STEADYING NEURAL FEATURE SELECTION THROUGH COUPLED RESIDUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Feature selection aims to retain the informative features in a dataset, reducing dimensionality and computational costs while improving model interpretability and performance. Recently, neural-network-based methods have emerged as promising approaches to feature selection. However, the non-identifiability of neural networks can make the selected features unstable, causing difficulties in interpretability. To address this issue, we propose a coupled residual connection, a simple variant of a residual connection that integrates a linear residual component and a nonlinear neural network through a shared learnable weight. This coupling strategy ensures the identifiability of feature-selection-related parameters while remaining compatible with existing neural network architectures, such as multilayer perceptrons and mixtures of experts. Based on this proposal, we introduce a straightforward training-then-trimming algorithm. Experiments on synthetic and real-world datasets demonstrate that the proposed algorithm typically yields more stable results and improves prediction performance by 9%–50%. We further observe that incorporating our approach into an existing method improves its stability and predictive performance.

Keywords: Interpretability, Feature Selection, Neural Network, Residual Learning

1 INTRODUCTION

Feature selection aims to identify the most relevant and informative features in a dataset while removing irrelevant, noisy, or redundant ones. It is central to many scientific and societal applications because it can provide insight into the underlying system. For example, in biomedicine, selecting informative features from high-dimensional omics and single-cell data supports reliable biomarker discovery (Hédou et al., 2024). In materials science, selecting compact physicochemical descriptors can reveal structure–property relationships and guide catalyst design (Han et al., 2024). Furthermore, feature selection can improve robustness and generalization while reducing effective model complexity (Li et al., 2024; Liao et al., 2024). Therefore, feature selection remains a fundamental and active problem in machine learning and data-driven science.

As neural networks can approximate complex nonlinear effects in data, feature selection methods designed for neural networks have received growing attention in recent years (Feng & Simon, 2017; Yamada et al., 2020; Chen et al., 2021; Yang et al., 2022; Park & Lee, 2024; Jiang et al., 2024; Luo & Halabi, 2025; Yeo et al., 2025). One important class of methods introduces an additional learnable vector $\boldsymbol{\beta}$ to scale the input vector $\mathbf{x}$ before feeding it into the neural network h_Θ (see Figure 1 for an example). The learned $\boldsymbol{\beta}$ is then used for feature selection: variables associated with large-magnitude coefficients are selected. These approaches are widely adopted because they enable simple feature selection via neural networks, with the feature selection results obtained once neural network training is complete.

Despite the computational simplicity and decent performance of these proposed neural-network-based feature selection methods, the inherent non-identifiability of neural networks poses challenges to the stability of feature selection. The non-identifiability of neural networks refers to the phenomenon in which multiple distinct parameter configurations of a neural network can represent the same function.

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

Figure 1 conceptually illustrates this phenomenon: two neural networks essentially represent the same function but have different parameter configurations. Different parameter configurations lead to different feature selection results—the first neural network selects the second feature because β_2 is larger, whereas the second network selects the first feature because β_1 is larger. In practice, this non-identifiability is also likely to occur (see Figure 2 for a practical example). Random initialization and minibatch sampling can cause training to converge to different parameterizations representing similar functions, thereby inducing variability in the learned feature weights. In summary, both the conceptual and practical examples illustrate that non-identifiability can cause instability in feature selection, which we would like to mitigate by reducing run-to-run variation in the selected features.

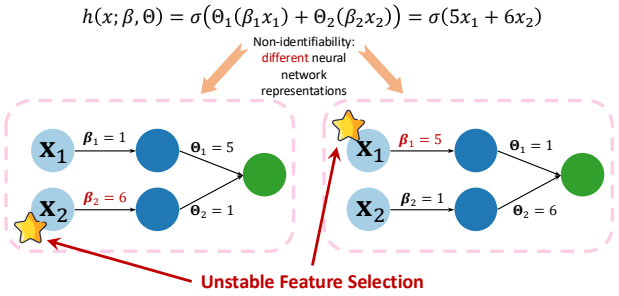

Figure 1: An analytical example of two neural networks without hidden layers that represent the same function. The two different parameterizations can lead to different feature selections when selection is based on the values of β, demonstrating the potential instability caused by neural-network non-identifiability.

In this paper, we propose a variant of the residual connection (He et al., 2016), called the coupled residual connection. Our analysis shows that the coupled residual connection enforces the identifiability of parameters associated with feature selection. Based on the coupled residual connection, we develop a lightweight two-step feature selection algorithm consisting of training and trimming, where features with small learned coefficients are trimmed. This connection can be integrated seamlessly with standard neural networks without requiring modifications to the original architecture and can thus be applied to diverse tasks, including regression and classification. Empirically, we conduct extensive numerical studies to test the proposed method. The results from more than 20 dataset settings show that (i) it helps steady learned neural feature selection and improves its downstream predictive performance; (ii) its selected features are typically associated with superior or competitive predictive performance on real-world datasets; and (iii) incorporating the proposed connection into an existing neural feature selection method generally reduces variation and improves its downstream predictive performance.

2 Related Work

We review the two lines of work that are most related to the proposal. To save space, additional related work is deferred to Appendix A.

Neural-Network-Based Feature Selection. Basically, there are three categories of feature selection methods based on neural networks: (i) regularization-based methods, (ii) iterative search methods, and (iii) stochastic gating methods. Regularization-based methods penalize input-layer parameters or impose hierarchical constraints to encourage sparse inputs (Feng & Simon, 2017; Dinh & Ho, 2020; Lemhadri et al., 2021; Wang et al., 2021; Luo & Halabi, 2025; Saha & Pal, 2024; Yeo et al., 2025). Iterative search methods extend forward and backward selection, as well as their advanced variants, to predictors fed into the neural network (Marcano-Cedeño et al., 2010; Haq et al., 2019; Chen et al., 2021; Yasuda et al., 2023; Atashgahi et al., 2024; Liu et al., 2024; Yang et al., 2024). Stochastic-gate methods use continuous relaxations of discrete random variables (Maddison et al., 2017; Jang et al., 2017; Louizos et al., 2018) to facilitate feature selection (Yamada et al., 2020; Lee et al., 2021; Yang et al., 2022; Park & Lee, 2024; Jiang et al., 2024).

Existing methods primarily aim to identify relevant features, whereas our proposal targets steadying feature selection across runs. Its simple design also allows it to be integrated with different network architectures and existing feature selection methods.

Identifiability of Neural Networks. Parameter identifiability is a critical topic in neural network research. Early studies demonstrated that neural networks with smooth activation functions, such as tanh, logistic sigmoid, and Gaussian-type functions, can achieve identifiability under certain conditions (Sussmann, 1992; Albertini et al., 1993; Fefferman, 1994; Kainen et al., 1994; Kůrková & Kainen, 1994). In modern deep learning, the ReLU activation function (Nair & Hinton, 2010) has gained widespread popularity due to its ability to mitigate the vanishing gradient. However, ReLU networks are generally not identifiable (Schmidt-Hieber, 2020), except under specific sufficient conditions (Bui Thi Mai & Lampert, 2020; Petzka et al., 2020; Rolnick & Kording, 2020; Grigsby et al., 2023; Bona-Pellissier et al., 2023) or when considered under the weaker notion of local identifiability (Bona-Pellissier et al., 2022; Stock & Gribonval, 2023).

Unlike prior work on hidden-unit identifiability, we use the coupled residual connection to ensure the partial identifiability needed for stable feature selection, without requiring the entire network to be identifiable.

3 Background and Motivation

Notation. Suppose we are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, consisting of n independent and identically distributed observations, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$ represents a p-dimensional feature vector and $y_i \in \mathbb{R}$ denotes the corresponding response. Finally, $\mathcal{L}(\cdot, \cdot)$ represents a loss function appropriate for the supervised learning task. $\mathbf{x} \circ \boldsymbol{\beta} = (x_1\beta_1, \ldots, x_j\beta_j, \ldots, x_p\beta_p)^\top \in \mathbb{R}^p$ denotes the Hadamard product of $\mathbf{x} = (x_1, \ldots, x_p)^\top \in \mathbb{R}^p$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$.

Let $h_\Theta(\cdot)$ denote an arbitrary neural network that maps p-dimensional feature vectors to real-valued predictions, where Θ collects the learnable parameters $\Theta^{(1)}, \Theta^{(2)}, \ldots$ in the neural network, and $\Theta^{(i)}$ denotes the parameters mapping the hidden units in the $(i-1)$th layer to those in the ith layer[1]. We say that a parameterized model $f_\Omega(\mathbf{x})$ is identifiable if $f_\Omega(\mathbf{x}) = f_{\Omega'}(\mathbf{x})$ for every $\mathbf{x}$ implies $\Omega = \Omega'$. It is well known that a neural network $h_\Theta(\cdot)$ that uses ReLU as its activation function is not identifiable (Schmidt-Hieber, 2020).

Existing methods and identifiability. Many neural feature-selection methods introduce learnable input-level gates or masks and optimize them jointly with the neural network parameters (Yamada et al., 2020; Chen et al., 2021; Yang et al., 2022; Dong & Arik, 2023; Park & Lee, 2024; Jiang et al., 2024; Pad et al., 2025). A representative deterministic formulation introduces an additional learnable parameter $\boldsymbol{\beta}$ that multiplies $\mathbf{x}$:

$$\widehat{\boldsymbol{\beta}}, \widehat{\Theta} \leftarrow \arg\min_{\boldsymbol{\beta}, \Theta} \sum_{i=1}^n \frac{1}{n} \mathcal{L}(y_i, h_\Theta(\mathbf{x}_i \circ \boldsymbol{\beta})).$$

The learned $\widehat{\boldsymbol{\beta}}$ is then used for feature selection. For example, Chen et al. (2021) selects the jth variable if $|\widehat{\boldsymbol{\beta}}_j|$ exceeds a specified threshold $\eta > 0$. However, it is easy to see that both Θ and $\boldsymbol{\beta}$ are non-identifiable in the unconstrained formulation above. One example is that scaling $\boldsymbol{\beta}$ up by a constant factor c and scaling down the first layer of the neural network $\Theta^{(1)}$ lead to the same neural network function. This causes the learned parameter $\boldsymbol{\beta}$ to become unstable during training. An example is shown in Figure 2. Across different datasets drawn from the same distribution, the empirical distribution of $\boldsymbol{\beta}$ (shown in green) returned by the existing method exhibits substantial variation. This variation would propagate to feature selection because the existing method selects features via $\{j : |\widehat{\boldsymbol{\beta}}_j| > \eta\}$.

Related methods instead induce sparsity directly through feature-wise groups of first-layer weights, without introducing a separate input gate (Dinh & Ho, 2020; Wang et al., 2021; Gong et al., 2022; Luo & Halabi, 2025; Sandberg et al., 2024; Saha & Pal, 2024):

$$\Theta \leftarrow \arg\min_{\Theta} \sum_{i=1}^n \frac{1}{n} \mathcal{L}(y_i, h_\Theta(\mathbf{x}_i)) + \sum_{j=1}^p \rho(\Theta_j^{(1)})$$

[1]For consistency, we define the zeroth-layer hidden units as the input vector.

where $\Theta_j^{(1)}$ is the parameter mapping x_i to the hidden units in the first layer, and $\rho(\cdot)$ is a group-sparsity-inducing penalty, such as the group Lasso penalty (Yuan & Lin, 2005). Because of the non-identifiability of $\Theta^{(1)}$, this approach may still suffer from instability in feature selection.

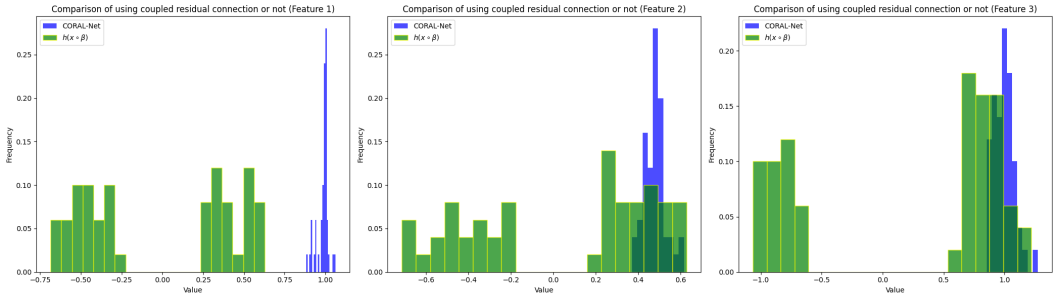

Figure 2: Density plots of the estimated coefficients β_1, β_2, and β_3 for $h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$ and CORAL-Net ($\mathbf{x}^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$) over 50 replications. Both networks share the same architecture and were trained independently on the Artificial dataset. The green bars represent $h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$, while the blue bars represent CORAL-Net.

4 Coupled Residual Learning

4.1 Coupled Residual Connection

We introduce a neural network architecture with a coupled residual (CORAL) connection, defined as follows:

$$f(\mathbf{x}; \boldsymbol{\beta}, \Theta) = \mathbf{x}^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x} \circ \boldsymbol{\beta}),$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ and $h_\Theta(\cdot)$ represents a neural network parameterized by Θ. For convenience, we refer to this architecture as CORAL-Net, and its structure is illustrated in Figure 3. The network consists of two additive components: (i) a linear model $\mathbf{x}^\top \boldsymbol{\beta}$, which captures the main effects of the features, and (ii) a neural network $h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$, which learns complex nonlinear relationships while weighting the features via $\boldsymbol{\beta}$. This design separates linear and nonlinear effects, a strategy with a long history in econometrics, statistics, and machine learning (Cox, 1972; Engle et al., 1986; He et al., 2016).

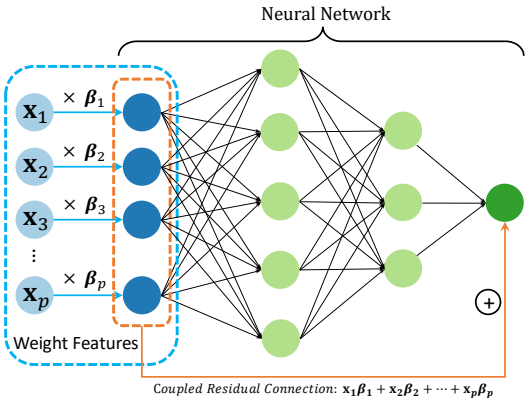

Figure 3: The architecture of the coupled residual connection.

The main innovation of CORAL-Net lies in applying a shared learnable weight $\boldsymbol{\beta}$ to both the residual connection $\mathbf{x}^\top \boldsymbol{\beta}$ and the neural network $h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$, effectively coupling the linear and nonlinear components to provide a unified representation. A key property of CORAL-Net is its partial identifiability with respect to $\boldsymbol{\beta}$, which is demonstrated in the following proposition.

Proposition 1 (Partial Identifiability)**.** *For an arbitrary standard neural network $h_\Theta(\mathbf{x})$, if its last hidden layer has finitely many units and its activation functions are bounded, then CORAL-Net is identifiable with respect to $\boldsymbol{\beta}$, i.e., $\mathbf{x}^\top \boldsymbol{\beta}_1 + h_{\Theta_1}(\mathbf{x} \circ \boldsymbol{\beta}_1) = \mathbf{x}^\top \boldsymbol{\beta}_2 + h_{\Theta_2}(\mathbf{x} \circ \boldsymbol{\beta}_2)$ implies $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$.*

Partial identifiability removes one structural source of variability in $\boldsymbol{\beta}$ by ruling out the parameterization ambiguity illustrated in Figure 1, in which the same network function is represented by different values of $\boldsymbol{\beta}$. Figure 2 provides empirical evidence consistent with this effect: the estimates of $\boldsymbol{\beta}$ obtained by CORAL-Net are more tightly concentrated around their means. Proposition 1 continues to hold even when $h_\Theta(\mathbf{x})$ is very deep or non-identifiable, provided that the coupled residual

connection $\mathbf{x}^\top \boldsymbol{\beta}$ is retained. However, without the coupled residual connection, this identifiability guarantee does not generally apply, and the estimates exhibit greater variation, as can be seen in Figure 2. Additionally, the boundedness assumption on the activation functions can be satisfied by many bounded activations, such as sigmoid and ReLU6 (Howard et al., 2017). Nonetheless, our experiments demonstrate that with the commonly used ReLU activation, CORAL achieves strong empirical performance, indicating that boundedness is a sufficient condition but may not be a necessary one.

Beyond feature-level identifiability, CORAL-Net retains several desirable properties of standard neural networks in terms of function representation, computation, and implementation.

- *Function representation.* A useful property is that CORAL-Net retains the expressive power of its nonlinear backbone. The following proposition shows that the coupled residual connection preserves the universal approximation property (Hornik et al., 1989; Kidger & Lyons, 2020; Park et al., 2021):

  Proposition 2. *If the class of neural networks h_Θ has the universal approximation property on compact domains, then the CORAL-Net class $\mathbf{x}^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$ also has this property.*

  Thus, the coupling strategy introduces identifiability without narrowing the class of functions that can be approximated.
- *Computation.* CORAL-Net is also lightweight to train. It introduces p additional parameters, which is typically a marginal increase relative to the number of parameters in h_Θ. Thus, training CORAL-Net has nearly the same order of computational complexity as training h_Θ alone, and it retains the scalability of the original backbone h_Θ to large datasets (see Figure A2).
- *Implementation.* CORAL can be added as a wrapper around a standard differentiable network without redesigning the backbone h_Θ, loss function, or optimizer. This wrapper-style construction makes CORAL applicable to different neural-network backbones, such as multilayer perceptrons (MLPs) and mixture-of-experts (MoE) models, and to both regression and classification tasks (see Figure 6). It can also be incorporated into existing methods to improve their performance (see Sections 4.3–5.4 for illustration).

4.2 Feature selection via CORAL-Net

In CORAL-Net, if $\beta_j = 0$, the jth feature contributes neither linearly nor nonlinearly to the prediction and is therefore excluded. If $\beta_j \neq 0$, the feature remains available to both branches and is retained by the model. This direct link between β_j and feature inclusion motivates the following feature-selection procedure.

Specifically, we design a two-stage method consisting of a *training* stage and a *trimming* stage. In the training stage, we estimate $\widehat{\boldsymbol{\beta}}$ by solving the following optimization problem:

$$\widehat{\boldsymbol{\beta}}, \widehat{\Theta} \leftarrow \underset{\boldsymbol{\beta} \in \mathbb{R}^p, \Theta}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathbf{x}_i^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x}_i \circ \boldsymbol{\beta})). \tag{1}$$

Because training does not enforce exact sparsity, $\widehat{\boldsymbol{\beta}}$ is generally dense, and thus, we construct the selected subset by retaining features with sufficiently large coefficient magnitudes. Specifically, in the trimming stage, we separate relevant features from irrelevant ones by introducing a threshold δ. Specifically, features with $|\widehat{\beta}_j|$ exceeding δ are selected, while others are excluded. This trimming step effectively screens out irrelevant features with empirically negligible weights. The overall feature selection procedure, comprising training via optimization problem (1) and trimming through thresholding, is summarized in Algorithm 1.

Algorithm 1 CORAL for feature selection.

Require: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and a positive value δ.

1: Solve the optimization problem (1) to obtain the estimated model parameters $\widehat{\boldsymbol{\beta}}$ and $\widehat{\Theta}$.
2: $\widehat{\mathcal{A}} \leftarrow \{j \mid |\widehat{\boldsymbol{\beta}}_j| > \delta\}$.

Ensure: $\widehat{\mathcal{A}}$.

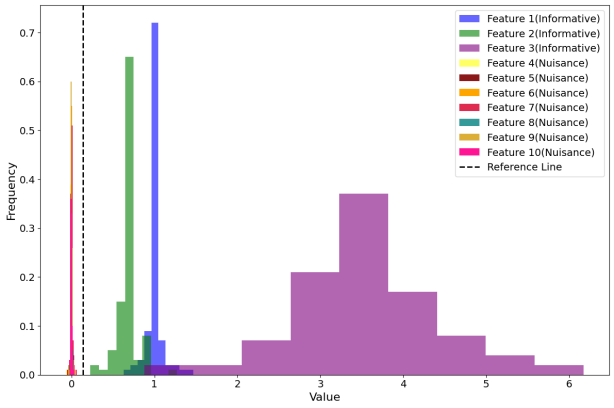

Figure 4: The empirical distribution of $\widehat{\beta}_1, \ldots, \widehat{\beta}_{10}$ estimated by CORAL-Net using 50 datasets randomly sampled from the Artificial dataset. Different colors represent different features.

The effectiveness of Algorithm 1 is illustrated by training CORAL-Net on a synthetic dataset. The parameter estimates are shown in Figure 4, in which the first three features are informative while the remaining features are irrelevant. As shown in Figure 4, the estimated values of $\widehat{\boldsymbol{\beta}}_1, \widehat{\boldsymbol{\beta}}_2$, and $\widehat{\boldsymbol{\beta}}_3$ are significantly larger in magnitude than the near-zero values of irrelevant features. This clear separation allows us to apply an appropriate threshold (illustrated by the dashed line) to distinguish effectively between relevant and irrelevant features.

The choice of the threshold δ in Algorithm 1 depends on the desired feature budget. In our experiments, all methods are compared at matched feature budgets. For a prescribed budget s, we retain the s features with the largest values of $|\widehat{\beta}_j|$, which is equivalent to choosing δ between the sth and $(s+1)$th largest coefficient magnitudes. When the feature budget is unknown, δ can instead be selected using cross-validation.

4.3 Extension: Multi-output Responses

Multi-output settings arise in applications such as classification and multi-response regression. Next, we design CORAL-Net to select features that are useful for predicting at least one response. Suppose the response has m components. Let $\boldsymbol{B} \in \mathbb{R}^{p \times m}$ be a learnable parameter matrix, and let $\boldsymbol{B}_j \in \mathbb{R}^m$ denote its jth row, which collects the parameters of the jth feature across the m outputs. The norm $\|\boldsymbol{B}_j\|_2$ aggregates these output-specific parameters and acts as a feature-level gate shared by the linear and nonlinear branches. The resulting CORAL-Net is defined as follows:

$$f(\mathbf{x}; \boldsymbol{B}, \Theta) = \mathbf{x}^\top \boldsymbol{B} + h_\Theta(\mathbf{x} \circ (\|\boldsymbol{B}_1\|_2, \ldots, \|\boldsymbol{B}_p\|_2)) .$$

After training, we use $\|\boldsymbol{B}_j\|_2$ for feature selection. If this norm is approximately zero, the jth feature is removed simultaneously from both the linear and nonlinear branches. We therefore select $\widehat{\mathcal{A}} \leftarrow \{j \mid \|\boldsymbol{B}_j\|_2 > \delta\}$.

4.4 Analytical comparison with related methods

We conclude the methodology section by comparing CORAL-Net with three structurally related neural-network architectures.

- *Input-gating models.* As discussed above, the gate-only formulation $h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$ does not identify $\boldsymbol{\beta}$ and leads to less stable feature selection.
- *Partial linear models.* Lemhadri et al. (2021) considers $\mathbf{x}^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x})$ and enforces feature selection by penalizing $\|\boldsymbol{\beta}\|_1$ and constraining $\|\Theta_j^{(1)}\|_\infty \leq M|\beta_j|$ (for $j = 1, \ldots, p$). This formulation involves additional optimization effort and two tuning parameters, λ and M. In contrast, our method can be trained with a standard optimizer and selects features via a simple trimming step.
- *Single-index models.* The architecture $\mathbf{x}^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x}^\top \boldsymbol{\beta})$ is a single-index model (Li, 1991; Alquier & Biau, 2013; Zweig et al., 2023). It restricts nonlinear effects to the scalar projection $\mathbf{x}^\top \boldsymbol{\beta}$, whereas CORAL-Net passes the p-dimensional gated input to the nonlinear backbone.

5 Experiments

We conduct extensive experiments to evaluate CORAL. Our empirical study is designed to answer the following questions:

1. *Does our method help steady feature selection compared with $h(\mathbf{x} \circ \boldsymbol{\beta})$?*
2. *What is the numerical performance of the proposal compared with the existing methods?*
3. *Can CORAL steady existing neural-network-based feature selection?*

We study the first question using synthetic experiments and 21 real-world datasets (see Sections 5.1–5.2). The studies on the second and third questions are provided in Section 5.3 and Section 5.4, respectively. To save space, the implementation details are deferred to Appendix C.

5.1 SYNTHETIC EXPERIMENTS

To assess the steadying effect when the true support is known, we compare three neural network architectures: $\mathbf{x}^\top\boldsymbol{\beta}$, $h(\mathbf{x} \circ \boldsymbol{\beta})$, and CORAL-Net. The comparison considers support recovery, its variation across independent runs, and generalization error. The Friedman dataset contains five informative and five nuisance features, whereas the Artificial dataset contains three informative and seven nuisance features. Details on dataset generation are provided in Appendix C.1. For both datasets, we conduct 100 independent experiments.

Table 1: Results of the ablation study on synthetic datasets. Values in parentheses represent the standard deviation.

	Friedman			Artificial		
	MSE	Recall	Precision	MSE	Recall	Precision
CORAL-Net	**0.06 (0.01)**	**1.00(0.00)**	**1.00(0.00)**	**0.09(0.02)**	**1.00(0.00)**	**1.00 (0.00)**
$\mathbf{x}^\top\boldsymbol{\beta}$	1.36(0.00)	0.79(0.00)	0.79(0.00)	1.99(0.00)	**1.00(0.00)**	**1.00(0.00)**
$h(\mathbf{x} \circ \boldsymbol{\beta})$	**0.06(0.02)**	**1.00(0.00)**	**1.00(0.00)**	0.16(0.13)	0.89(0.18)	0.89(0.18)

Table 1 reports both selection quality and its variation across experiments. CORAL-Net attains recall and precision of 1.00 (0.00) on both datasets, indicating accurate support recovery with little reported variation. In contrast, $\mathbf{x}^\top\boldsymbol{\beta}$ misses informative features on the Friedman dataset, while $h(\mathbf{x} \circ \boldsymbol{\beta})$ reports recall and precision of 0.89 (0.18) on the Artificial dataset. Together with the tighter coefficient distributions over 50 replications in Figure 2, these results are consistent with a steadying effect of the coupled residual connection in this controlled setting. CORAL-Net also achieves the lowest or tied-lowest MSE.

5.2 EFFECT OF CORAL ON STEADYING REAL-WORLD PERFORMANCE

We next evaluate CORAL on 12 real-world datasets, including 9 classification and 3 regression datasets, collected from the ASU variable selection database (Li et al., 2018) and OpenML (Van Rijn et al., 2013). Their characteristics are summarized in Tables A1 and A2. For classification, the sample sizes range from $O(10^4)$ to $O(10^5)$ and the numbers of features range from tens to thousands. For regression, the sample sizes range from $O(10^4)$ to $O(10^7)$, with roughly one hundred features. We additionally consider 9 large-scale classification datasets from the Deep tabular benchmark (Cherepanova et al., 2023), with up to $O(10^7)$ observations; these results are deferred to Appendix D.2. Because the true relevant sets are unavailable for these datasets, we evaluate downstream predictive performance after feature selection. The selected features are used to retrain a neural network, and we report accuracy for classification and negative RMSE for regression on a held-out test set.

We compare CORAL-MLP and CORAL-MoE with their uncoupled MLP and MoE counterparts at 50 selected features. Figure 5 shows that each CORAL variant improves mean accuracy on eight of the nine classification datasets and reduces mean RMSE on all three regression datasets. Furthermore, both CORAL-MLP and CORAL-MoE have shorter standard-deviation error bars than their uncoupled counterparts on all nine classification and all three regression datasets. In summary, the coupled connection improves downstream prediction on 11 of the 12 datasets while narrowing the error bars on all 12. This effectiveness extends to nine additional large-scale classification datasets (see Figure A1). On all of these datasets, the error bars are narrowed together with an improvement in prediction.

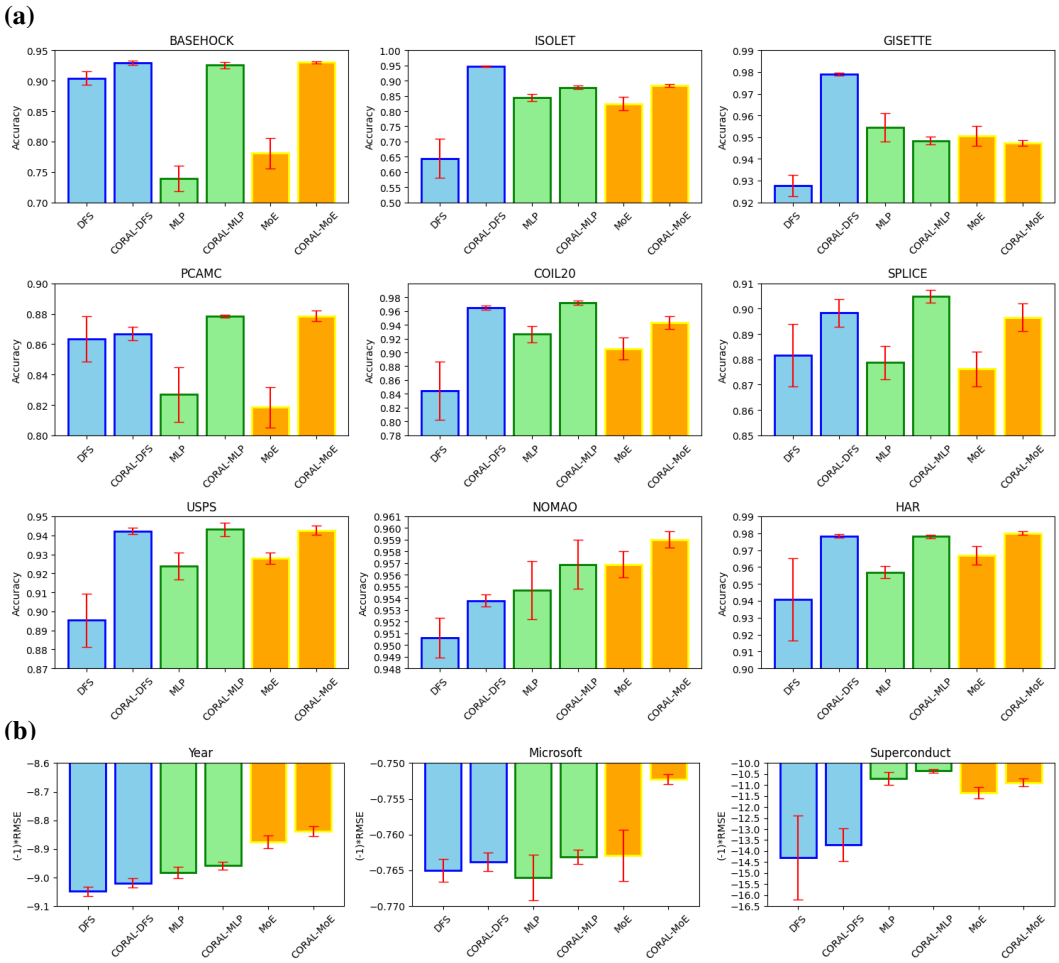

Figure 5: Error bar charts with and without the coupled residual connection. (a) Classification accuracy on nine real-world datasets, with 50 selected variables and five independent experiments for each dataset. (b) Regression performance on three regression datasets. Bar length indicates mean performance, with red segments representing the standard deviation.

5.3 Comparison to existing methods

In this part, we compare the proposal with four representative baselines: Lasso, LassoNet, stochastic gates (STG), and deep feature selection (DFS). All deep-learning methods use the same MLP backbone with the same selected-feature budget, ranging from 10 to 50. The results on the 12 datasets considered in Section 5.2 are summarized in Figure 6. Generally, CORAL-MLP reduces classification error relative to each of the four baselines on all nine classification datasets and reduces RMSE relative to each baseline on all three regression datasets. Averaged across datasets, the classification-error reduction relative to the baselines ranges from 39.6% to 51.6% as can be seen from Table A4, while the RMSE reduction ranges from 9.1% to 17.2% (see Table A5). CORAL-MoE performs comparably to CORAL-MLP, suggesting that the observed benefit is not specific to the MLP backbone.

The advantage of nonlinear methods over Lasso is expected, as many real-world relationships exhibit complex nonlinear patterns. CORAL differs from LassoNet through its use of a standard optimizer and a simple trimming rule, whereas LassoNet uses additional regularization and constrained optimization, as discussed in Section 4.4. As for DFS and STG, they essentially select features based on the network $h(\mathbf{x} \circ \boldsymbol{\beta})$. The controlled results in Figure 2 and Table 1 provide clear evidence that, without the coupled connection, feature selection may be unstable, which may degrade their predictive performance. Appendix D.2 shows that these predictive gains from CORAL persist on 9 additional large-scale datasets (Figure A2).

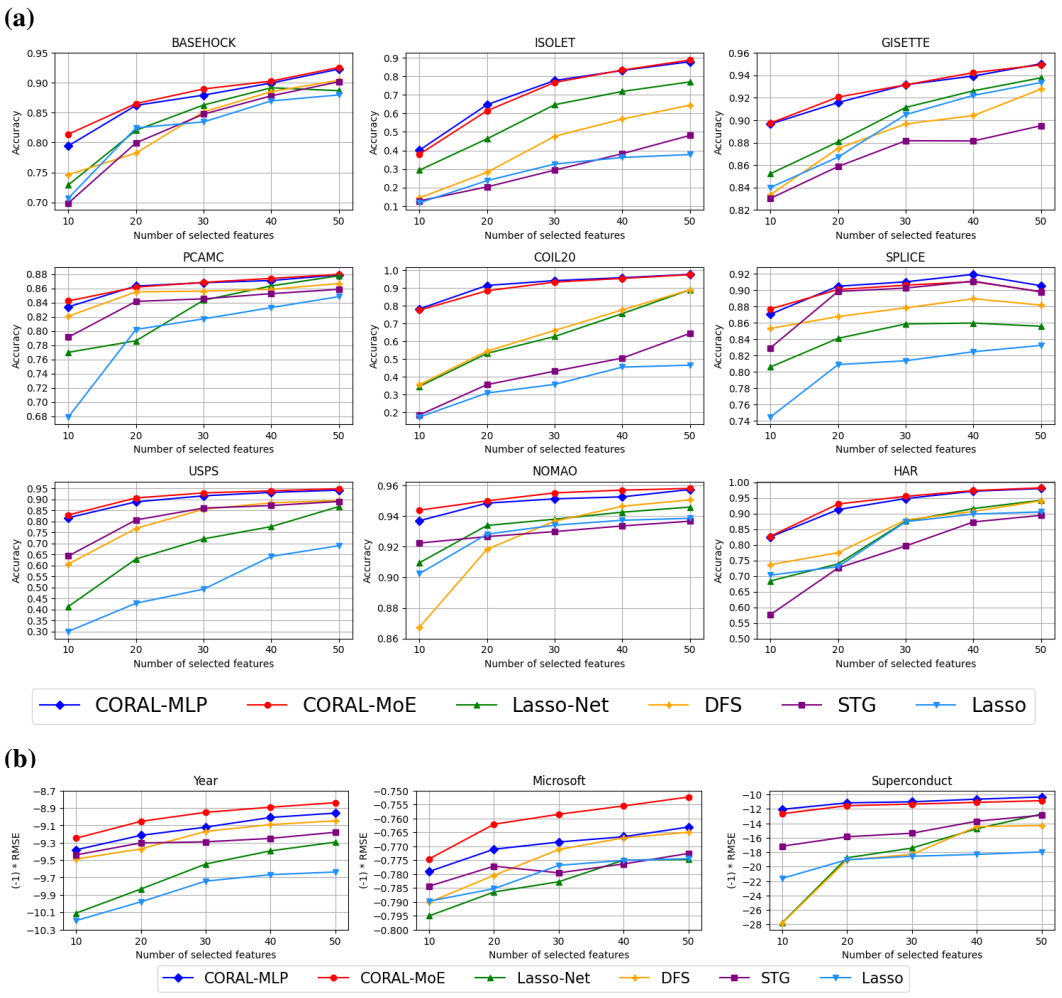

Figure 6: Performance comparison across real-world datasets. (a) Classification accuracy versus the number of selected variables across nine classification datasets. (b) Negative RMSE versus the number of selected features across three regression datasets. Each panel corresponds to one dataset.

5.4 Combining CORAL with an existing feature selection method

To examine whether CORAL can improve an existing selector, we add the coupled residual connection to DFS, yielding CORAL-DFS. Figure 5 shows that CORAL-DFS improves mean predictive performance over DFS on all 12 datasets. Its error bars are also shorter than those of DFS on 11 of the 12 datasets. This reflects the potential of CORAL to boost existing neural-network-based feature selection.

6 Conclusions and Discussion

In this paper, we study how to steady feature selection in neural networks. We address this problem by introducing a coupled residual connection with an identifiability guarantee on parameters for feature selection. The resulting neural feature-selection algorithm effectively steadies feature selection and achieves strong performance across the evaluated synthetic and real-world benchmarks. Determining the threshold δ via cross-validation remains challenging when the magnitude of signals is weak. A promising direction is to leverage cross-validation to guide adaptive selection strategies, supported by recent theoretical advances (Bayle et al., 2020).

AI USE STATEMENT

In preparing this paper, LLMs were used only to aid and polish the writing and to retrieve and discover related work, which did not affect the core methodology. The authors manually checked all cited references to confirm that the cited works exist.

ETHICS STATEMENT

This work uses only synthetic and existing public benchmark datasets and involves no participant recruitment or new collection of personal data. Because selected features may reflect biases in the data or be causally misinterpreted, scientific and high-stakes applications require domain-specific validation and appropriate oversight.

REPRODUCIBILITY STATEMENT

The main text specifies the model, optimization objective, algorithm and evaluation protocols, while the appendix provides the proofs, dataset summaries and additional details on experiments.

REFERENCES

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4699–4711, 2021.

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In Emanuel Parzen, Kunio Tanabe, and Genshiro Kitagawa (eds.), *Selected Papers of Hirotugu Akaike*, Springer Series in Statistics, pp. 199–213. Springer, 1998. doi: 10.1007/978-1-4612-1694-0_15.

Francesca Albertini, Eduardo D Sontag, and Vincent Maillot. Uniqueness of weights for neural networks. *Artificial neural networks for speech and vision*, (4), 1993.

Pierre Alquier and Gérard Biau. Sparse single-index model. *Journal of Machine Learning Research*, 14(1), 2013.

Zahra Atashgahi, Tennison Liu, Mykola Pechenizkiy, Raymond Veldhuis, Decebal Constantin Mocanu, and Mihaela van der Schaar. Unveiling the power of sparse neural networks for feature selection. In *ECAI 2024: 27th European Conference on Artificial Intelligence*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pp. 2669–2676. IOS Press, 2024. doi: 10.3233/FAIA240799.

Ray Bai, Gemma E Moran, Joseph L Antonelli, Yong Chen, and Mary R Boland. Spike-and-slab group lassos for grouped regression and sparse generalized additive models. *Journal of the American Statistical Association*, 117(537):184–197, 2022. doi: 10.1080/01621459.2020.1765784.

Stephen Bates, Emmanuel Candès, Lucas Janson, and Wenshuo Wang. Metropolized knockoff sampling. *Journal of the American Statistical Association*, 116(535):1413–1427, 2021.

Pierre Bayle, Alexandre Bayle, Lucas Janson, and Lester Mackey. Cross-validation confidence intervals for test error. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16339–16350. Curran Associates, Inc., 2020.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *Annals of Statistics*, 44(2):813–852, 04 2016. doi: 10.1214/15-AOS1388.

Joachim Bona-Pellissier, François Malgouyres, and François Bachoc. Local identifiability of deep ReLU neural networks: the theory. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27549–27562, 2022.

Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. Parameter identifiability of a deep feedforward ReLU neural network. *Machine Learning*, 112(11):4431–4493, 2023.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Phuong Bui Thi Mai and Christoph Lampert. Functional vs. parametric equivalence of ReLU networks. In *International Conference on Learning Representations*, 2020.

Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Jianbo Chen, Mitchell Stern, Martin J Wainwright, and Michael I Jordan. Kernel feature selection via conditional covariance minimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Xingshu Chen, Yifeng Guo, and Jin-Hong Du. Flow-disentangled feature importance. In *International Conference on Learning Representations*, pp. 57346–57388, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5e2217482fa75556f1970be809acd3f8-Paper-Conference.pdf.

Yao Chen, Qingyi Gao, Faming Liang, and Xiao Wang. Nonlinear variable selection via deep neural networks. *Journal of Computational and Graphical Statistics*, 30(2):484–492, 2021. doi: 10.1080/10618600.2020.1814305.

Valeriia Cherepanova, Roman Levin, Gowthami Somepalli, Jonas Geiping, C. Bayan Bruss, Andrew Gordon Wilson, Tom Goldstein, and Micah Goldblum. A performance-driven benchmark for feature selection in tabular deep learning. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=v4PMCdSaAT.

Héctor Climente-González, Chloé-Agathe Azencott, Samuel Kaski, and Makoto Yamada. Block HSIC lasso: model-free biomarker detection for ultra-high dimensional data. *Bioinformatics*, 35 (14):i427–i435, 2019.

David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

Wenying Deng, Beau Coker, Rajarshi Mukherjee, Jeremiah Liu, and Brent Coull. Towards a unified framework for uncertainty-aware nonlinear variable selection with theoretical guarantees. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27636–27651, 2022.

Vu C Dinh and Lam S Ho. Consistent feature selection for analytic deep neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2420–2431, 2020.

Yihe Dong and Sercan O. Arik. SLM: End-to-end feature selection via sparse learnable masks. *arXiv preprint arXiv:2304.03202*, 2023. URL https://arxiv.org/abs/2304.03202.

Jin-Hong Du, Kathryn Roeder, and Larry Wasserman. Disentangled feature importance. *arXiv preprint arXiv:2507.00260*, 2025.

Robert F Engle, Clive WJ Granger, John Rice, and Andrew Weiss. Semiparametric estimates of the relation between weather and electricity sales. *Journal of the American Statistical Association*, 81 (394):310–320, 1986.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. doi: 10.1198/016214501753382273.

Charles Fefferman. Reconstructing a neural net from its output. *Revista Matemática Iberoamericana*, 10(3):507–555, 1994. doi: 10.4171/RMI/160.

Jean Feng and Noah Simon. Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*, 2017.

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

Xiaoling Gong, Ling Yu, Jian Wang, Kai Zhang, Xiao Bai, and Nikhil R. Pal. Unsupervised feature selection via adaptive autoencoder with redundancy control. *Neural Networks*, 150:87–101, 2022. doi: 10.1016/j.neunet.2022.03.004.

Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. Hidden symmetries of ReLU networks. In *International Conference on Machine Learning*, pp. 11734–11760. PMLR, 2023.

Zhiyuan Han, An Chen, Zejian Li, et al. Machine learning-based design of electrocatalytic materials towards high-energy lithium–sulfur batteries development. *Nature Communications*, 15:8433, 2024. doi: 10.1038/s41467-024-52550-9.

Amin Ul Haq, Jianping Li, Muhammad Hammad Memon, Muhammad Hunain Memon, Jalaluddin Khan, and Syeda Munazza Marium. Heart disease prediction system using model of machine learning and sequential backward selection algorithm for features selection. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1–4, 2019. doi: 10.1109/I2CT45611.2019.9033683.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Julien Hédou, Ivana Marić, Grégoire Bellan, et al. Discovery of sparse, reliable omic biomarkers with Stabl. *Nature Biotechnology*, 42:1581–1593, 2024. doi: 10.1038/s41587-023-02033-x.

Ronald R Hocking and RN Leslie. Selection of the best subset in regression analysis. *Technometrics*, 9(4):531–540, 1967.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. doi: 10.1016/0893-6080(89)90020-8.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Jian Huang, Joel L Horowitz, and Fengrong Wei. Variable selection in nonparametric additive models. *Annals of Statistics*, 38(4):2282, 2010. doi: 10.1214/09-aos781.

Jian Huang, Yuling Jiao, Yanyan Liu, and Xiliang Lu. A constructive approach to ℓ_0 penalized regression. *Journal of Machine Learning Research*, 19(10):1–37, 2018.

Bertrand Iooss and Clémentine Prieur. Shapley effects for sensitivity analysis with correlated inputs: Comparisons with Sobol' indices, numerical estimation and applications. *International Journal for Uncertainty Quantification*, 9(5):493–514, 2019. doi: 10.1615/Int.J.UncertaintyQuantification.2019028372.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Xiangjian Jiang, Andrei Margeloiu, Nikola Simidjievski, and Mateja Jamnik. ProtoGate: Prototype-based neural networks with global-to-local feature selection for tabular biomedical data. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21844–21878. PMLR, 2024.

Paul C Kainen, Vera Kurková, Vladik Kreinovich, and Ongard Sirisaengtaksin. Uniqueness of network parametrization and faster learning. *Neural Parallel Sci. Comput.*, 2(4):459–466, 1994.

Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on Learning Theory*, pp. 2306–2327. PMLR, 2020.

Věra Kůrková and Paul C Kainen. Functionally equivalent feedforward neural networks. *Neural Computation*, 6(3):543–558, 1994.

Changhee Lee, Fergus Imrie, and Mihaela van der Schaar. Self-supervision enhanced feature selection with correlated gates. In *International Conference on Learning Representations*, 2021.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018.

Ismael Lemhadri, Feng Ruan, Louis Abraham, and Robert Tibshirani. LassoNet: A neural network with feature sparsity. *Journal of Machine Learning Research*, 22(127):1–29, 2021.

Chun-Na Li, Ling-Wei Huang, Yuan-Hai Shao, Tingting Guo, and Yu Mao. Feature selection by Universum embedding. *Pattern Recognition*, 153:110514, 2024. doi: 10.1016/j.patcog.2024.110514.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys*, 50(6):94, 2018.

Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

Huming Liao, Hongmei Chen, Yong Mi, Chuan Luo, Shi-Jinn Horng, and Tianrui Li. Sparse orthogonal supervised feature selection with global redundancy minimization, label scaling, and robustness. *Information Sciences*, 666:120454, 2024. doi: 10.1016/j.ins.2024.120454.

Kaiting Liu, Zahra Atashgahi, Ghada Sokar, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Supervised feature selection via ensemble gradient information from sparse neural networks. In *International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3952–3960. PMLR, 2024.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 4765–4774. Curran Associates, Inc., 2017.

Bin Luo and Susan Halabi. Sparse-input neural network using group concave regularization. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=m9UsLHZYeX.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. G. Cortina-Januchs, and D. Andina. Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 2845–2850, 2010. doi: 10.1109/IECON.2010.5675075.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*, pp. 807–814, 2010.

Pedram Pad, Hadi Hammoud, Mohamad Dia, Nadim Maamari, and Liza Andrea Dunbar. SAND: One-shot feature selection with additive noise distortion. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47443–47455. PMLR, 2025. URL https://proceedings.mlr.press/v267/pad25a.html.

Hyeryn Park and Changhee Lee. Feature selection with group-sparse stochastic gates. *IEEE Access*, 12:102299–102312, 2024. doi: 10.1109/ACCESS.2024.3432509.

Sejun Park, Chulhee Yun, Jaeho Lee, and Jinwoo Shin. Minimum width for universal approximation. In *International Conference on Learning Representations*, 2021.

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

Henning Petzka, Martin Trimmel, and Cristian Sminchisescu. Notes on the symmetries of 2-layer ReLU-networks. In *Proceedings of the Northern Lights Deep Learning Workshop*, volume 1, pp. 6–6, 2020.

Benjamin Poignard and Makoto Yamada. Sparse hilbert-schmidt independence criterion regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 538–548. PMLR, 2020.

Zhimei Ren, Yuting Wei, and Emmanuel Candès. Derandomizing knockoffs. *Journal of the American Statistical Association*, 118(542):948–958, 2023.

David Rolnick and Konrad Kording. Reverse-engineering deep ReLU networks. In *International Conference on Machine Learning*, pp. 8178–8187. PMLR, 2020.

Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872, 2020.

Aytijhya Saha and Nikhil R. Pal. Group-feature (sensor) selection with controlled redundancy using neural networks. *Neurocomputing*, 610:128596, 2024. doi: 10.1016/j.neucom.2024.128596.

Johannes Sandberg, Thomas Voigtmann, Emilie Devijver, and Noel Jakse. Feature selection for high-dimensional neural network potentials with the adaptive group lasso. *Machine Learning: Science and Technology*, 5:025043, 2024. doi: 10.1088/2632-2153/ad450e.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, 48(4):1875–1897, 2020. doi: 10.1214/19-aos1875.

Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

Xiaotong Shen, Wei Pan, Yunzhang Zhu, and Hui Zhou. On constrained and regularized high-dimensional regression. *Annals of the Institute of Statistical Mathematics*, 65(5):807–832, 2013. doi: 10.1007/s10463-012-0396-3.

Pierre Stock and Rémi Gribonval. An embedding of ReLU networks and an analysis of their identifiability. *Constructive Approximation*, 57(2):853–899, 2023.

Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9:307, 2008. doi: 10.1186/1471-2105-9-307.

Héctor J Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 5(4):589–593, 1992.

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

Jan N Van Rijn, Bernd Bischl, Luis Torgo, Bo Gao, Venkatesh Umaashankar, Simon Fischer, Patrick Winter, Bernd Wiswedel, Michael R Berthold, and Joaquin Vanschoren. OpenML: A collaborative science platform. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pp. 645–649. Springer, 2013. doi: 10.1007/978-3-642-40994-3_46.

Isabella Verdinelli and Larry Wasserman. Decorrelated variable importance. *Journal of Machine Learning Research*, 25(7):1–27, 2024.

Jian Wang, Huaqing Zhang, Junze Wang, Yifei Pu, and Nikhil R. Pal. Feature selection using a neural network with group lasso regularization and controlled redundancy. *IEEE Transactions on Neural Networks and Learning Systems*, 32(3):1110–1123, 2021. doi: 10.1109/TNNLS.2020.2980383.

Brian Williamson and Jean Feng. Efficient nonparametric statistical inference on population feature importance using Shapley values. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10282–10291. PMLR, 2020.

Brian D. Williamson, Peter B. Gilbert, Marco Carone, and Noah Simon. Nonparametric variable importance assessment using machine learning techniques. *Biometrics*, 77(1):9–22, 2021. doi: 10.1111/biom.13392.

Maksymilian Wojtas and Ke Chen. Feature importance ranking for deep learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5105–5114, 2020.

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In Hal Daumé III and Aarti Singh (eds.), *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10648–10659. PMLR, 13–18 Jul 2020.

Junchen Yang, Ofir Lindenbaum, and Yuval Kluger. Locally sparse neural networks for tabular biomedical data. In *International Conference on Machine Learning*, pp. 25123–25153. PMLR, 2022.

Kaixu Yang, Arkaprabha Ganguli, and Tapabrata Maiti. ENNS: Variable selection, regression, classification, and deep neural network for high-dimensional data. *Journal of Machine Learning Research*, 25(335):1–45, 2024. URL `http://jmlr.org/papers/v25/21-0893.html`.

Lei Yang, Shaogao Lv, and Junhui Wang. Model-free variable selection in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 17(1):2885–2908, 2016.

Taisuke Yasuda, MohammadHossein Bateni, Lin Chen, Matthew Fahrbach, Gang Fu, and Vahab Mirrokni. Sequential attention for feature selection. In *International Conference on Learning Representations*, 2023.

Chaeyun Yeo, Namjoon Suh, and Younghoon Kim. Fused LassoNet: Sequential feature selection for spectral data with neural networks. *Chemometrics and Intelligent Laboratory Systems*, 257: 105315, 2025. doi: 10.1016/j.chemolab.2024.105315.

Ming Yuan and Yi Lin. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 12 2005. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2005.00532.x.

Xiaotong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18(166):1–43, 2018.

Junxian Zhu, Canhong Wen, Jin Zhu, Heping Zhang, and Xueqin Wang. A polynomial algorithm for best-subset selection problem. *Proceedings of the National Academy of Sciences*, 117(52): 33117–33123, 2020.

Zifan Zhu, Yingying Fan, Yinfei Kong, Jinchi Lv, and Fengzhu Sun. DeepLINK: Deep learning inference using knockoffs with applications to genomics. *Proceedings of the National Academy of Sciences*, 118(36):e2104683118, 2021.

Aaron Zweig, Loucas Pillaud-Vivien, and Joan Bruna. On single-index models beyond Gaussian data. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Additional related work

Feature selection. Feature selection has been studied extensively for both linear and nonlinear models. For linear models, relaxation-based methods include Lasso and its variants (Tibshirani, 1996; Fan & Li, 2001; Candes & Tao, 2005; Shen et al., 2013). Model-selection criteria and discrete optimization include the Akaike and Bayesian information criteria, enumeration and mixed-integer programming (Akaike, 1998; Schwarz, 1978; Hocking & Leslie, 1967; Bertsimas et al., 2016; Gómez & Prokopyev, 2021), while iterative procedures include forward–backward greedy algorithms and hard thresholding (Yuan et al., 2018; Huang et al., 2018; Zhu et al., 2020). Nonlinear feature selection includes model-based approaches such as generalized additive and neural additive models (Huang et al., 2010; Bai et al., 2022; Agarwal et al., 2021), kernel dependence methods (Yang et al., 2016; Chen et al., 2017; Climente-González et al., 2019; Poignard & Yamada, 2020), knockoff procedures designed to control false discoveries (Romano et al., 2020; Bates et al., 2021; Zhu et al., 2021; Ren et al., 2023) and feature-importance methods based on fitted predictors such as random forests or SHAP (Breiman, 2001; Lundberg & Lee, 2017; Wojtas & Chen, 2020; Deng et al., 2022). CORAL differs from these broad feature-selection frameworks in its target, since it is an architecture for training feature coefficients jointly with a nonlinear predictor rather than a structural regression model, a false-discovery-control procedure or a post-hoc importance measure.

Feature importance. Model-agnostic feature importance methods quantify a predictor's contribution by perturbing, removing or reallocating its predictive information. Permutation and leave-one-covariate-out methods assess importance through changes in predictive risk induced by shuffling or excluding a feature (Breiman, 2001; Lei et al., 2018; Williamson et al., 2021), whereas Shapley-based approaches allocate predictive value across feature subsets (Lundberg & Lee, 2017; Williamson & Feng, 2020). A central complication is predictor dependence, under which marginal perturbations or allocations may not correspond to meaningful feature interventions. This has motivated conditional permutation schemes and, more recently, dependence-aware formulations based on Shapley effects, decorrelation and latent-space disentanglement (Strobl et al., 2008; Iooss & Prieur, 2019; Verdinelli & Wasserman, 2024; Du et al., 2025; Chen et al., 2026).

B Proofs

B.1 Proof of Proposition 1

To prove Proposition 1 formally, we first restate it in an equivalent and mathematically precise form.

Proposition (Restatement of Proposition 1)**.** *Suppose $h_\Theta(\mathbf{x})$, whose parameters Θ consist of the layer parameters $\Theta^{(1)}, \dots, \Theta^{(H+1)}$ and an output bias b, is a neural network with H hidden layers of the form*

$$h_\Theta(\mathbf{x}) = b + \sum_{i=1}^{k} \Theta_i^{(H+1)} \sigma\Big(h_i'(\mathbf{x}; \Theta^{(1)}, \dots, \Theta^{(H)})\Big),$$

where k is the number of units in the last hidden layer, $\sigma(\cdot)$ is a bounded activation function, $\Theta_i^{(H+1)}$ is the output weight of the ith unit and h_i' is the pre-activation of that unit, computed by the first H layers with arbitrary activations. If $\boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$, then

$$\mathbf{x}^\top \boldsymbol{\beta}_1 + h_{\Theta_1}(\mathbf{x} \circ \boldsymbol{\beta}_1) \neq \mathbf{x}^\top \boldsymbol{\beta}_2 + h_{\Theta_2}(\mathbf{x} \circ \boldsymbol{\beta}_2)$$

as functions of $\mathbf{x}$, for any two network parameters Θ_1 and Θ_2.

Proof. We prove the result by contradiction. Write $f(\mathbf{x}; \boldsymbol{\beta}, \Theta) = \mathbf{x}^\top \boldsymbol{\beta} + h_\Theta(\mathbf{x} \circ \boldsymbol{\beta})$ and suppose there exist $(\boldsymbol{\beta}_1, \Theta_1)$ and $(\boldsymbol{\beta}_2, \Theta_2)$ with $\boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$ and

$$f(\mathbf{x}; \boldsymbol{\beta}_1, \Theta_1) = f(\mathbf{x}; \boldsymbol{\beta}_2, \Theta_2)$$

for every $\mathbf{x}$. For $r = 1, 2$, write

$$f(\mathbf{x}; \boldsymbol{\beta}_r, \Theta_r) = \mathbf{x}^\top \boldsymbol{\beta}_r + b_r + \sum_{i=1}^{k} \Theta_{ri}^{(H+1)} \sigma\Big(h_i'(\mathbf{x} \circ \boldsymbol{\beta}_r; \Theta_r^{(1)}, \dots, \Theta_r^{(H)})\Big).$$

It follows that

$$\mathbf{x}^\top(\boldsymbol{\beta}_2 - \boldsymbol{\beta}_1) = b_1 - b_2 + \sum_{i=1}^{k} \Theta_{1i}^{(H+1)} \sigma\Big(h_i'(\mathbf{x} \circ \boldsymbol{\beta}_1; \Theta_1^{(1)}, \ldots, \Theta_1^{(H)})\Big) - \sum_{i=1}^{k} \Theta_{2i}^{(H+1)} \sigma\Big(h_i'(\mathbf{x} \circ \boldsymbol{\beta}_2; \Theta_2^{(1)}, \ldots, \Theta_2^{(H)})\Big).$$

Without loss of generality, assume $(\boldsymbol{\beta}_1)_1 \neq (\boldsymbol{\beta}_2)_1$ and let $\mathbf{x} = (x_1, 0, \ldots, 0)^\top$. As $|x_1| \to \infty$, the absolute value of the left-hand side diverges, whereas the right-hand side remains bounded because σ is bounded and k is finite, which contradicts the identity above. Therefore, CORAL-Net is identifiable with respect to $\boldsymbol{\beta}$, which proves Proposition 1. □

B.2 Proof of Proposition 2

Proof. Recall that a class of functions on $\mathbb{R}^p$ has the universal approximation property if, for every compact set $K \subset \mathbb{R}^p$, it is dense in $C(K)$ under the supremum norm. Given such a set K, a function $g^\star \in C(K)$ and $\epsilon > 0$, we set $\boldsymbol{\beta}^\star = \mathbf{1}_p = (1, \ldots, 1)^\top$, so that it suffices to find $\Theta^\star$ such that

$$\sup_{\mathbf{x} \in K} \left| \big(g^\star(\mathbf{x}) - \mathbf{x}^\top \mathbf{1}_p\big) - h_{\Theta^\star}(\mathbf{x}) \right| \leq \epsilon.$$

Let $g(\mathbf{x}) = g^\star(\mathbf{x}) - \mathbf{x}^\top \mathbf{1}_p$. Since $g \in C(K)$ and $\{h_\Theta : \Theta\}$ has the universal approximation property, there exists $\Theta^\star$ such that

$$\sup_{\mathbf{x} \in K} |g(\mathbf{x}) - h_{\Theta^\star}(\mathbf{x})| \leq \epsilon.$$

Because $\mathbf{x} \circ \boldsymbol{\beta}^\star = \mathbf{x}$, this is equivalent to

$$\sup_{\mathbf{x} \in K} \left| g^\star(\mathbf{x}) - \mathbf{x}^\top \boldsymbol{\beta}^\star - h_{\Theta^\star}(\mathbf{x} \circ \boldsymbol{\beta}^\star) \right| \leq \epsilon,$$

which proves the proposition. □

C Experimental details

C.1 Generation of synthetic datasets

Friedman dataset. We generate the n-by-p Friedman dataset with the `sklearn.datasets.make_friedman1` function in `scikit-learn`, so that the first five of the p features are informative and the remaining $p - 5$ features are nuisance features.

Artificial dataset. We create the n-by-p Artificial dataset by following three consecutive steps.

1. Generate n independent samples from the p-dimensional standard normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$.
2. Let the first three features X_1, X_2, X_3 be informative features and the remaining $p - 3$ features be nuisance features.
3. Let $Y = X_1 + \arctan(X_2) + \exp(X_3)$.

C.2 Basic information on real-world datasets

Table A1: Basic information on the nine classification datasets used for empirical evaluation, including the number of features, sample size, and number of classes.

	BASEHOCK	COIL20	GISETTE	HAR	ISOLET	NOMAO	PCMAC	SPLICE	USPS
n	1993	1440	7000	10299	1560	34465	1943	3190	9298
p	4862	1024	5000	561	617	118	3289	60	256
Classes	2	20	2	6	26	2	2	3	10

Table A2: Basic information on the three regression datasets.

	Year	Microsoft	Superconduct
n	515345	1200192	21263
p	90	136	81

Table A3: Basic information for the nine large-scale classification datasets used in the empirical evaluation. Cov., CPE, Otto and Vehicle abbreviate Covertype, CovPokElec, Otto-Group-Product and Vehicle_sensIT.

Dataset	ALOI	Cov.	CPE	Epsilon	Jannis	Meta	MNIST	Otto	Vehicle
p	128	54	72	2,000	54	74	784	93	100
n	108,000	581,012	1,455,525	500,000	83,733	45,164	70,000	61,878	98,528
Classes	1,000	7	10	2	4	11	10	9	2

C.3 Evaluation Criteria

Predictive performance is measured by the root mean squared error of prediction in regression tasks and by accuracy in classification tasks. The relative classification-error reduction is computed as $\frac{\text{Accuracy}_{\text{CORAL}} - \text{Accuracy}_{\text{Method}}}{1 - \text{Accuracy}_{\text{Method}}} \times 100\%$. Relative RMSE improvement is computed as $\frac{\text{RMSE}_{\text{Method}} - \text{RMSE}_{\text{CORAL}}}{\text{RMSE}_{\text{Method}}} \times 100\%$.

C.4 Implementation of baselines

We run the baselines with the code provided by their authors, which is available from the following sources.

- **DFS**: https://github.com/cyustcer/Deep-Feature-Selection.
- **LassoNet**: https://github.com/lasso-net/lassonet.
- **STG**: https://github.com/runopti/stg.
- **Lasso**'s implementation is provided by `scikit-learn` (Pedregosa et al., 2011).

All methods use identical train/validation/test splits. For methods permitting a configurable MLP backbone, we use the same architecture. Method-specific training epochs, learning rates, and optimizers follow the respective default configurations.

C.5 Computational devices for experiments

All experiments were conducted on an AMD EPYC 9654 96-core CPU and an NVIDIA Tesla P100 GPU.

Table A4: Average relative classification-error reduction (%) of CORAL-MLP over competing methods on nine classification datasets.

Method	BASEHOCK	ISOLET	GISETTE	PCMAC	COIL20	SPLICE	USPS	NOMAO	HAR	Avg.
Lasso	29.40	60.72	30.10	30.44	87.19	48.65	81.28	31.29	65.50	51.62
LassoNet	21.91	34.05	24.72	17.84	77.42	36.01	69.00	25.04	62.18	40.91
DFS	23.49	52.41	35.05	8.49	76.52	21.00	50.05	29.81	59.62	39.60
STG	26.64	60.36	45.16	15.73	86.01	8.87	48.48	30.52	73.48	43.92

Table A5: Average relative RMSE improvement (%) of CORAL over competing methods on three regression datasets.

Method	Year	Microsoft	Superconduct	Avg.
Lasso	8.07	1.60	42.19	17.29
LassoNet	6.04	1.73	37.65	15.14
DFS	1.99	0.89	24.52	9.13
STG	2.62	1.31	27.41	10.45

D Additional Experimental Results

D.1 Relative improvement over baselines

D.2 Additional results on large-scale classification tasks

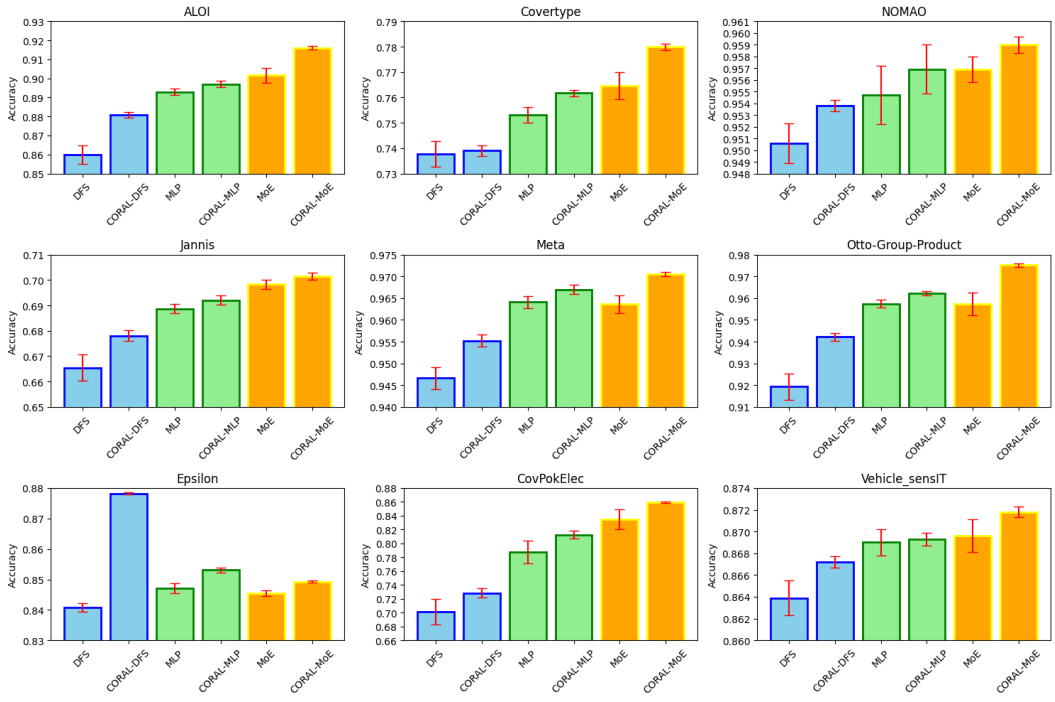

Figure A1: Error bar chart showing classification accuracy on nine large-scale classification datasets with and without the coupled residual connection. The end of each bar marks the mean accuracy on a truncated axis, while red segments show one standard deviation.

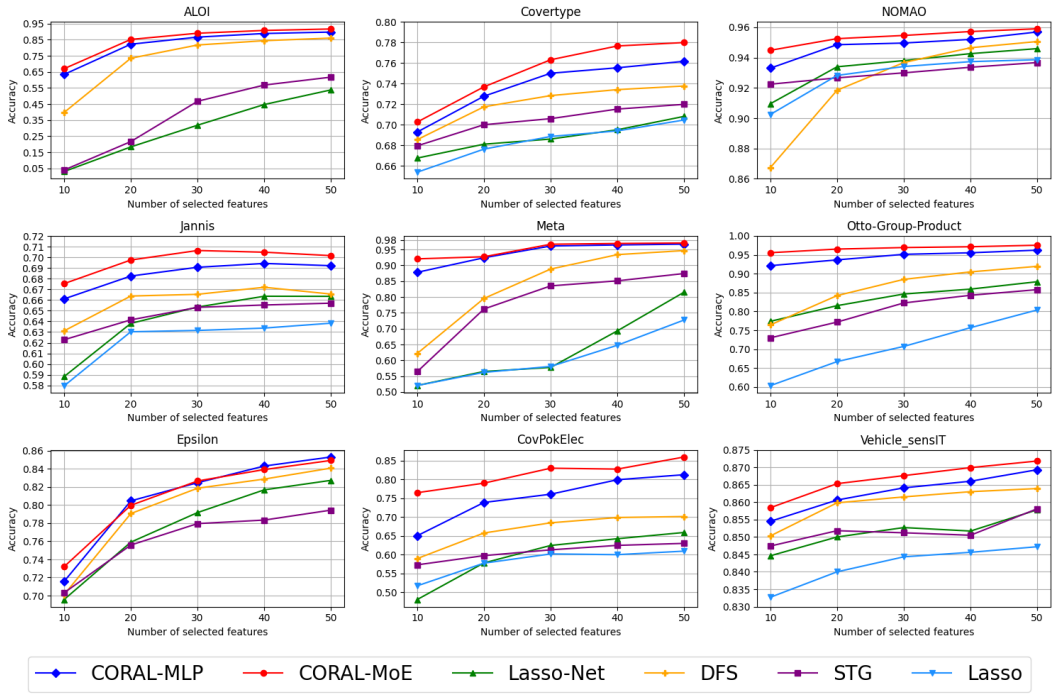

Figure A2: Classification accuracy vs. number of selected features across nine large-scale classification datasets. Each panel represents a specific dataset.

Table A6: Average relative classification-error reduction (%) of CORAL-MLP over competing methods on nine large-scale classification datasets. Otto and Vehicle denote Otto-Group-Product and Vehicle_sensIT.

Method	ALOI	Covertype	MNIST	Jannis	Meta	Otto	Epsilon	CovPokElec	Vehicle	Avg.
LassoNet	76.93	15.97	50.38	12.35	85.12	62.98	14.12	36.87	8.28	40.33
DFS	35.62	6.02	58.59	7.93	58.63	52.90	6.16	23.96	4.06	28.21
STG	74.10	11.31	63.95	11.47	75.79	68.50	20.24	35.36	8.48	41.02
Lasso	82.85	17.07	65.25	16.84	87.02	78.88	22.19	39.48	14.19	47.09

D.3 Compatibility with bounded activations

Proposition 1 uses boundedness as a sufficient condition, whereas the main benchmark implementations use ReLU for comparability with the baselines. We therefore also evaluated CORAL on the Artificial dataset with ReLU6 and sigmoid, both of which produce bounded nonlinear components in a finite network. Table A7 reports the existing results on this task, with only the activation changed.

Table A7: CORAL with unbounded and bounded activations. Parentheses report standard errors.

Activation	MSE	Recall	Precision
ReLU	0.09 (0.02)	1.00 (0.00)	1.00 (0.00)
ReLU6	0.09 (0.02)	1.00 (0.00)	1.00 (0.00)
Sigmoid	0.11 (0.02)	1.00 (0.00)	1.00 (0.00)

ReLU6 matches ReLU on the synthetic task and slightly improves the reported MNIST accuracy, while sigmoid remains competitive. These results show that the sufficient bounded-activation regime in Proposition 1 is compatible with the empirical behavior of CORAL, while the ReLU results indicate that boundedness is not required for the method to work in practice.