	2.7	0.036	0.036
Riboflavin	71	4088	13.4	0.163	0.299	13.8	0.269	0.297	7.5	0.319	0.229
Avg.	-	-	26.75	0.056	0.096	25.80	0.089	0.099	5.90	0.105	0.085
Std. dev	-	-	11.62	0.072	0.136	8.92	0.121	0.134	2.45	0.098	0.098

Table 1: Average performance of methods across a suite of real-world datasets where the ground truth is unknown (and may not be sparse), sorted by how overdetermined the dataset is (n/p), and separated into the underdetermined and overdetermined cases. We also report standard deviations across the underdetermined and overdetermined datasets.

stability-adjusted method, while the kCV error is on average 21.8% smaller than the test set error for sparse ridge regression. This is especially visible for the most underdetermined datasets (Vessel and Riboflavin), where the kCV error for sparse ridge regression is significantly lower than all other methods, but this does not translate to better out-of-sample performance. In contrast, the nested kCV error is a substantially more accurate estimator of the test-set error.

3.2 Tree-Based Methods

We now empirically validate our nested cross-validation scheme on a suite of tree-based methods, for the same datasets as those studied in the previous section. The goal of this section is to establish that stability-regularization also improves the performance of tree-based methods.

We benchmark cyclic coordinate descent for the following two methods:

- A Julia implementation of CART [12] via the `DecisionTree.jl` package, where we iteratively optimize the 5-fold and nested 5-fold cross-validation error with respect to the tree depth over a grid of the integers $\{1, \ldots, 10\}$ and the `min_samples` parameter over a grid of the integers $\{2, \ldots, 10\}$, with the tree depth initially fixed to 5 for both approaches. Note that the `DecisionTree.jl` package holds all unspecified parameters to their default values, and thus all remaining CART parameters will take default parameters in this experiment.
- A Julia implementation of XGBoost [14] via the `XGBoost.jl` package, where we iteratively optimize the 5-fold and nested 5-fold cross-validation error with respect to the `max_depth` parameter over a grid of the integers $\{1, \ldots, 10\}$ and the `subsample` parameter over the grid[3] $\{0.01, 0.02, 0.03, \ldots, 1.0\}$, with the tree depth initially fixed to 5 for both approaches.

Table 2 depicts the dimensionality of each dataset, the average k-fold cross-validation error ("CV") or nested k-fold cross-validation error ("nCV"), and the average test set error ("MSE"), with and without nested cross-validation. For both methods, the first two columns correspond to the performance of k-fold cross-validation, and the second two correspond to nested stability-adjusted cross-validation.

[3]We initially tried increments of 0.05, where XGBoost selected identical hyperparameters with/without stability regularization.

As in the previous experiment, we measure the improvement from nested cross-validation vs. k-fold cross-validation by computing the average MSE across each dataset, normalizing by the MSE of k-fold cross-validation on that dataset to compute a percentage improvement across each dataset, and then taking the geometric mean across all datasets (to account for the fact that percentage improvement is an asymmetric measure). The average (geometric mean over averages for each dataset) runtime was 5.1s (resp. 285.8s) seconds for XGBoost without (with) nesting, and < 0.01s (resp. < 0.01s) for CART with/without nesting.

Dataset	n	p	CART				XGBoost			
			CV	MSE	nCV	MSE	CV	MSE	nCV	MSE
Wine	6497	11	0.534	0.530	0.539	0.527	0.422	0.406	0.429	0.406
Housing	506	13	19.149	20.33	20.03	20.35	12.83	12.95	14.30	12.95
Auto-MPG	392	25	14.93	17.09	16.83	17.34	10.37	11.98	11.23	11.28
Hitters	263	19	0.046	0.059	0.052	0.059	0.035	0.039	0.038	0.039
Prostate	97	8	0.691	0.803	0.780	0.814	0.545	0.672	0.609	0.667
Servo	167	19	0.314	0.403	0.390	0.394	0.18	0.163	0.217	0.163
Toxicity	38	9	0.078	0.102	0.089	0.101	0.066	0.074	0.079	0.075
Steam	25	8	1.230	1.067	1.542	1.069	0.844	1.251	0.991	1.282
Alcohol2	44	21	0.756	0.966	0.798	0.874	0.886	1.054	0.980	1.054
Avg.	-	-	4.192	4.596	4.561	4.614	2.908	3.176	3.208	3.101
Std. dev	-	-	7.369	8.053	7.915	8.109	5.483	5.287	4.796	5.144
TopGear	242	373	0.077	0.083	0.087	0.086	0.050	0.044	0.062	0.044
Bardet	120	200	0.016	0.018	0.017	0.020	0.012	0.016	0.013	0.016
Vessel	180	486	0.083	0.216	0.113	0.158	0.136	0.097	0.16	0.097
Riboflavin	71	4088	0.637	1.302	0.701	1.046	0.364	0.450	0.436	0.479
Avg.	-	-	0.203	0.405	0.168	0.328	0.140	0.151	0.168	0.159
Std. dev	-	-	0.291	0.604	0.317	0.483	0.189	0.202	0.158	0.202

Table 2: Average performance of methods across a suite of real-world datasets where the ground truth is unknown (and may not be sparse), sorted by how overdetermined the dataset is (n/p), and separated into the underdetermined and overdetermined cases. For both methods, the first two columns correspond to the performance of k-fold cross-validation, and the second two correspond to nested stability-regularized cross-validation.

We observe that for XGBoost, there is no benefit to the nested cross-validation procedure (average improvement of -0.3%), and we select the same hyperparameters whether or not we account for stability via nested cross-validation on 91.5% of instances. This is likely because XGBoost generates stable models by default (i.e., omitting one fold of the data hardly changes its predictions) and thus explicitly accounting for model stability does not improve its performance.

However, there is a significant benefit to nested cross-validation for CART: it improves the out-of-sample MSE by 4.1% on average, with a 1.2% average improvement on overdetermined datasets, and a 11.1% average improvement on underdetermined datasets. Moreover, the average test set MSE is within 7.4% of the average nested cross-validation error for CART (and 5.0% for XGBoost), vs. 27.2% inaccurate for k-fold cross-validation (and 8.19% for XGBoost), and thus nested cross-validation also significantly reduces out-of-sample disappointment.

4 Conclusion

In this work, we proposed a new approach to hyperparameter selection, namely selecting hyperparameters that minimize a weighted sum of the cross-validation error and the empirical hypothesis stability, with the weight in the weighted sum selected via a nested cross-validation procedure. Across a suite of real-world datasets, our approach improves the out-of-sample MSE by 4% on average for sparse ridge regression and CART, although it does not improve the performance of XGBoost. It also dramatically reduces the amount of out-of-sample disappointment experienced by the user.

Future work could involve developing a tighter bound on the test set error than the bound derived by [11], by controlling more moments of hypothesis stability. Indeed, [5] reports that controlling more moments of a pseudodistribution provides strictly tighter probabilistic guarantees on a quantity. It

would also be interesting to investigate the performance of our nested cross-validation procedure across a broader range of machine learning and operations contexts (e.g., neural networks), and to more extensively quantify the role that algorithmic stability plays in out-of-sample performance.

References

[1] G.-Y. Ban and C. Rudin. The big data newsvendor: Practical insights from machine learning. Operations Research, 67(1):90–108, 2019.

[2] G.-Y. Ban, N. El Karoui, and A. E. Lim. Machine learning and portfolio optimization. Management Science, 64(3):1136–1154, 2018.

[3] S. Bates, T. Hastie, and R. Tibshirani. Cross-validation: what does it estimate and how well does it do it? arXiv preprint arXiv:2104.00673, 2021.

[4] P. Bayle, A. Bayle, L. Janson, and L. Mackey. Cross-validation confidence intervals for test error. Advances in Neural Information Processing Systems, 33:16339–16350, 2020.

[5] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. SIAM Journal on Optimization, 15(3):780–804, 2005.

[6] D. Bertsimas and B. Sturt. Computation of exact bootstrap confidence intervals: Complexity and deterministic algorithms. Operations Research, 68(3):949–964, 2020.

[7] D. Bertsimas and B. Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. The Annals of Statistics, 48(1):300–323, 2020.

[8] D. Bertsimas, J. Pauphilet, and B. Van Parys. Sparse regression: Scalable algorithms and empirical performance. Statistical Science, 35(4):555–578, 2020.

[9] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. A unified approach to mixed-integer optimization problems with logical constraints. SIAM Journal on Optimization, 31(3):2340–2367, 2021.

[10] L. Bottmer, C. Croux, and I. Wilms. Sparse regression for large data sets with outliers. European Journal of Operational Research, 297(2):782–794, 2022.

[11] O. Bousquet and A. Elisseeff. Stability and generalization. The Journal of Machine Learning Research, 2:499–526, 2002.

[12] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Wadsworth and Brooks, 1984.

[13] G. C. Cawley and N. L. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. The Journal of Machine Learning Research, 11:2079–2107, 2010.

[14] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794, 2016.

[15] A.-A. Christidis, L. Lakshmanan, E. Smucler, and R. Zamar. Split regularized regression. Technometrics, 62(3):330–338, 2020.

[16] P. Cortez, A. Cerdeira, F. Almeida, and J. Matos, T.and Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

[17] D. Gamarnik and I. Zadik. Sparse high-dimensional linear regression. estimating squared error and a phase transition. The Annals of Statistics, 50(2):880–903, 2022.

[18] S. Geisser. The predictive sample reuse method with applications. Journal of the American statistical Association, 70(350):320–328, 1975.

[19] A. Gómez and O. A. Prokopyev. A mixed-integer fractional optimization approach to best subset selection. INFORMS Journal on Computing, 33(2):551–565, 2021.

[20] B. L. Gorissen, İ. Yanıkoğlu, and D. Den Hertog. A practical guide to robust optimization. Omega, 53:124–137, 2015.

[21] V. Gupta and N. Kallus. Data pooling in stochastic optimization. Management Science, 68(3): 1595–1615, 2022.

[22] V. Gupta and P. Rusmevichientong. Small-data, large-scale linear optimization with uncertain objectives. Management Science, 67(1):220–241, 2021.

[23] V. Gupta, M. Huang, and P. Rusmevichientong. Debiasing in-sample policy performance for small-data, large-scale optimization. Operations Research, 72(2):848–870, 2024.

[24] G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. A comment on "computational complexity of stochastic programming problems". Mathematical Programming, 159:557–569, 2016.

[25] M. Hardt and B. Recht. Patterns, predictions, and actions: A story about machine learning. arXiv preprint arXiv:2102.05242, 2021.

[26] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In International conference on machine learning, pages 1225–1234. PMLR, 2016.

[27] J. R. Harrison and J. G. March. Decision making and postdecision surprises. Administrative Science Quarterly, pages 26–42, 1984.

[28] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

[29] T. Hastie, R. Tibshirani, and R. Tibshirani. Best subset, forward stepwise or Lasso? analysis and recommendations based on extensive comparisons. Statistical Science, 35(4):579–592, 2020.

[30] H. Hazimeh and R. Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. Operations Research, 68(5):1517–1537, 2020.

[31] H. Hazimeh, R. Mazumder, and A. Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. Mathematical Programming, 196(1):347–388, 2022.

[32] D. Homrighausen and D. McDonald. The lasso, persistence, and cross-validation. In International conference on machine learning, pages 1031–1039. PMLR, 2013.

[33] D. A. Iancu and N. Trichakis. Pareto efficiency in robust optimization. Management Science, 60(1):130–147, 2014.

[34] M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In Proceedings of the tenth annual conference on Computational learning theory, pages 152–162, 1997.

[35] J. Liu, S. Rosen, C. Zhong, and C. Rudin. Okridge: Scalable optimal k-sparse ridge regression. Advances in neural information processing systems, 36:41076–41258, 2023.

[36] S. Liu and E. Dobriban. Ridge regression: Structure, cross-validation, and sketching. Proc. Int. Conf. Learn. Repres., 2020.

[37] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research, 29(3):464–484, 1981.

[38] R. O. Michaud. The markowitz optimization enigma: Is 'optimized'optimal? Financial analysts journal, 45(1):31–42, 1989.

[39] A. Y. Ng. Preventing" overfitting" of cross-validation data. In ICML, volume 97, pages 245–253, 1997.

[40] P. Patil, Y. Wei, A. Rinaldo, and R. Tibshirani. Uniform consistency of cross-validation estimators for high-dimensional ridge regression. In International conference on artificial intelligence and statistics, pages 3178–3186. PMLR, 2021.

[41] M. Plutowski, S. Sakata, and H. White. Cross-validation estimates imse. Advances in neural information processing systems, 6, 1993.

[42] M. Powell, M. Hosseini, J. Collins, C. Callahan-Flintoft, W. Jones, H. Bowman, and B. Wyble. I tried a bunch of things: the dangers of unexpected overfitting in classification. BioRxiv, page 078816, 2016.

[43] R. Quinlan. Auto MPG. UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5859H.

[44] R. B. Rao, G. Fung, and R. Rosales. On the dangers of cross-validation. an experimental evaluation. In Proceedings of the 2008 SIAM international conference on data mining, pages 588–596. SIAM, 2008.

[45] J. Reunanen. Overfitting in making comparisons between variable selection methods. Journal of Machine Learning Research, 3(Mar):1371–1382, 2003.

[46] R. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt. A meta-analysis of overfitting in machine learning. Advances in neural information processing systems, 32, 2019.

[47] P. Rousseeuw, C. Croux, V. Todorov, A. Ruckstuhl, M. Salibian-Barrera, T. Verbeke, M. Koller, and M. Maechler. Robustbase: basic robust statistics. R package version 0.4-5, 2009.

[48] J. Shao. Linear model selection by cross-validation. J. Amer. Stat. Assoc., 88(422):486–494, 1993.

[49] A. Shapiro, D. Dentcheva, and A. Ruszczynski. Lectures on stochastic programming: modeling and theory. SIAM, 2021.

[50] J. P. Simmons, L. D. Nelson, and U. Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. Psychological science, 22(11):1359–1366, 2011.

[51] J. E. Smith and R. L. Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. Management Science, 52(3):311–322, 2006.

[52] W. Stephenson, Z. Frangella, M. Udell, and T. Broderick. Can we globally optimize cross-validation loss? quasiconvexity in ridge regression. Advances in Neural Information Processing Systems, 34, 2021.

[53] M. Stone. Cross-validatory choice and assessment of statistical predictions. Journal of the royal statistical society: Series B (Methodological), 36(2):111–133, 1974.

[54] K. Ulrich. Servo. UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5Q30F.

[55] B. P. Van Parys, P. J. Goulart, and D. Kuhn. Generalized gauss inequalities via semidefinite programming. Mathematical Programming, 156:271–302, 2016.

[56] B. P. Van Parys, P. M. Esfahani, and D. Kuhn. From data to decisions: Distributionally robust optimization is optimal. Management Science, 67(6):3387–3402, 2021.

[57] V. N. Vapnik. An overview of statistical learning theory. IEEE transactions on neural networks, 10(5):988–999, 1999.

[58] K. Weber, R. Eisman, L. Morey, A. Patty, J. Sparks, M. Tausek, and Z.-B. Zeng. An analysis of polygenes affecting wing shape on chromosome 3 in drosophila melanogaster. Genetics, 153 (2):773–786, 1999.

[59] W. Xie and X. Deng. Scalable algorithms for the sparse ridge regression. SIAM Journal on Optimization, 30(4):3359–3386, 2020.

[60] C. Ye, Y. Yang, and Y. Yang. Sparsity oriented importance learning for high-dimensional linear regression. Journal of the American Statistical Association, 113(524):1797–1812, 2018.

A Data Generation Process for Heatmaps

Our data generation process for the motivating example follows the data generation process used by [8] and is as follows:

1. The rows of the model matrix are generated iid from a p-dimensional multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\Sigma_{ij} = \rho^{|i-j|}$ for all $i, j \in [p]$.
2. A "ground-truth" vector $\boldsymbol{\beta}_{\text{true}}$ is sampled with exactly τ_{true} non-zero coefficients. The position of the non-zero entries is randomly chosen from a uniform distribution, and the value of the non-zero entries is either 1 or -1 with equal probability.
3. The response vector is generated as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_{\text{true}} + \boldsymbol{\varepsilon}$, where each ε_i is generated iid from a scaled normal distribution such that $\sqrt{\nu} = \|\boldsymbol{X}\boldsymbol{\beta}_{\text{true}}\|_2 / \|\boldsymbol{\varepsilon}\|_2$.
4. We standardize $\boldsymbol{X}, \boldsymbol{y}$ to normalize and center them.
5. We generate a separate test set of $n_{\text{test}} = 10,000$ observations drawn from the same underlying stochastic process to measure test set performance, and normalized using the same coefficients as the training set.

B Heatmaps From Globally Minimizing Five-Fold Cross-Validation Error

We now revisit the problem setting considered in Figure 1, using five-fold rather than leave-one-out cross-validation (Figure 2). Our conclusions remain consistent with Figure 1.

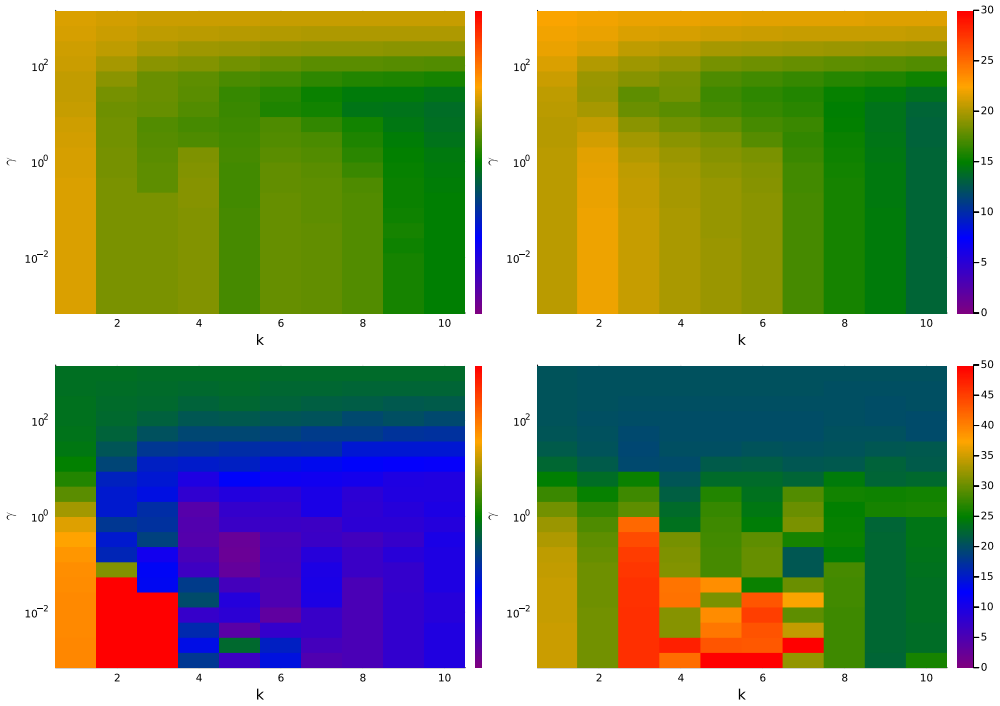

Figure 2: Five-fold (left) and test (right) error for varying τ and γ, for the overdetermined setting (top, $n = 50, p = 10$) and an underdetermined setting (bottom, $n = 10, p = 50$) considered in Figure 1. In the overdetermined setting, the five-fold error is a good estimate of the test error for most values of parameters (γ, τ). In contrast, in the underdetermined setting, the five-fold error is a poor approximation of the test error, and the estimator that minimizes the five-fold error ($\gamma = 6.15$, $\tau = 5$) significantly disappoint out-of-sample.

B.1 Proof of Theorem 1

Proof. The result follows analogously to [11, Theorem 11]; the main novelty in this proof compared to [11, Theorem 11] is the use of a more general notion of hypothesis stability. In particular, Bousquet and Elisseeff [11]'s definition is sufficient to derive the result for leave-one-out, but not for k-fold.

Let $\mathcal{R} := \frac{1}{|\mathcal{S}|}\ell(y_i, \boldsymbol{\beta}(\boldsymbol{\theta}, \boldsymbol{x}_i))$ denote the average test set error on an unseen observation, and $\mathcal{R}_{CV} := \frac{1}{n}\sum_{j\in[k]} h_j(\boldsymbol{\theta})$ denote the average k-fold cross-validation error. Further, let $\mathbb{E}[\ell(A_{\mathcal{S}}, z)]$ denote the expected generalization error of a regressor trained on a training set $\mathcal{S}$ and evaluated on an example $z = (\boldsymbol{x}_i, y_i)$ drawn from the same distribution but not included in the test set. Let z' denote an independent draw to z, and $\mathcal{S}^{(\mathcal{N}_j)}$ denote a training set with the jth fold of the data omitted.

Then, analogously to [11, Lemma 9], letting $i \neq j$, one can show that

$$\begin{aligned}
\mathbb{E}[(\mathcal{R} - \mathcal{R}_{CV})^2] &\leq \mathbb{E}_{\mathcal{S},z,z'}[\ell(A_{\mathcal{S}}, z)\ell(A_{\mathcal{S}}, z')] - 2\mathbb{E}_{\mathcal{S},z}[\ell(A_{\mathcal{S}}, z)\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)] \qquad (9)\\
&\quad + \frac{n - n/k}{n}\mathbb{E}_{\mathcal{S}}[\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)\ell(A_{\mathcal{S}^{(\mathcal{N}_j)}}, z_j)] + \frac{M}{k}\mathbb{E}_{\mathcal{S}}[\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)]\\
&= \frac{1}{k}\mathbb{E}_{\mathcal{S}}[\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)(M - \ell(A_{\mathcal{S}^{(\mathcal{N}_j)}}, z_j)] \qquad (10)\\
&\quad + \mathbb{E}_{\mathcal{S},z,z'}[\ell(A_{\mathcal{S}}, z)\ell(A_{\mathcal{S}}, z') - \ell(A_{\mathcal{S}}, z)\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)]\\
&\quad + \mathbb{E}_{\mathcal{S},z,z'}[\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)\ell(A_{\mathcal{S}^{(\mathcal{N}_j)}}, z_j) - \ell(A_{\mathcal{S}}, z)\ell(A_{\mathcal{S}^{(\mathcal{N}_i)}}, z_i)]\\
&= I_1 + I_2 + I_3,
\end{aligned}$$

where we let I_1, I_2, I_3 stand for the terms in the first, second, and third lines of the right-hand side.

Further, it follows directly from Schwarz's inequality [see also 11, pp. 522] that $I_1 \leq \frac{M^2}{2k}$ and it follows analogously to [11, pp. 522] that $I_2 + I_3 \leq 3M\mu_h$. Therefore, we have that

$$\mathbb{E}[(\mathcal{R} - \mathcal{R}_{cv})^2] \leq \frac{M^2}{2k} + 3M\mu_h.$$

Finally, the result follows from Chebyshev's inequality. □

C Pseudocode for Nested Stability-Regularized Cross-Validation

D Dataset Description

We use a variety of real datasets from the literature in our computational experiments. The information of each dataset is summarized in Table 3. Note that we increased the number of features on selected datasets by including second-order interactions.

Our sources for these datasets are as follows:

- Four UCI datasets: Auto-MPG, Housing, Servo, and Wine. We obtained these datasets from the online supplement to [19].
- The `alcohol` dataset distributed via the `R` package `robustbase`. Note that we increased the number of features for this dataset by including second-order interactions.
- The `bardet` dataset provided by [60].
- The `hitters` Kaggle dataset, after preprocessing the dataset to remove rows with any missing entries, and transforming the response by taking log(Salary), as is standard when predicting salaries via regression.
- The `Prostate` dataset distributed via the R package `ncvreg`.
- The `Riboflavin` dataset distributed by the `R` package `hdi`.
- The `steamUse` dataset provided by [47].
- The `topgear` dataset provided by [10].
- The `toxicity` dataset provided by [47].
- The `vessel` dataset made publicly available by [15].
- The `wing` dataset made publicly available by [58]

Algorithm 1: Stability-regularized nested k-fold cross-validation for model selection

Input: dataset $\{(\mathbf{x}_i, y_i)\}_{i\in[n]} \in \mathcal{X} \times \mathcal{Y}$; number of folds k; learning algorithm $\boldsymbol{\beta}$; loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$; hyper-parameter grid $\mathcal{H}$; stability grid Λ

Output: outer scores $\{s_1, \ldots, s_K\}$, chosen weight λ, chosen hyper-parameters $h^\star$

Randomly partition dataset into k disjoint partitions indexed by $\mathcal{N}_1, \ldots, \mathcal{N}_K$

for $\lambda \in \Lambda$ **do** // Stability weight loop

  for $t \leftarrow 1$ **to** k **do** // outer loop to evaluate λ

    $\mathcal{D}_{\text{outer}} \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{N}_t}$

    $\mathcal{D}_{\text{rest}} \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i\in[n]\setminus\mathcal{N}_t}$

    // inner loop: hyper-parameter search

    foreach $h \in \mathcal{H}$ **do**

      $\boldsymbol{\beta}^{(\mathcal{N}_t)}(h) \leftarrow \text{fit\_model}(\mathcal{D}_{\text{rest}}, h)$

      for $t_2 \leftarrow 1$ **to** $k : t \neq t_2$ **do**

        $\mathcal{D}_{\text{train}}\{(\boldsymbol{x}_i, y_i)\}_{i\in[n]\setminus\mathcal{N}_t\cup\mathcal{N}_{t_2}}$

        $\mathcal{D}_{\text{val}} \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{N}_{t_2}}$

        $\boldsymbol{\beta}^{(\mathcal{N}_t\cup\mathcal{N}_{t_2})}(h) \leftarrow \text{fit\_model}(\mathcal{D}_{\text{train}}, h)$

        $s_{t_2}(h) \leftarrow \frac{1}{|\mathcal{N}_{t_2}|}\sum_{i\in\mathcal{N}_{t_2}} \ell(y_i, \boldsymbol{\beta}^{(\mathcal{N}_t\cup\mathcal{N}_{t_2})}(h, \boldsymbol{x}_i))$

        $\mu(h) \leftarrow$
        $\max(\mu(h), \frac{1}{|n-|\mathcal{N}_t|}\sum_{i\in[n]\setminus\mathcal{N}_t} \left|\ell\left(y_i, \boldsymbol{\beta}^{(\mathcal{N}_t\cup\mathcal{N}_{t_2})}(h, \boldsymbol{x}_i)\right) - \left(y_i, \boldsymbol{\beta}^{(\mathcal{N}_t)}(h, \boldsymbol{x}_i)\right)\right|$

      $\tilde{s}(h) \leftarrow \dfrac{1}{L}\sum_{t_2} s_{t_2}(h)$

    // mean inner kCV score

    $h^\star_\lambda \leftarrow \arg\min_{h\in\mathcal{H}} \tilde{s}(h) + \lambda \cdot \mu(h)$ // pick best regularized score

    $m_{\lambda,t} \leftarrow \sum_{i\in\mathcal{N}_t} \ell(y_i, \boldsymbol{\beta}^{(\mathcal{N}_t)}(h^\star_\lambda, \boldsymbol{x}_i))$

    // Evaluate performance of optimized model on remaining fold

$\bar{m}_\lambda \leftarrow \dfrac{1}{n}\sum_{t=1}^{k} m_{\lambda,t}$

// Estimated performance with regularization λ

$\lambda^\star \leftarrow \arg\min_{\lambda\in\Lambda} \bar{m}_\lambda$

// Find best λ out-of-sample

foreach $h \in \mathcal{H}$ **do**

  // Now do regularized k-fold CV

  $\boldsymbol{\beta}(h) \leftarrow \text{fit\_model}(\{(\boldsymbol{x}_i, y_i)\}_{i\in[n]}, h)$

  foreach $t \in [k]$ **do**

    $\boldsymbol{\beta}^{(\mathcal{N}_t)}(h) \leftarrow \text{fit\_model}(\{(\boldsymbol{x}_i, y_i)\}_{i\in[n]\setminus\mathcal{N}_t}, h)$

    $s_t(h) \leftarrow \frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t} \ell(y_i, \boldsymbol{\beta}^{(\mathcal{N}_t)}(h, \boldsymbol{x}_i))$

    $\mu(h) \leftarrow \max(\mu(h), \frac{1}{n}\sum_{i\in[n]} \left|\ell\left(y_i, \boldsymbol{\beta}^{(\mathcal{N}_t)}(h, \boldsymbol{x}_i)\right) - (y_i, \boldsymbol{\beta}(h, \boldsymbol{x}_i))\right|$

  $h^\star \leftarrow \arg\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{t\in[k]} s_t(h) + \lambda^\star\mu(h)$

return $\min_{\lambda\in\Lambda}(\bar{m}_\lambda), \lambda^\star, h^\star$

E Supplementary Results for Sparse Ridge Regression

Dataset	n	p	Notes	Reference
Wing	701	38	[58]	
Housing	506	13		[19]
Wine	6497	11		[16]
AutoMPG	392	25		[43]
Hitters	263	19	Removed rows with missing data $\boldsymbol{y} = \log(\text{salary})$	Kaggle
Prostate	97	8		R Package `ncvreg`
Servo	167	19	One-hot encoding of features	[54]
Toxicity	38	9		[47]
SteamUse	25	8		[47]
Alcohol2	44	21	2nd order interactions added	[47]
TopGear	242	373		[10]
BarDet	120	200		[60]
Vessel	180	486		[15]
Riboflavin	71	4088		R package `hdi`

Table 3: Real datasets used.

Dataset	n	p	SCAD			GLMNet			L0Learn		
			τ	CV	MSE	τ	kCV	MSE	τ	CV	MSE
Wine	6497	11	11	0.543	0.542	11	0.542	0.542	11	0.542	0.542
Housing	506	13	11.2	24.34	23.69	12.1	23.33	23.79	11	23.48	23.61
Auto-MPG	392	25	17.3	8.894	9.028	19.6	8.705	8.880	15.7	8.766	8.979
Hitters	263	19	12.1	0.083	0.085	12.2	0.075	0.080	8.9	0.077	0.082
Prostate	97	8	6.7	0.529	0.557	6.9	0.508	0.569	3.2	0.511	0.547
Servo	167	19	12.3	0.707	0.706	15.8	0.676	0.709	11.3	0.687	0.735
Toxicity	38	9	3.5	0.044	0.055	6.1	0.038	0.054	3.2	0.032	0.063
Steam	25	8	2.9	0.485	0.504	4.1	0.450	0.497	4.4	0.493	0.428
Alcohol2	44	21	2.4	0.249	0.271	5.8	0.222	0.240	7.4	0.200	0.287
Avg. Overdet	-	-	8.82	3.986	3.937	10.40	3.839	3.929	8.46	3.865	3.919
Std. dev. Overdet	-	-	5.18	8.139	7.942	5.146	7.975	7.964	4.280	8.139	7.915
TopGear	242	373	9.8	0.053	0.063	31.8	0.044	0.048	44.1	0.050	0.060
Bardet	120	200	9.4	0.009	0.009	32.8	0.007	0.009	35.9	0.007	0.009
Vessel	180	486	9.7	0.034	0.037	40.0	0.018	0.021	14.5	0.023	0.025
Riboflavin	71	4088	16.3	0.331	0.358	88.8	0.195	0.276	39.3	0.205	0.352
Avg. Underdet	-	-	11.30	0.106	0.117	48.35	0.066	0.088	33.45	0.071	0.112
Std. Dev. Underdet	-	-	3.39	0.144	0.162	27.22	0.151	0.126	13.07	0.091	0.162

Table 4: Results from Table 1 (continued).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We would like to refer to sections 2, 3 and 4

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We would like to refer to section 2, section 3 and section 4

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We would like to refer to Section 2 and Appendix B.1

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We would like to refer to Section 3

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The main contribution of the paper lies in the theory and methodology for improving the performance of cross-validation. All algorithms and details on experiments are present in the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We would like to refer to Section 3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We confirm that we report standard deviations for all results averaged over in tables. We also highlight in the abstract and introduction that our results are averaged over 13 real-world datasets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: We would like to refer to the description of the computational resources at the beginning of Section 3

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have verified that we follow all items in the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: there is no societal impact of the work performed, because the paper proposes new techniques for improving the performance of cross-validation, rather than any direct societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: the paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: the paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not used in any part of the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.