SEQUENTIAL ATTENTION FOR FEATURE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Feature selection is the problem of selecting a subset of features for a machine learning model that maximizes model quality subject to a resource budget constraint. For neural networks, prior methods, including those based on ℓ_1 regularization, attention, and stochastic gates, typically select all of the features in one evaluation round, ignoring the *residual* value of the features during selection (i.e., the marginal contribution of a feature conditioned on the previously selected features). We propose a feature selection algorithm called Sequential Attention that achieves state-of-the-art empirical results for neural networks. This algorithm is based on an efficient implementation of greedy forward selection and uses attention weights at each step as a proxy for marginal feature importance. We provide theoretical insights into our Sequential Attention algorithm for linear regression models by showing that an adaptation to this setting is *equivalent* to the classical Orthogonal Matching Pursuit algorithm (Pati et al., 1993), and thus inherits all of its provable guarantees. Lastly, our theoretical and empirical analyses provide new explanations towards the effectiveness of attention and its connections to overparameterization, which might be of independent interest.

1 INTRODUCTION

Feature selection is a classic problem in machine learning and statistics where one is asked to find a subset of k features from a larger set of d features, such that the prediction quality of the model trained using the subset of features is approximately as good as training on all d features. Finding such a feature subset is desirable for a number of reasons: improving model interpretability, reducing inference latency, decreasing model size, and regularizing models by removing redundant or noisy features to reduce overfitting and improve generalization. We direct the reader to Li et al. (2017b) for a comprehensive survey on the role of feature selection in machine learning.

The widespread success of deep learning has prompted an intense study of feature selection algorithms for *neural networks*, especially in the supervised setting. While many methods have been proposed, we concentrate on a particular line of work centered around using *attention* for feature selection. The attention mechanism in machine learning roughly refers to applying a trainable softmax mask to the input of a layer. This allows the model to "focus on" important parts of the input during training. Attention has recently led to major breakthroughs in computer vision, natural language processing, and several other areas of machine learning (Vaswani et al., 2017). For feature selection, the works of Wang et al. (2014); Gui et al. (2019); Skrlj et al. (2020); Wojtas & Chen (2020); Liao et al. (2021) all present new approaches for feature attribution, ranking, and selection that are inspired by attention.

One problem with naively using attention for feature selection is that it can ignore the *residual* values of features, i.e., the marginal contribution a feature has on the loss conditioned on the presence of the previously-selected features. This can lead to several problems such as selecting redundant features or ignoring features that are uninformative in isolation but valuable in the presence of others.

This work introduces the *Sequential Attention* algorithm for supervised feature selection. Our algorithm addresses the shortcomings above by using attention-based selection adaptively over multiple rounds. Sequential Attention simplifies earlier attention-based feature selection algorithms by directly training one global feature mask instead of aggregating many instance-wise feature masks that are the outputs of different subnetworks. This technique reduces the overhead of our algorithm, removes the need to tune unnecessary hyperparameters, *works directly with any model architecture*, facilitates a streaming implementation, and gives higher-quality estimates for the marginal gains in

prediction quality for the unselected features. Empirically, Sequential Attention achieves state-of-the-art feature selection results for neural networks on standard benchmarks. We have made the code for our algorithm publicly available in a demo notebook[1].

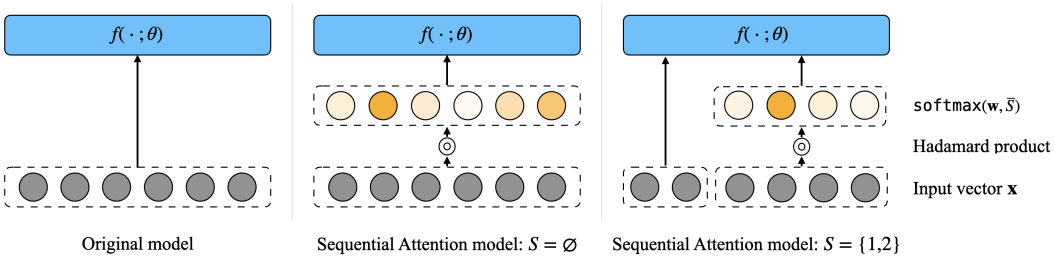

Figure 1: Sequential attention applied to model $f(\cdot;\boldsymbol{\theta})$. At each step, the selected features $i \in S$ are used as direct inputs to the model and the unselected features $i \notin S$ are downscaled by the scalar value $\mathrm{softmax}_i(\mathbf{w}, \overline{S})$, where $\mathbf{w} \in \mathbb{R}^d$ is the vector of learned attention weights and $S := [d] \setminus S$.

Sequential Attention. Our starting point for Sequential Attention is the well-known greedy forward selection algorithm, which iteratively selects the feature with the *largest marginal improvement* in model loss when added to the set of currently selected features (see, e.g., Das & Kempe (2011); Elenberg et al. (2018)). Greedy forward selection is known to select high-quality features, but requires training $O(kd)$ models and is therefore impractical for many modern machine learning problems. To reduce this cost, one natural idea is to only train k models, where the model trained in each step approximates the marginal gains of all $O(d)$ unselected features. These approximate marginal gains are used as the criteria for greedy selection. Said another way, we relax the greedy algorithm to fractionally consider all $O(d)$ feature candidates simultaneously rather than computing their exact marginal gains one-by-one with separate model trainings. We implement this idea by introducing a new set of trainable variables $\mathbf{w} \in \mathbb{R}^d$ that represent *feature importance*, or *attention weights*. Then, in each step, we select the feature with maximum importance score and add it to the selected set. To ensure the score-augmented models in each step (1) have differentiable architectures and (2) are encouraged to hone in on the best unselected feature, we take the *softmax* of the importance weights and multiply each input feature value by its corresponding softmax value as illustrated in Figure 1.

Formally, given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$ represented as a matrix with n rows of examples and d feature columns, suppose we want to select k features. Let $f(\cdot;\boldsymbol{\theta})$ be a differentiable model, e.g., a neural network, that outputs the predictions $f(\mathbf{X};\boldsymbol{\theta})$. Let $\mathbf{y} \in \mathbb{R}^n$ be the labels, $\ell(f(\mathbf{X};\boldsymbol{\theta}), \mathbf{y})$ be the loss between the model's predictions and the labels, and $\circ$ be the Hadamard product. Sequential Attention outputs a set $S \subseteq \{1, 2, \ldots, d\}$ of k feature indices, and is presented below in Algorithm 1.

Algorithm 1 Sequential Attention for feature selection.

function SEQUENTIALATTENTION(dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, labels $\mathbf{y} \in \mathbb{R}^n$, model f, loss ℓ, size k)
  Initialize $S \leftarrow \varnothing$
  for $t = 1$ to k **do**
    Let $(\boldsymbol{\theta}^*, \mathbf{w}^*) \leftarrow \arg\min_{\boldsymbol{\theta}, \mathbf{w}} \ell(f(\mathbf{X} \circ \mathbf{W}; \boldsymbol{\theta}), \mathbf{y})$, where $\mathbf{W} = \mathbf{1}_n \mathrm{softmax}(\mathbf{w}, \overline{S})^\top$ for

$$\mathrm{softmax}_i(\mathbf{w}, \overline{S}) := \begin{cases} 1 & \text{if } i \in S \\ \dfrac{\exp(\mathbf{w}_i)}{\sum_{j \in \overline{S}} \exp(\mathbf{w}_j)} & \text{if } i \in \overline{S} := [d] \setminus S \end{cases} \tag{1}$$

    Set $i^* \leftarrow \arg\max_{i \notin S} \mathbf{w}_i^*$ ▷ unselected feature with largest attention weight
    Update $S \leftarrow S \cup \{i^*\}$
  return S

[1]Colab notebook is available here.

1.1 OUR CONTRIBUTIONS

Theoretical guarantees. We give provable guarantees for Sequential Attention for least squares linear regression by analyzing a variant of the algorithm called *regularized linear Sequential Attention*.

This variant (1) uses *Hadamard product overparameterization* directly between the attention weights and feature values without normalizing the attention weights via $\mathrm{softmax}(\mathbf{w}, \overline{S})$, and (2) adds ℓ_2 regularization to the objective, hence the "linear" and "regularized" terms. Note that ℓ_2 regularization, or *weight decay*, is common practice when using gradient-based optimizers (Tibshirani, 2021). We give theoretical and empirical evidence that replacing the $\mathrm{softmax}$ by different overparameterization schemes leads to similar results (Section 4.2) while offering more tractable analysis. In particular, our main result shows that regularized linear Sequential Attention has the same provable guarantees as the celebrated *Orthogonal Matching Pursuit* (OMP) algorithm of Pati et al. (1993) for sparse linear regression, without making any assumptions on the design matrix or response vector.

Theorem 1.1. *For linear regression, regularized linear Sequential Attention is equivalent to OMP.*

We prove this equivalence using a novel two-step argument. First, we show that regularized linear Sequential Attention is equivalent to a greedy version of LASSO (Tibshirani, 1996), which Luo & Chen (2014) call *Sequential LASSO*. However, prior to our work, Sequential LASSO was only analyzed in a restricted "sparse signal plus noise" setting, offering limited insight into its success in practice. Second, we prove that Sequential LASSO is equivalent to OMP in the fully general setting for linear regression by analyzing the geometry of the associated polyhedra. This ultimately allows us to transfer the guarantees of OMP to Sequential Attention.

Theorem 1.2. *For linear regression, Sequential LASSO (Luo & Chen, 2014) is equivalent to OMP.*

We present the full argument for our results in Section 3. This analysis takes significant steps towards explaining the success of attention in feature selection and the various theoretical phenomena at play.

Towards understanding attention. An important property of OMP is that it provably approximates the marginal gains of features—Das & Kempe (2011) showed that for any subset of features, the gradient of the least squares loss at its minimizer approximates the marginal gains up to a factor that depends on the *sparse condition numbers* of the design matrix. This suggests that Sequential Attention could also approximate some notion of the marginal gains for more sophisticated models when selecting the next-best feature. We observe this phenomenon empirically in our marginal gain experiments in Appendix B.6. These results also help refine the widely-assumed conjecture that attention weights correlate with feature importances by specifying an exact measure of "importance" at play. Since a countless number of feature importance definitions are used in practice, it is important to determine which best explains how the attention mechanism works.

Connections to overparameterization. In our analysis of regularized linear Sequential Attention for linear regression, we do not use the presence of the softmax in the attention mechanism—rather, the crucial ingredient in our analysis is the Hadamard product parameterization of the learned weights. We conjecture that the empirical success of attention-based feature selection is primarily due to the explicit overparameterization[2]. Indeed, our experiments in Section 4.2 verify this claim by showing that if we substitute the softmax in Sequential Attention with a number of different (normalized) overparamterized expressions, we achieve nearly identical performance. This line of reasoning is also supported in the recent work of Ye et al. (2021), who claim that attention largely owes its success to the "smoother and stable [loss] landscapes" that Hadamard product overparameterization induces.

1.2 RELATED WORK

We discuss recent advances in supervised feature selection for deep neural networks (DNNs), as these works are the most relevant for our empirical results. In particular, we omit a discussion of a large body of works on unsupervised feature selection Zou et al. (2015); Altschuler et al. (2016); Balın et al. (2019).

[2]Note that "overparameterization" here refers to the addition of d trainable variables in the Hadamard product overparameterization, rather than another common use of the word which refers to the use of a massive number of parameters in neural networks in, e.g., Bubeck & Sellke (2021).

The *group LASSO* has been used in deep learning to achieve structured sparsity by pruning neurons (Alvarez & Salzmann, 2016) and even filters or channels in convolutional neural networks (Lebedev & Lempitsky, 2016; Wen et al., 2016; Li et al., 2017a). For feature selection in particular, the LASSO and group LASSO were applied to neural networks in Zhao et al. (2015); Li et al. (2016); Scardapane et al. (2017); Lemhadri et al. (2021).

The LASSO is the most widely-used method for relaxing the ℓ_0 sparsity constraint in feature selection, but several recent works have proposed new relaxations based on *stochastic gates* (Srinivas et al., 2017; Louizos et al., 2018; Balın et al., 2019; Trelin & Procházka, 2020; Yamada et al., 2020). This approach introduces (learnable) Bernoulli random variables for each feature during model training and minimizes the expected loss over realizations of the 0-1 variables (accepting or rejecting features).

There are several other recently suggested approaches to feature selection for DNNs. Roy et al. (2015) use the magnitudes of weights in the first hidden layer to select features. Lu et al. (2018) proposed the DeepPINK architecture, extending the idea of *knockoffs* (Benjamini et al., 2001) to neural networks. Here, each feature is paired with a "knockoff" version that competes with the original feature; if the knockoff wins, the feature is removed. Borisov et al. (2019) introduced the *CancelOut* DNN layer, which suppresses irrelevant features via independent per-feature activation functions, i.e., sigmoids, that act as (soft) bitmasks.

In contrast to the aforementioned works, combinatorial optimization is rich with sequential algorithms that are applied in machine learning (Zadeh et al., 2017; Fahrbach et al., 2019b;a; Chen et al., 2021; Halabi et al., 2022; Bilmes, 2022). In fact, most influential feature selection algorithms from this literature are sequential, e.g., greedy forward and backward selection (Ye & Sun, 2018; Das et al., 2022), Orthogonal Matching Pursuit (Pati et al., 1993), and information-theoretic methods (Fleuret, 2004; Ding & Peng, 2005; Bennasar et al., 2015). However, these methods are often not tailored to neural networks, and suffer from quality, efficiency, or both.

Finally, we study *global* feature selection, i.e., selecting the same subset of important features across all training examples, but there are also many important works that consider *local* (or instance-wise) feature selection. This problem is often categorized as *model interpretability*, and is also known as computing *feature attribution* or *saliency maps*. These methods also naturally yield global feature selection, by aggregating instance-wise scores (Cancela et al., 2020). Instance-wise feature selection has been explored using a variety of techniques, including gradients (Smilkov et al., 2017; Sundararajan et al., 2017; Srinivas & Fleuret, 2019), attention (Arik & Pfister, 2021; Ye et al., 2021), mutual information (Chen et al., 2018), and Shapley values from cooperative game theory (Lundberg & Lee, 2017).

2 PRELIMINARIES

Before discussing our theoretical guarantees for Sequential Attention in Section 3, we present several known results about feature selection for linear regression, also called *sparse linear regression*. Recall that in the least squares linear regression problem, we have

$$\ell(f(\mathbf{X};\boldsymbol{\theta}),\mathbf{y}) = \|f(\mathbf{X};\boldsymbol{\theta}) - \mathbf{y}\|_2^2 = \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2. \tag{2}$$

We work in the most challenging setting for obtaining relative error guarantees for this objective by making *no distributional assumptions* on $\mathbf{X} \in \mathbb{R}^{n\times d}$, i.e., we seek $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^d$ such that

$$\|\mathbf{X}\tilde{\boldsymbol{\theta}} - \mathbf{y}\|_2^2 \leq \kappa \min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2, \tag{3}$$

for some $\kappa = \kappa(\mathbf{X}) > 0$, where $\mathbf{X}$ is not assumed to follow any particular input distribution. This is far more applicable in practice than, e.g., assuming the entries of $\mathbf{X}$ are i.i.d. Gaussian. In large-scale applications, the number of examples, n, often greatly exceeds the number of features, d, resulting in an optimal loss that is nonzero. Therefore, we focus on the *overdetermined* regime and refer to (Price et al., 2022) for an excellent discussion on the long history of this problem.

Notation. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote the design matrix with ℓ_2 unit columns and let $\mathbf{y} \in \mathbb{R}^n$ denote the label vector, also assumed to be an ℓ_2 unit vector.[3] For $S \subseteq [d]$, let $\mathbf{X}_S$ denote the $n \times |S|$ matrix

[3]These assumptions are without loss of generality by scaling.

consisting of the columns of $\mathbf{X}$ indexed by S. For singleton sets $S = \{j\}$, we simply write $\mathbf{X}_j$ for $\mathbf{X}_{\{j\}}$. Let $\mathbf{P}_S \coloneqq \mathbf{X}_S \mathbf{X}_S^+$ denote the projection matrix onto the column span $\mathrm{colspan}(\mathbf{X}_S)$ of $\mathbf{X}_S$, where $\mathbf{X}_S^+$ denotes the pseudoinverse of $\mathbf{X}_S$. Let $\mathbf{P}_S^\perp = \mathbf{I}_n - \mathbf{P}_S$ denote the projection matrix onto the orthogonal complement of $\mathrm{colspan}(\mathbf{X}_S)$.

Feature selection algorithms for linear regression. Perhaps the most natural algorithm for sparse linear regression is greedy forward selection, which was shown to have guarantees of the form of (3) in the breakthrough works of Das & Kempe (2011); Elenberg et al. (2018), where $\kappa = \kappa(\mathbf{X})$ depends on *sparse condition numbers* of $\mathbf{X}$, i.e., the spectrum of $\mathbf{X}$ restricted to a subset of its columns. Greedy forward selection can be expensive in practice, but these works also prove analogous guarantees for the more efficient Orthogonal Matching Pursuit algorithm, which we present formally in Algorithm 2.

Algorithm 2 Orthogonal Matching Pursuit (Pati et al., 1993).

function OMP(design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, response $\mathbf{y} \in \mathbb{R}^n$, size constraint k)
  Initialize $S \leftarrow \varnothing$
  for $t = 1$ to k **do**
    Set $\boldsymbol{\beta}_S^* \leftarrow \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^S} \|\mathbf{X}_S \boldsymbol{\beta} - \mathbf{y}\|_2^2$
    Let $i^* \notin S$ maximize ▷ maximum correlation with residual

$$\langle \mathbf{X}_i, \mathbf{y} - \mathbf{X}_S \boldsymbol{\beta}_S^* \rangle^2 = \langle \mathbf{X}_i, \mathbf{y} - \mathbf{P}_S \mathbf{y} \rangle^2 = \left\langle \mathbf{X}_i, \mathbf{P}_S^\perp \mathbf{y} \right\rangle^2$$

    Update $S \leftarrow S \cup \{i^*\}$
  return S

The LASSO algorithm (Tibshirani, 1996) is another popular feature selection method, which simply adds ℓ_1-regularization to the objective in Equation (2). Theoretical guarantees for LASSO are known in the *underdetermined* regime (Donoho & Elad, 2003; Candes & Tao, 2006), but it is an open problem whether LASSO has the guarantees of Equation (3). Sequential LASSO is a related algorithm that uses the LASSO to select features one by one. Luo & Chen (2014) analyzed this algorithm in a specific parameter regime, but until our work, *no relative error guarantees were known in full generality (e.g., the overdetermined regime)*. We present the Sequential LASSO in Algorithm 3.

Algorithm 3 Sequential LASSO (Luo & Chen, 2014).

function SEQUENTIALLASSO(design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, response $\mathbf{y} \in \mathbb{R}^n$, size constraint k)
  Initialize $S \leftarrow \varnothing$
  for $t = 1$ to k **do**
    Let $\boldsymbol{\beta}^*(\lambda, S)$ denote the optimal solution to

$$\arg\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\beta}_{\overline{S}}\|_1 \tag{4}$$

    Set $\lambda^*(S) \leftarrow \sup\{\lambda > 0 : \boldsymbol{\beta}^*(\lambda, S)_{\overline{S}} \neq \mathbf{0}\}$ ▷ largest λ with nonzero LASSO on $\overline{S}$
    Let $A(S) = \lim_{\varepsilon \to 0} \{i \in \overline{S} : \boldsymbol{\beta}^*(\lambda^* - \varepsilon, S)_i \neq 0\}$
    Select any $i^* \in A(S)$ ▷ non-empty by Lemma 3.5
    Update $S \leftarrow S \cup \{i^*\}$
  return S

Note that Sequential LASSO as stated requires a search for the optimal λ^* in each step. In practice, however, λ can simply be set to a large enough value to obtain similar results, since beyond a critical value of λ, the feature ranking according to LASSO coefficients does not change (Efron et al., 2004).

3 EQUIVALENCE FOR LEAST SQUARES: OMP AND SEQUENTIAL ATTENTION

In this section, we show that the following algorithms are equivalent for least squares linear regression: regularized linear Sequential Attention, Sequential LASSO, and Orthogonal Matching Pursuit.

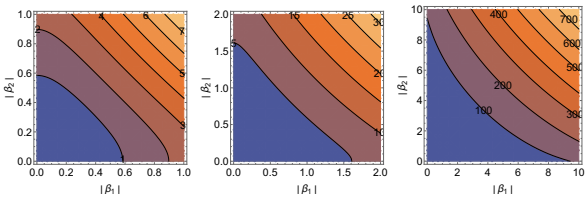

Figure 2: Contour plot of $Q^*(\boldsymbol{\beta} \circ \boldsymbol{\beta})$ for $\boldsymbol{\beta} \in \mathbb{R}^2$ at different zoom-levels of $|\boldsymbol{\beta}_i|$.

3.1 Regularized linear Sequential Attention and Sequential LASSO

We start by formalizing our modified version of Sequential Attention, for which we obtain provable guarantees.

Definition 3.1 (Regularized linear Sequential Attention)**.** Let $S \subseteq \{1, 2, \ldots, d\}$ be the set of currently selected features. We define the *regularized linear Sequential Attention* objective by removing the $\mathrm{softmax}(\mathbf{w}, \overline{S})$ normalization in Algorithm 1 and introducing ℓ_2 regularization on the importance weights $\mathbf{w} \in \mathbb{R}^{\overline{S}}$ and the model parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ restricted to $\overline{S}$. That is, we consider the objective

$$\min_{\mathbf{w} \in \mathbb{R}^d, \boldsymbol{\theta} \in \mathbb{R}^d} \|\mathbf{X}(\mathbf{s}(\mathbf{w}) \circ \boldsymbol{\theta}) - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\Big(\|\mathbf{w}\|_2^2 + \|\boldsymbol{\theta}_{\overline{S}}\|_2^2\Big), \tag{5}$$

where $\mathbf{s}(\mathbf{w}) \circ \boldsymbol{\theta}$ denotes the Hadamard product, $\boldsymbol{\theta}_{\overline{S}} \in \mathbb{R}^{\overline{S}}$ is $\boldsymbol{\theta}$ restricted to indices in $\overline{S}$, and

$$\mathbf{s}_i(\mathbf{w}, \overline{S}) := \begin{cases} 1 & \text{if } i \in S, \\ \mathbf{w}_i & \text{if } i \notin S. \end{cases}$$

By a simple argument due to Hoff (2017), the objective function in (5) is equivalent to

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}_{\overline{S}}\|_1. \tag{6}$$

It follows that attention (or more generally *overparameterization* by trainable weights $\mathbf{w}$) can be seen as a way to implement ℓ_1 regularization for least squares linear regression, i.e., the LASSO (Tibshirani, 1996). This connection between overparameterization and ℓ_1 regularization has also been observed in several other recent works (Vaskevicius et al., 2019; Zhao et al., 2022; Tibshirani, 2021).

By this transformation and reasoning, regularized linear Sequential Attention can be seen as iteratively using the LASSO with ℓ_1 regularization applied only to the unselected features—which is precisely the Sequential LASSO algorithm in (Luo & Chen, 2014). If we instead use $\mathrm{softmax}(\mathbf{w}, \overline{S})$ as in (1), then this only changes the choice of regularization, as shown in Lemma 3.2 (proof in Appendix A.3).

Lemma 3.2. *Let $D : \mathbb{R}^d \to \mathbb{R}^{\overline{S}}$ be the function defined by $D(\mathbf{w})_i = 1/\mathrm{softmax}_i^2(\mathbf{w}, \overline{S})$, for $i \in \overline{S}$. Denote its range and preimage by $\mathrm{ran}(D) \subseteq \mathbb{R}^{\overline{S}}$ and $D^{-1}(\cdot) \subseteq \mathbb{R}^d$, respectively. Moreover, define the functions $Q : \mathrm{ran}(D) \to \mathbb{R}$ and $Q^* : \mathbb{R}^{\overline{S}} \to \mathbb{R}$ by*

$$Q(\mathbf{q}) = \inf_{\mathbf{w} \in D^{-1}(\mathbf{q})} \|\mathbf{w}\|_2^2 \qquad \textit{and} \qquad Q^*(\mathbf{x}) = \inf_{\mathbf{q} \in \mathrm{ran}(D)} \left(\sum_{i \in \overline{S}} \mathbf{x}_i \mathbf{q}_i + Q(\mathbf{q}) \right).$$

Then, the following two optimization problems with respect to $\boldsymbol{\beta} \in \mathbb{R}^d$ are equivalent:

$$\inf_{\substack{\boldsymbol{\beta} \in \mathbb{R}^d \\ \text{s.t. } \boldsymbol{\beta} = \mathrm{softmax}(\mathbf{w}, \overline{S}) \circ \boldsymbol{\theta} \\ \mathbf{w} \in \mathbb{R}^d, \boldsymbol{\theta} \in \mathbb{R}^d}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\Big(\|\mathbf{w}\|_2^2 + \|\boldsymbol{\theta}_{\overline{S}}\|_2^2\Big) \equiv \inf_{\boldsymbol{\beta} \in \mathbb{R}^d} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} Q^*(\boldsymbol{\beta} \circ \boldsymbol{\beta}). \tag{7}$$

We present contour plots of $Q^*(\boldsymbol{\beta} \circ \boldsymbol{\beta})$ for $\boldsymbol{\beta} \in \mathbb{R}^2$ in Figure 2. These plots suggest that $Q^*(\boldsymbol{\beta} \circ \boldsymbol{\beta})$ is a concave regularizer when $|\beta_1| + |\beta_2| > 2$, which would thus approximate the ℓ_0 regularizer and induce a sparse solution of $\boldsymbol{\beta}$ (Zhang & Zhang, 2012), as ℓ_1 regularization does (Tibshirani, 1996).

3.2 Sequential LASSO and OMP

This connection between Sequential Attention and Sequential LASSO gives us a new perspective about how Sequential Attention works. The only known guarantee for Sequential LASSO, to the best

of our knowledge, is a statistical recovery result when the input is a sparse linear combination with Gaussian noise in the ultra high-dimensional setting (Luo & Chen, 2014). This does not, however, fully explain why Sequential Attention is such an effective feature selection algorithm.

To bridge our main results, we prove a novel equivalence between Sequential LASSO and OMP. For the remainder of this section, let $[d] := \{1, 2, \ldots, d\}$.

Theorem 3.3. *Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a design matrix with ℓ_2 unit vector columns, and let $\mathbf{y} \in \mathbb{R}^d$ denote the response, also an ℓ_2 unit vector. The Sequential LASSO algorithm maintains a set of features $S \subseteq [d]$ such that, at each feature selection step, it selects a feature $i \in \overline{S}$ such that*

$$\left|\langle \mathbf{X}_i, \mathbf{P}_S^{\perp}\mathbf{y}\rangle\right| = \left\|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\right\|_\infty,$$

where $\mathbf{X}_S$ is the $n \times |S|$ matrix given formed by the columns of $\mathbf{X}$ indexed by S, and $\mathbf{P}_S^{\perp}$ is the projection matrix onto the orthogonal complement of the span of $\mathbf{X}_S$.

Note that this is extremely close to saying that the Sequential LASSO and OMP select the exact same set of features. The only difference appears when there are multiple features witnessing the ℓ_∞ norm $\left\|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\right\|_\infty$. In this case, it is possible that Sequential LASSO chooses the next feature from a set of features that is strictly smaller than the set of features from which OMP chooses, so the "tie-breaking" can differ between the two algorithms. In practice, this rarely happens; for instance, if only one feature is selected at each step, which is the case with probability 1 if random continuous noise is added to the data, then Sequential LASSO and OMP will select the exact same set of features.

Remark 3.4. It was shown in (Luo & Chen, 2014) that Sequential LASSO is equivalent to OMP in the statistical recovery regime, i.e., when $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}$ for some true sparse weight vector $\boldsymbol{\beta}^*$ and i.i.d. Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma \mathbf{I}_n)$, under an ultra high-dimensional regime where the dimension d is exponential in the number of examples n. We prove this equivalence in the *fully general setting*.

The argument below shows that Sequential LASSO and OMP are equivalent, thus establishing that regularized linear Sequential Attention and Sequential LASSO offer the same approximation guarantees as OMP.

Geometry of Sequential LASSO. We first study the geometry of optimal solutions to Equation (4). Let $S \subseteq [d]$ be the set of currently selected features. Following work on the LASSO in (Tibshirani & Taylor, 2011), we rewrite (4) as the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^d} \quad & \frac{1}{2}\|\mathbf{z} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\beta}_{\overline{S}}\|_1 \\ \text{subject to} \quad & \mathbf{z} = \mathbf{X}\boldsymbol{\beta}. \end{aligned} \tag{8}$$

It can then be shown that the dual problem is equivalent to finding the projection, i.e., closest point in Euclidean distance, $\mathbf{u} \in \mathbb{R}^n$ of $\mathbf{P}_S^{\perp}\mathbf{y}$ onto the polyhedral section $C_\lambda \cap \operatorname{colspan}(\mathbf{X}_S)^\perp$, where

$$C_\lambda := \left\{\mathbf{u}' \in \mathbb{R}^n : \left\|\mathbf{X}^\top \mathbf{u}'\right\|_\infty \leq \lambda\right\}$$

and $\operatorname{colspan}(\mathbf{X}_S)^\perp$ denotes the orthogonal complement of $\operatorname{colspan}(\mathbf{X}_S)$. See Appendix A.1 for the full details. The primal and dual variables are related through $\mathbf{z}$ by

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{z} = \mathbf{y} - \mathbf{u}. \tag{9}$$

Selection of features in Sequential LASSO. Next, we analyze how Sequential LASSO selects its features. Let $\boldsymbol{\beta}_S^* = \mathbf{X}_S^{+}\mathbf{y}$ be the optimal solution for features restricted in S. Then subtracting $\mathbf{X}_S\boldsymbol{\beta}_S^*$ from both sides of (9) gives

$$\mathbf{X}\boldsymbol{\beta} - \mathbf{X}_S\boldsymbol{\beta}_S^* = \mathbf{y} - \mathbf{X}_S\boldsymbol{\beta}_S^* - \mathbf{u} = \mathbf{P}_S^{\perp}\mathbf{y} - \mathbf{u}. \tag{10}$$

Note that if λ is at least $\lambda \geq \|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\|_\infty$, then the projection of $\mathbf{P}_S^{\perp}\mathbf{y}$ onto C_λ is just $\mathbf{u} = \mathbf{P}_S^{\perp}\mathbf{y}$, so by (10),

$$\mathbf{X}\boldsymbol{\beta} - \mathbf{X}_S\boldsymbol{\beta}_S^* = \mathbf{P}_S^{\perp}\mathbf{y} - \mathbf{P}_S^{\perp}\mathbf{y} = \mathbf{0},$$

meaning that $\boldsymbol{\beta}$ is zero outside of S. We now show that for λ slightly smaller than $\|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\|_\infty$, the residual $\mathbf{P}_S^{\perp}\mathbf{y} - \mathbf{u}$ is in the span of features $\mathbf{X}_i$ that maximize the correlation with $\mathbf{P}_S^{\perp}\mathbf{y}$.

Lemma 3.5 (Projection residuals of the Sequential LASSO)**.** *Let* $\mathbf{p}$ *denote the projection of* $\mathbf{P}_S^{\perp}\mathbf{y}$ *onto* $C_\lambda \cap \operatorname{colspan}(\mathbf{X}_S)^{\perp}$*. There exists* $\lambda_0 < \left\|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\right\|_\infty$ *such that for all* $\lambda \in (\lambda_0, \|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\|_\infty)$ *the residual* $\mathbf{P}_S^{\perp}\mathbf{y} - \mathbf{p}$ *lies on* $\operatorname{colspan}(\mathbf{X}_T)$*, for*

$$T := \left\{i \in [d] : \left|\left\langle \mathbf{X}_i, \mathbf{P}_S^{\perp}\mathbf{y}\right\rangle\right| = \left\|\mathbf{X}^\top \mathbf{P}_S^{\perp}\mathbf{y}\right\|_\infty\right\}.$$

We defer the proof of Lemma 3.5 to Appendix A.2.

By Lemma 3.5 and (10), the optimal $\boldsymbol{\beta}$ when selecting the next feature has the following properties:

1. if $i \in S$, then $\boldsymbol{\beta}_i$ is equal to the i-th value in the previous solution $\boldsymbol{\beta}_S^*$; and
2. if $i \notin S$, then $\boldsymbol{\beta}_i$ can be nonzero only if $i \in T$.

It follows that Sequential LASSO selects a feature that maximizes the correlation $|\langle \mathbf{X}_j, \mathbf{P}_S^{\perp}\mathbf{y}\rangle|$, just as OMP does. Thus, we have shown an equivalence between Sequential LASSO and OMP *without any additional assumptions*.

4 Experiments

4.1 Feature selection for neural networks

Small-scale experiments. We investigate the performance of Sequential Attention, as presented in Algorithm 1, through extensive experiments on standard feature selection benchmarks for neural networks. In these experiments, we consider six datasets used in experiments in (Lemhadri et al., 2021; Balın et al., 2019), and select $k = 50$ features using a one-layer neural network with hidden width 67 and ReLU activation (just as in these previous works). For additional points of comparison, we implement the attention-based feature selection algorithms of (Balın et al., 2019; Liao et al., 2021) and the Group LASSO, which has been considered in many works that aim to sparisfiy neural networks, as discussed in Section 1.2. We also implement natural adaptations of the Sequential LASSO and OMP for neural networks and evaluate their performance.

In Figure 3 we see that Sequential Attention is competitive with or outperforms all of these feature selection algorithms on this benchmark suite. For each algorithm, we report the mean of the prediction accuracies averaged over 5 feature selection trials. We provide additional details about the experimental setup in Appendix B.2, including specifications about each dataset in Table 1 and the raw mean prediction accuracies with standard deviations in Table 2. We also visualize the selected features on MNIST in Figure 5.

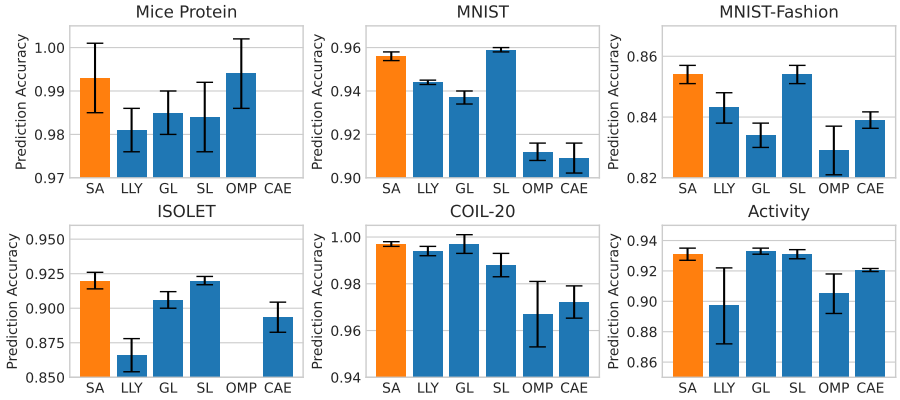

Figure 3: Feature selection results on small-scale datasets. Here, SA = Sequential Attention, LLY = (Liao et al., 2021), GL = Group LASSO, SL = Sequential LASSO, OMP = OMP, and CAE = Concrete Autoencoder (Balın et al., 2019).

We note that our algorithm is considerably more efficient compared to prior feature selection algorithms, especially those designed for neural networks. This is because many of these prior algorithms

introduce entire subnetworks to train Balın et al. (2019); Gui et al. (2019); Wojtas & Chen (2020); Liao et al. (2021), whereas Sequential Attention only adds d additional trainable variables. Furthermore, in these experiments, we implement an optimized version of Algorithm 1 which only trains one model rather than k models, by partitioning the training epochs into k parts and selecting a feature in each of these k parts. These two aspects combined make for an extremely efficient algorithm. We provide an evaluation of the efficiency of Sequential Attention in Appendix B.2.3.

Large-scale experiments. To demonstrate the scalability of our algorithm, we perform large-scale feature selection experiments on the Criteo click dataset, which consists of 39 features and over three billion examples for predicting click-through rates (Diemert Eustache, Meynet Julien et al., 2017). Our results in Figure 4 show that Sequential Attention outperforms these other methods when at least 15 features are selected. In particular, these plots highlight the fact that Sequential Attention excels at finding valuable features once a few features are already in the model, and that it has substantially less variance than LASSO-based feature selection algorithms. See Appendix B.3 for further discussion.

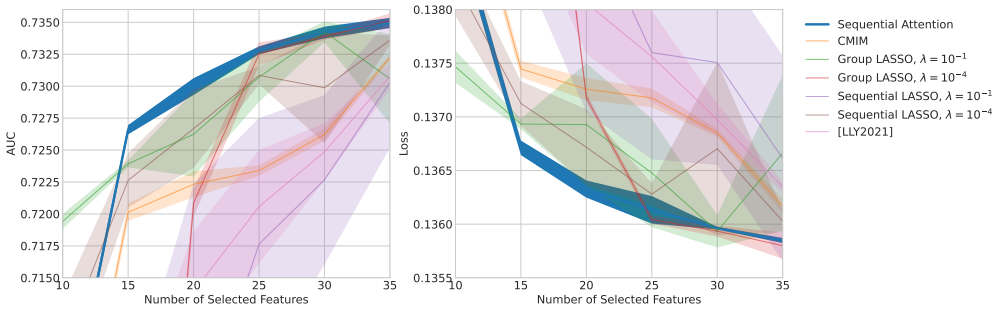

Figure 4: AUC and log loss when selecting $k \in \{10, 15, 20, 25, 30, 35\}$ features for Criteo dataset.

4.2 The role of Hadamard product overparameterization in attention

In Section 1.1, we argue that Sequential Attention has provable guarantees for least squares linear regression by showing that a version that removes the softmax and adds ℓ_2 regularization results in an algorithm that is equivalent to OMP. Thus, there is a gap between the implementation of Sequential Attention in Algorithm 1 and our theoretical analysis. We empirically bridge this gap by showing that regularized linear Sequential Attention yields results that are almost indistinguishable to the original version. In Figure 11 (Appendix B.5), we compare the following Hadamard product overparameterization schemes:

- softmax: as described in Section 1
- ℓ_1: $\mathbf{s}_i(\mathbf{w}) = |\mathbf{w}_i|$ for $i \in \overline{S}$, which captures the provable variant discussed in Section 1.1
- ℓ_2: $\mathbf{s}_i(\mathbf{w}) = |\mathbf{w}_i|^2$ for $i \in \overline{S}$
- ℓ_1 normalized: $\mathbf{s}_i(\mathbf{w}) = |\mathbf{w}_i| / \sum_{j \in \overline{S}} |\mathbf{w}_j|$ for $i \in \overline{S}$
- ℓ_2 normalized: $\mathbf{s}_i(\mathbf{w}) = |\mathbf{w}_i|^2 / \sum_{j \in \overline{S}} |\mathbf{w}_j|^2$ for $i \in \overline{S}$

Further, for each of the benchmark datasets, all of these variants outperform LassoNet and the other baselines considered in (Lemhadri et al., 2021). See Appendix B.5 for more details.

5 Conclusion

This work introduces Sequential Attention, an adaptive attention-based feature selection algorithm designed in part for DNNs. Empirically, Sequential Attention improves significantly upon previous methods on widely-used benchmarks. Theoretically, we show that a relaxed variant of Sequential Attention is equivalent to the Sequential LASSO algorithm (Luo & Chen, 2014). In turn, we prove a novel connection between Sequential LASSO and Orthogonal Matching Pursuit, thus transferring its provable guarantees to Sequential Attention and shedding light on our strong empirical results. Our analysis also provides new insight into the the role of attention for feature selection via adaptivity, overparameterization, and connections to marginal gains.

BIBLIOGRAPHY

Jason M. Altschuler, Aditya Bhaskara, Gang Fu, Vahab S. Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy column subset selection: New bounds and distributed algorithms. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48, pp. 2539–2548. JMLR, 2016.

Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. *Advances in neural information processing systems*, 29, 2016.

Sercan Ö Arik and Tomas Pfister. TabNet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6679–6687, 2021.

Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *International conference on machine learning*, pp. 444–453. PMLR, 2019.

Yoav Benjamini, Dan Drai, Greg Elmer, Neri Kafkafi, and Ilan Golani. Controlling the false discovery rate in behavior genetics research. *Behavioural Brain Research*, 125(1-2):279–284, 2001.

Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

Jeff Bilmes. Submodularity in machine learning and artificial intelligence. *arXiv preprint arXiv:2202.00132*, 2022.

Vadim Borisov, Johannes Haug, and Gjergji Kasneci. CancelOut: A layer for feature selection in deep neural networks. In *International conference on artificial neural networks*, pp. 72–83. Springer, 2019.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. In *Advances in Neural Information Processing Systems*, pp. 28811–28822, 2021.

Brais Cancela, Verónica Bolón-Canedo, Amparo Alonso-Betanzos, and João Gama. A scalable saliency-based feature selection method with instance-level information. *Knowl. Based Syst.*, 192: 105326, 2020.

Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pp. 883–892. PMLR, 2018.

Lin Chen, Hossein Esfandiari, Gang Fu, Vahab S. Mirrokni, and Qian Yu. Feature Cross Search via Submodular Optimization. In *29th Annual European Symposium on Algorithms (ESA 2021)*, pp. 31:1–31:16, 2021.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1057–1064, 2011.

Sandipan Das, Alireza M Javid, Prakash Borpatra Gohain, Yonina C Eldar, and Saikat Chatterjee. Neural greedy pursuit for feature selection. *arXiv preprint arXiv:2207.09390*, 2022.

Diemert Eustache, Meynet Julien, Pierre Galland, and Damien Lefortier. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, Halifax, NS, Canada, August, 14, 2017*. ACM, 2017.

Chris H. Q. Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.*, 3(2):185–206, 2005.

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5): 2197–2202, 2003.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568, 2018.

Matthew Fahrbach, Vahab Mirrokni, and Morteza Zadimoghaddam. Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In *International Conference on Machine Learning*, pp. 1833–1842. PMLR, 2019a.

Matthew Fahrbach, Vahab Mirrokni, and Morteza Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 255–273. SIAM, 2019b.

François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine learning research*, 5(9), 2004.

Ning Gui, Danni Ge, and Ziyin Hu. AFS: An attention-based mechanism for supervised feature selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3705–3713, 2019.

Marwa El Halabi, Suraj Srinivas, and Simon Lacoste-Julien. Data-efficient structured pruning via submodular optimization. *arXiv preprint arXiv:2203.04940*, 2022.

Peter D Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. *Computational Statistics & Data Analysis*, 115:186–198, 2017.

Vadim Lebedev and Victor Lempitsky. Fast convnets using group-wise brain damage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2554–2564, 2016.

Ismael Lemhadri, Feng Ruan, and Rob Tibshirani. Lassonet: Neural networks with feature sparsity. In *International Conference on Artificial Intelligence and Statistics*, pp. 10–18. PMLR, 2021.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *5th International Conference on Learning Representations (ICLR)*, 2017a.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017b.

Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: Theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

Yiwen Liao, Raphaël Latty, and Bin Yang. Feature selection using batch-wise attenuation and feature mask normalization. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2021.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through L_0 regularization. In *6th International Conference on Learning Representations (ICLR)*, 2018.

Yang Lu, Yingying Fan, Jinchi Lv, and William Stafford Noble. DeepPINK: Reproducible feature selection in deep neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

Shan Luo and Zehua Chen. Sequential lasso cum EBIC for feature selection with ultra-high dimensional feature space. *Journal of the American Statistical Association*, 109(507):1229–1240, 2014.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44. IEEE, 1993.

Eric Price, Sandeep Silwal, and Samson Zhou. Hardness and algorithms for robust and sparse optimization. In *International Conference on Machine Learning*, pp. 17926–17944. PMLR, 2022.

Debaditya Roy, K Sri Rama Murty, and C Krishna Mohan. Feature selection using deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2015.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Blaz Skrlj, Saso Dzeroski, Nada Lavrac, and Matej Petkovic. Feature importance estimation with self-attention networks. In *24th European Conference on Artificial Intelligence (ECAI)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 1491–1498. IOS Press, 2020.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: Removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. *Advances in Neural Information Processing Systems*, 32, 2019.

Suraj Srinivas, Akshayvarun Subramanya, and R Venkatesh Babu. Training sparse neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pp. 138–145, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

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

Ryan J Tibshirani. Equivalences between sparse models and neural networks. *Working Notes. URL https://www. stat. cmu. edu/˜ ryantibs/papers/sparsitynn. pdf*, 2021.

Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371, 2011.

Andrii Trelin and Aleš Procházka. Binary stochastic filtering: Feature selection and beyond. *arXiv preprint arXiv:2007.03920*, 2020.

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. *Advances in Neural Information Processing Systems*, 32, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Qian Wang, Jiaxing Zhang, Sen Song, and Zheng Zhang. Attentional neural network: Feature selection using cognitive feedback. *Advances in Neural Information Processing Systems*, 27, 2014.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in Neural Information Processing Systems*, 29, 2016.

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

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In *International Conference on Machine Learning*, pp. 10648–10659. PMLR, 2020.

Mao Ye and Yan Sun. Variable selection via penalized neural network: A drop-out-one loss approach. In *International Conference on Machine Learning*, pp. 5620–5629. PMLR, 2018.

Xiang Ye, Zihang He, Heng Wang, and Yong Li. Towards understanding the effectiveness of attention mechanism. *arXiv preprint arXiv:2106.15067*, 2021.

Sepehr Abbasi Zadeh, Mehrdad Ghadiri, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Scalable feature selection via distributed diversity maximization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 2876–2883. AAAI Press, 2017.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statist. Sci.*, 27(4):576–593, 2012.

Lei Zhao, Qinghua Hu, and Wenwu Wang. Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso. *IEEE Transactions on Multimedia*, 17(11):1936–1948, 2015.

Peng Zhao, Yun Yang, and Qiao-Chu He. High-dimensional linear regression via implicit regularization. *Biometrika*, 2022.

Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote. Sens. Lett.*, 12(11):2321–2325, 2015.

A MISSING PROOFS FROM SECTION 3

A.1 LAGRANGIAN DUAL OF SEQUENTIAL LASSO

We will first show that the Lagrangian dual of (8) is equivalent to the following problem:

$$\begin{aligned} \min_{\mathbf{u}\in\mathbb{R}^n} \quad & \frac{1}{2}\|\mathbf{y}-\mathbf{u}\|_2^2 \\ \text{subject to} \quad & \left\|\mathbf{X}^\top\mathbf{u}\right\|_\infty \le \lambda \\ & \mathbf{X}_j^\top\mathbf{u} = 0, \qquad j\in S \end{aligned} \tag{11}$$

We will then use the Pythagorean theorem to replace $\mathbf{y}$ by $\mathbf{P}_S^\perp\mathbf{y}$.

We first consider the Lagrangian dual problem:

$$\max_{\mathbf{u}\in\mathbb{R}^n}\min_{\mathbf{z}\in\mathbb{R}^n,\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{z}-\mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}|_{\overline{S}}\|_1 + \mathbf{u}^\top(\mathbf{z}-\mathbf{X}\boldsymbol{\beta}). \tag{12}$$

Note that the primal problem is strictly feasible and convex, so strong duality holds (see, e.g., Section 5.2.3 of Boyd & Vandenberghe (2004)). Considering just the terms involving the variable $\mathbf{z}$ in (12), we have that

$$\begin{aligned} \frac{1}{2}\|\mathbf{z}-\mathbf{y}\|_2^2 + \mathbf{u}^\top\mathbf{z} &= \frac{1}{2}\|\mathbf{z}\|_2^2 - (\mathbf{y}-\mathbf{u})^\top\mathbf{z} + \frac{1}{2}\|\mathbf{y}\|_2^2 \\ &= \frac{1}{2}\|\mathbf{z}-(\mathbf{y}-\mathbf{u})\|_2^2 + \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{1}{2}\|\mathbf{y}-\mathbf{u}\|_2^2, \end{aligned}$$

which is minimized at $\mathbf{z}=\mathbf{y}-\mathbf{u}$ as $\mathbf{z}$ varies over $\mathbb{R}^n$. On the other hand, consider just the terms involving the variable $\boldsymbol{\beta}$ in (12), that is,

$$\lambda\|\boldsymbol{\beta}_{\overline{S}}\|_1 - \mathbf{u}^\top\mathbf{X}\boldsymbol{\beta}. \tag{13}$$

Note that if $\mathbf{X}^\top\mathbf{u}$ is nonzero on any coordinate in S, then (13) can be made arbitrarily negative by setting $\boldsymbol{\beta}_{\overline{S}}$ to be zero and $\boldsymbol{\beta}_S$ appropriately. Similarly, if $\|\mathbf{X}^\top\mathbf{u}\|_\infty > \lambda$, then (13) can also be made to be arbitrarily negative. On the other hand, if $(\mathbf{X}^\top\mathbf{u})_S = \mathbf{0}$ and $\left\|\mathbf{X}^\top\mathbf{u}\right\|_\infty \le \lambda$, then (13) is minimized at 0. This gives the dual in Equation (11).

We now show that by the Pythagorean theorem, we can project $\mathbf{P}_S^\perp\mathbf{y}$ in (11) rather than $\mathbf{y}$. In (11), recall that $\mathbf{u}$ is constrained to be in $\operatorname{colspan}(\mathbf{X}_S)^\perp$. Then, by the Pythagorean theorem, we have

$$\frac{1}{2}\|\mathbf{y}-\mathbf{u}\|_2^2 = \frac{1}{2}\left\|\mathbf{y}-\mathbf{P}_S^\perp\mathbf{y}+\mathbf{P}_S^\perp\mathbf{y}-\mathbf{u}\right\|_2^2 = \frac{1}{2}\left\|\mathbf{y}-\mathbf{P}_S^\perp\mathbf{y}\right\|_2^2 + \frac{1}{2}\left\|\mathbf{P}_S^\perp\mathbf{y}-\mathbf{u}\right\|_2^2,$$

since $\mathbf{y}-\mathbf{P}_S^\perp\mathbf{y} = \mathbf{P}_S\mathbf{y}$ is orthogonal to $\operatorname{colspan}(\mathbf{X}_S)^\perp$, and both $\mathbf{P}_S^\perp\mathbf{y}$ and $\mathbf{u}$ are in $\operatorname{colspan}(\mathbf{X}_S)^\perp$. The first term in the above does not depend on $\mathbf{u}$ and thus we may discard it. Our problem therefore reduces to projecting $\mathbf{P}_S^\perp\mathbf{y}$ onto $C_\lambda\cap\operatorname{colspan}(\mathbf{X}_S)^\perp$, rather than $\mathbf{y}$.

A.2 PROOF OF LEMMA 3.5

Proof of Lemma 3.5. Our approach is to reduce the projection of $\mathbf{P}_S^\perp\mathbf{y}$ onto the polytope defined by $C_\lambda\cap\operatorname{colspan}(\mathbf{X})^\perp$ to a projection onto an affine space.

We first argue that it suffices to project onto the faces of C_λ specified by T. For $\lambda>0$, feature indices $i\in[d]$, and signs $\pm$, we define the faces

$$F_{\lambda,i,\pm} := \{\mathbf{u}\in\mathbb{R}^n : \pm\langle\mathbf{X}_i,\mathbf{u}\rangle = \lambda\}$$

of C_λ. Let $\lambda = (1-\varepsilon)\|\mathbf{X}^\top\mathbf{P}_S^\perp\mathbf{y}\|_\infty$, for $\varepsilon>0$ to be chosen sufficiently small. Then clearly

$$(1-\varepsilon)\mathbf{P}_S^\perp\mathbf{y}\in C_\lambda\cap\operatorname{colspan}(\mathbf{X}_S)^\perp,$$

so

$$\min_{\mathbf{u}\in C_\lambda\cap\operatorname{colspan}(\mathbf{X}_S)^\perp}\left\|\mathbf{P}_S^\perp\mathbf{y}-\mathbf{u}\right\|_2^2 \le \left\|\mathbf{P}_S^\perp\mathbf{y}-(1-\varepsilon)\mathbf{P}_S^\perp\mathbf{y}\right\|_2^2 = \varepsilon^2\left\|\mathbf{P}_S^\perp\mathbf{y}\right\|_2^2.$$

In fact, $(1-\varepsilon)\mathbf{P}_S^\perp \mathbf{y}$ lies on the intersection of faces $F_{\lambda,i,\pm}$ for an appropriate choice of signs and $i \in T$. WLOG, we assume that these faces are just $F_{\lambda,i,+}$ for $i \in T$. Note also that for any $i \notin T$,

$$\begin{aligned}
\min_{\mathbf{u}\in F_{\lambda,i,\pm}} \left\|\mathbf{P}_S^\perp \mathbf{y} - \mathbf{u}\right\|_2^2 &\geq \min_{\mathbf{u}\in F_{\lambda,i,\pm}} \left\langle \mathbf{X}_i, \mathbf{P}_S^\perp \mathbf{y} - \mathbf{u}\right\rangle^2 && \text{(Cauchy–Schwarz, } \|\mathbf{X}_i\|_2 \leq 1) \\
&= \min_{\mathbf{u}\in F_{\lambda,i,\pm}} \left|\mathbf{X}_i^\top \mathbf{P}_S^\perp \mathbf{y} - \mathbf{X}_i^\top \mathbf{u}\right|^2 \\
&= \left(\left|\mathbf{X}_i^\top \mathbf{P}_S^\perp \mathbf{y}\right| - \lambda\right)^2 && (\mathbf{u} \in F_{\lambda,i,\pm}) \\
&\geq \left((1-\varepsilon)\left\|\mathbf{X}^\top \mathbf{P}_S^\perp \mathbf{y}\right\|_\infty - \left\|\mathbf{X}_T^\top \mathbf{P}_S^\perp \mathbf{y}\right\|_\infty\right)^2.
\end{aligned}$$

For all $\varepsilon < \varepsilon_0$, for ε_0 small enough, this is larger than $\varepsilon^2 \|\mathbf{P}_S^\perp \mathbf{y}\|_2^2$.

Thus, for ε small enough, $\mathbf{P}_S^\perp \mathbf{y}$ is closer to the faces $F_{\lambda,i,+}$ for $i \in T$ than any other face. Therefore, we set $\lambda_0 = (1-\varepsilon_0)\|\mathbf{X}^\top \mathbf{P}_S^\perp \mathbf{y}\|_\infty$.

Now, by the complementary slackness of the KKT conditions for the projection $\mathbf{u}$ of $\mathbf{P}_S^\perp \mathbf{y}$ onto C_λ, for each face of C_λ we either have that $\mathbf{u}$ lies on the face or that the projection does not change if we remove the face. For $i \notin T$, note that by the above calculation, the projection $\mathbf{u}$ cannot lie on $F_{\lambda,i,\pm}$, so $\mathbf{u}$ is simply the projection onto

$$C' = \left\{\mathbf{u} \in \mathbb{R}^n : \mathbf{X}_T^\top \mathbf{u} \leq \lambda \mathbf{1}_T\right\}.$$

By reversing the dual problem reasoning from before, the residual of the projection onto C' must lie on the column span of $\mathbf{X}_T$. □

A.3 Parameterization patterns and regularization

Proof of Lemma 3.2. The optimization problem on the left-hand side of Equation (7) with respect to $\boldsymbol{\beta}$ is equivalent to

$$\inf_{\boldsymbol{\beta}\in\mathbb{R}^d} \left(\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \inf_{\mathbf{w}\in\mathbb{R}^d} \left(\|\mathbf{w}\|_2^2 + \sum_{i\in\overline{S}} \frac{\boldsymbol{\beta}_i^2}{\mathbf{s}_i(\mathbf{w})^2} \right) \right). \tag{14}$$

If we define

$$\tilde{Q}^*(\mathbf{x}) = \inf_{\mathbf{w}\in\mathbb{R}^d} \left(\|\mathbf{w}\|_2^2 + \sum_{i\in\overline{S}} \frac{\mathbf{x}_i}{\mathbf{s}_i(\mathbf{w})^2} \right),$$

then the LHS of (7) and (14) are equivalent to $\inf_{\boldsymbol{\beta}\in\mathbb{R}^d}(\|\mathbf{X}\boldsymbol{\beta}-\mathbf{y}\|_2^2 + \frac{\lambda}{2}\tilde{Q}^*(\boldsymbol{\beta}\circ\boldsymbol{\beta}))$. Re-parameterizing the minimization problem in the definition of $\tilde{Q}^*(\mathbf{x})$ (by setting $\mathbf{q} = D(\mathbf{w})$), we obtain $\tilde{Q}^* = Q^*$. □

B Additional experiments

B.1 Visualization of selected features on MNIST

In Figure 5, we provide a visualization of the features selected by Sequential Attention as well as other baseline algorithms, to provide intuition on the nature of features selected by the algorithms. Similar visualizations on MNIST can be found in works such as Balın et al. (2019); Gui et al. (2019); Wojtas & Chen (2020); Lemhadri et al. (2021); Liao et al. (2021). Notably, Sequential Attention selects highly diverse pixels owing to its sequential feature selection process. Sequential LASSO also selects extremely similar pixels, as suggested by our theoretical analysis of Section 3. Curiously, OMP does not provide a good feature subset, demonstrating that OMP does not generalize well from least squares regression and generalized linear models to deep neural networks.

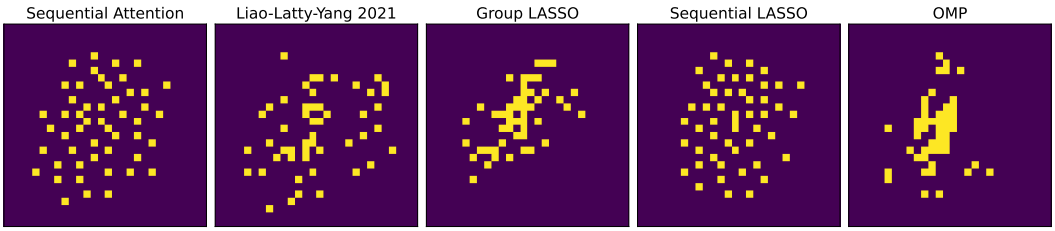

Figure 5: Visualizations of the $k = 50$ pixels selected by the feature selection algorithms on MNIST.

B.2 Further details on small-scale experiments

We start by presenting details about each of the datasets used for neural network feature selection in Balın et al. (2019); Lemhadri et al. (2021).

Table 1: Benchmark datasets in Balın et al. (2019); Lemhadri et al. (2021).

Dataset	# Examples	# Features	# Classes	Type
Mice Protein	1,080	77	8	Biology
MNIST	60,000	784	10	Image
MNIST-Fashion	60,000	784	10	Image
ISOLET	7,797	617	26	Speech
COIL-20	1,440	400	20	Image
Activity	5,744	561	6	Sensor

In Figure 3, the error bars are generated as the standard deviation over running the algorithm five times with different random seeds. The values used to generate the plot are provided in Table 2.

Table 2: Feature selection experimental results on small-scale datasets (see Figure 3 for a key). The numbers indicate the average accuracy on the test dataset, and the numbers in parentheses indicate the standard deviation.

Dataset	SA	LLY	GL	SL	OMP	CAE
Mice Protein	0.993 (0.008)	0.981 (0.005)	0.985 (0.005)	0.984 (0.008)	0.994 (0.008)	0.956 (0.012)
MNIST	0.956 (0.002)	0.944 (0.001)	0.937 (0.003)	0.959 (0.001)	0.912 (0.004)	0.909 (0.007)
MNIST-Fashion	0.854 (0.003)	0.843 (0.005)	0.834 (0.004)	0.854 (0.003)	0.829 (0.008)	0.839 (0.003)
ISOLET	0.920 (0.006)	0.866 (0.012)	0.906 (0.006)	0.920 (0.003)	0.727 (0.026)	0.893 (0.011)
COIL-20	0.997 (0.001)	0.994 (0.002)	0.997 (0.004)	0.988 (0.005)	0.967 (0.014)	0.972 (0.007)
Activity	0.931 (0.004)	0.897 (0.025)	0.933 (0.002)	0.931 (0.003)	0.905 (0.013)	0.921 (0.001)

B.2.1 Model accuracy on all features

To adjust for the differences between the values reported in Lemhadri et al. (2021) and ours due to factors such as the implementation framework, we list the accuracies obtained from training a model with all features in Table 4.

Table 3: Feature selection experimental results on small-scale datasets from Lemhadri et al. (2021).

Dataset	Fisher	HSIC-Lasso	PFA	LassoNet	Sequential Attention
Mice Protein	0.944	0.958	0.939	0.958	**0.993**
MNIST	0.813	0.870	0.873	0.873	**0.956**
MNIST-Fashion	0.671	0.785	0.793	0.800	**0.854**
ISOLET	0.793	0.877	0.863	0.885	**0.920**
COIL-20	0.986	0.972	0.975	0.991	**0.997**
Activity	0.769	0.829	0.779	0.849	**0.931**

Table 4: Accuracy of models trained on all features.

Dataset	All Features (Lemhadri et al., 2021)	All Features (this paper)
Mice Protein	0.990	0.963
MNIST	0.928	0.953
MNIST-Fashion	0.833	0.869
ISOLET	0.953	0.961
COIL-20	0.996	0.986
Activity	0.853	0.954

B.2.2 The Generalization of OMP to Neural Networks

As stated in Algorithm 2, it may be difficult to see exactly how OMP generalizes from a linear regression model to neural networks. To do this, first observe that OMP naturally generalizes to generalized linear models (GLMs) via the gradient of the link function, as shown in Elenberg et al. (2018). Then, to extend this to neural networks, we view the neural network as a GLM for any fixing of the hidden layer weights, and then we use the gradient of this GLM with respect to the inputs as the feature importance scores.

B.2.3 Efficiency Evaluation

In this section, we evaluate the efficiency of the Sequential Attention algorithm against our other baseline algorithms. We do so by fixing the number of epochs and batch size for all of the algorithms, and then evaluating the accuracy as well as the wall clock time of each of the algorithms. Figures 6 and 7 provide a visualization of the accuracy and wall clock time of feature selection, while Tables 6 and 7 provide the average and standard deviations. Table 5 provides the epochs and batch size fixed for these experiments.

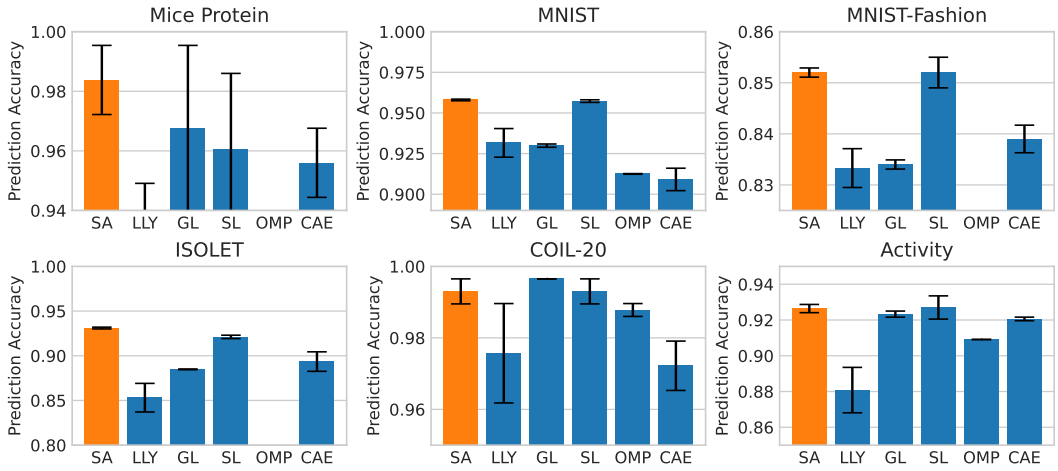

Figure 6: Feature selection accuracy for efficiency evaluation.

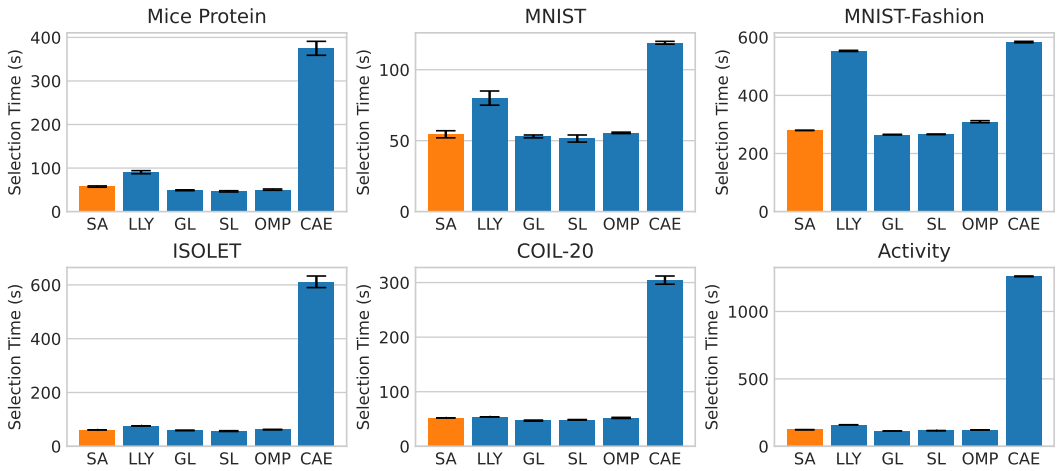

Figure 7: Feature selection wall clock time in seconds for efficiency evaluation.

Table 5: Feature selection epochs and batch size used to compare the efficiency of feature selection algorithms.

Dataset	Epochs	Batch Size
Mice Protein	2000	256
MNIST	50	256
MNIST-Fashion	250	128
ISOLET	500	256
COIL-20	1000	256
Activity	1000	512

Table 6: Feature selection accuracy for efficiency evaluation. The numbers indicate the average accuracy on the test dataset, and the numbers in parentheses indicate the standard deviation.

Dataset	SA	LLY	GL	SL	OMP	CAE
Mice Protein	0.993 (0.008)	0.981 (0.005)	0.985 (0.005)	0.984 (0.008)	0.994 (0.008)	0.956 (0.012)
MNIST	0.956 (0.002)	0.944 (0.001)	0.937 (0.003)	0.959 (0.001)	0.912 (0.004)	0.909 (0.007)
MNIST-Fashion	0.854 (0.003)	0.843 (0.005)	0.834 (0.004)	0.854 (0.003)	0.829 (0.008)	0.839 (0.003)
ISOLET	0.920 (0.006)	0.866 (0.012)	0.906 (0.006)	0.920 (0.003)	0.727 (0.026)	0.893 (0.011)
COIL-20	0.997 (0.001)	0.994 (0.002)	0.997 (0.004)	0.988 (0.005)	0.967 (0.014)	0.972 (0.007)
Activity	0.931 (0.004)	0.897 (0.025)	0.933 (0.002)	0.931 (0.003)	0.905 (0.013)	0.921 (0.001)

B.2.4 Notes on practical implementation

We make several remarks about practical implementation of Sequential Attention. First, as noted in Section 4.1, our practical implementation of Sequential Attention trains only one model rather than k models, by partitioning the training epochs into k parts. This yields a far more efficient running time than training k separate models. Second, we note that when there is a large number of candidate features d, then the softmax mask severely scales down the gradient updates to the model weights, which can lead to inefficient training. In these cases, it becomes important to prevent this by either

Table 7: Feature selection wall clock time in seconds for efficiency evaluation. The numbers indicate the average wall clock time on the test dataset, and the numbers in parentheses indicate the standard deviation.

Dataset	SA	LLY	GL	SL	OMP	CAE
Mice Protein	57.5	90.5	49.0	46.5	50.5	375.0
	(1.5)	(3.5)	(1.0)	(1.5)	(1.5)	(16.0)
MNIST	54.5	80.0	53.0	51.5	55.5	119.0
	(2.5)	(5.0)	(1.0)	(2.5)	(0.5)	(1.0)
MNIST-Fashion	279.5	553.0	265.0	266.0	309.5	583.5
	(0.5)	(2.0)	(1.0)	(1.0)	(3.5)	(2.5)
ISOLET	61.0	76.0	59.0	56.5	62.0	611.5
	(0.0)	(0.0)	(1.0)	(1.5)	(1.0)	(21.5)
COIL-20	52.0	54.0	47.0	48.5	52.0	304.5
	(0.0)	(0.0)	(1.0)	(0.5)	(1.0)	(7.5)
Activity	123.0	159.5	113.5	116.0	121.5	1260.5
	(1.0)	(0.5)	(0.5)	(0.0)	(0.5)	(2.5)

employing a temperature parameter in the softmax to counteract the small softmax values, or by adjusting the learning rate to be high enough. Note that these two approaches can be considered to be equivalent.

B.3 Large-Scale Experiments

In this section, we give additional details and discussion on our Criteo large dataset results. For this experiment, we use a dense neural network with 768, 256, and 128 neurons in each of the three hidden layers, with ReLU activations. In Figure 4, the error bars are generated as the standard deviation over running the algorithm three times with different random seeds, and the shadowed regions linearly interpolate between these error bars. The values used to generate the plot are provided in Tables 8 and 9.

We first note that this dataset is so large that it is expensive to make multiple passes through the dataset. Therefore, we modify the algorithms we use, both Sequential Attention and the other baselines, to make only one pass through the data by using separate fractions of the data for different "steps" of the algorithm. Hence, we select k features while only "training" one model.

Table 8: AUC of Criteo large experiments. SA is Sequential Attention, GL is generalized LASSO, and SL is Sequential LASSO. The values in the header for the LASSO methods is the ℓ_1 regularization strength used for each method.

k	SA	CMIM	GL, 10^{-1}	GL, 10^{-4}	SL, 10^{-1}	SL, 10^{-4}	Liao et al. (2021)
5	0.67232	0.63950	0.68342	0.50161	0.60278	0.67710	0.58300
	(0.00015)	(0.00076)	(0.00585)	(0.00227)	(0.04473)	(0.00873)	(0.06360)
10	0.70167	0.69402	0.71942	0.64262	0.62263	0.70964	0.68103
	(0.00060)	(0.00052)	(0.00059)	(0.00187)	(0.06097)	(0.00385)	(0.00137)
15	0.72659	0.72014	0.72392	0.65977	0.66203	0.72264	0.69762
	(0.00036)	(0.00067)	(0.00027)	(0.00125)	(0.04319)	(0.00213)	(0.00654)
20	0.72997	0.72232	0.72624	0.72085	0.70252	0.72668	0.71395
	(0.00066)	(0.00103)	(0.00330)	(0.00106)	(0.01985)	(0.00307)	(0.00467)
25	0.73281	0.72339	0.73072	0.73253	0.71764	0.73084	0.72057
	(0.00030)	(0.00042)	(0.00193)	(0.00091)	(0.00987)	(0.00070)	(0.00444)
30	0.73420	0.72622	0.73425	0.73390	0.72267	0.72988	0.72487
	(0.00046)	(0.00049)	(0.00081)	(0.00026)	(0.00663)	(0.00434)	(0.00223)
35	0.73495	0.73225	0.73058	0.73512	0.73029	0.73361	0.73078
	(0.00040)	(0.00024)	(0.00350)	(0.00058)	(0.00509)	(0.00037)	(0.00102)

Table 9: Log loss of Criteo large experiments. SA is Sequential Attention, GL is generalized LASSO, and SL is Sequential LASSO. The values in the header for the LASSO methods is the ℓ_1 regularization strength used for each method.

k	SA	CMIM	GL, 10^{-1}	GL, 10^{-4}	SL, 10^{-1}	SL, 10^{-4}	Liao et al. (2021)
5	0.14123	0.14323	0.14036	0.14519	0.14375	0.14073	0.14415
	(0.00005)	(0.00010)	(0.00046)	(0.00000)	(0.00163)	(0.00061)	(0.00146)
10	0.13883	0.13965	0.13747	0.14339	0.14263	0.13826	0.14082
	(0.00009)	(0.00008)	(0.00015)	(0.00019)	(0.00304)	(0.00032)	(0.00011)
15	0.13671	0.13745	0.13693	0.14227	0.14166	0.13713	0.13947
	(0.00007)	(0.00008)	(0.00005)	(0.00021)	(0.00322)	(0.00021)	(0.00050)
20	0.13633	0.13726	0.13693	0.13718	0.13891	0.13672	0.13806
	(0.00008)	(0.00010)	(0.00057)	(0.00004)	(0.00187)	(0.00035)	(0.00048)
25	0.13613	0.13718	0.13648	0.13604	0.13760	0.13628	0.13756
	(0.00013)	(0.00009)	(0.00051)	(0.00004)	(0.00099)	(0.00010)	(0.00043)
30	0.13596	0.13685	0.13593	0.13594	0.13751	0.13670	0.13697
	(0.00001)	(0.00004)	(0.00015)	(0.00005)	(0.00095)	(0.00080)	(0.00015)
35	0.13585	0.13617	0.13666	0.13580	0.13661	0.13603	0.13635
	(0.00002)	(0.00006)	(0.00073)	(0.00012)	(0.00096)	(0.00010)	(0.00005)

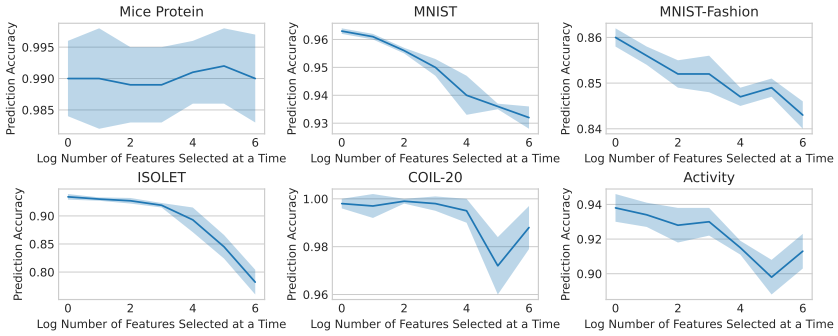

Figure 8: Sequential Attention with varying levels of adaptivity. We select 64 features for each model, but take 2^i feature in each round for increasing values of i. We plot model accuracy as a function of i.

B.4 THE ROLE OF ADAPTIVITY

We show in this section the effect of varying adaptivity on the quality of selected features in Sequential Attention. In the following experiments, we select 64 features on six datasets by selecting 2^i features at a time over a fixed number of epochs of training, for $i \in \{0, 1, 2, 3, 4, 5, 6\}$. That is, we investigate the following question: for a fixed budget of training epochs, what is the best way to allocate the training epochs over the rounds of the feature selection process? For most datasets, we find that feature selection quality decreases as we select more features at once. An exception is the mice protein dataset, which curiously exhibits the opposite trend, perhaps indicating that the features in the mice protein dataset are less redundant than in other datasets. Our results are summarized in Tables 8 and 10. We also illustrate the effect of adaptivity for Sequential Attention on MNIST in Figure 9. One observes that the selected pixels "clump together" as i increases, indicating a greater degree of redundancy.

Overall, our empirical results in this section suggest that adaptivity greatly enhances the quality of features selected by Sequential Attention, and more broadly, in feature selection algorithms.

Table 10: Sequential Attention with varying levels of adaptivity. We select 64 features for each model, but take 2^i feature in each round for increasing values of i. We show model accuracy as a function of i.

Dataset	$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
Mice Protein	0.990 (0.006)	0.990 (0.008)	0.989 (0.006)	0.989 (0.006)	0.991 (0.005)	0.992 (0.006)	0.990 (0.007)
MNIST	0.963 (0.001)	0.961 (0.001)	0.956 (0.001)	0.950 (0.003)	0.940 (0.007)	0.936 (0.001)	0.932 (0.004)
MNIST-Fashion	0.860 (0.002)	0.856 (0.002)	0.852 (0.003)	0.852 (0.004)	0.847 (0.002)	0.849 (0.002)	0.843 (0.003)
ISOLET	0.934 (0.005)	0.930 (0.003)	0.927 (0.005)	0.919 (0.004)	0.893 (0.022)	0.845 (0.021)	0.782 (0.022)
COIL-20	0.998 (0.002)	0.997 (0.005)	0.999 (0.001)	0.998 (0.003)	0.995 (0.005)	0.972 (0.012)	0.988 (0.009)
Activity	0.938 (0.008)	0.934 (0.007)	0.928 (0.010)	0.930 (0.008)	0.915 (0.004)	0.898 (0.010)	0.913 (0.010)

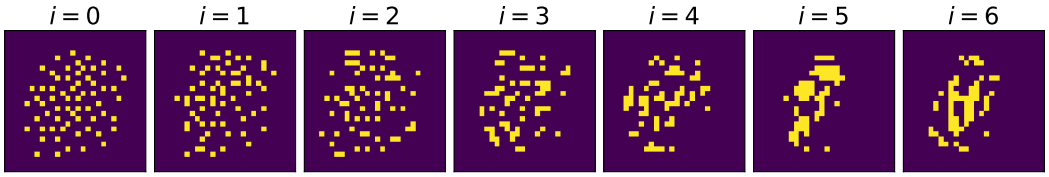

Figure 9: Sequential Attention with varying levels of adaptivity on the MNIST dataset. We select 64 features for each model, but take 2^i feature in each round for increasing values of i.

B.5 Variations on Hadamard product parameterization

We provide evaluations of various variations on the Hadamard product parameterization pattern, as described in Section 4.2. In Table 11, we provide the numerical values of the accuracies achieved.

Table 11: Accuracies achieved by Sequential Attention with various Hadamard product parameterization variations.

Dataset	Softmax	ℓ_1	ℓ_2	ℓ_1 Norm.	ℓ_2 Norm.
Mice Protein	0.990 (0.006)	0.993 (0.010)	0.993 (0.010)	0.994 (0.006)	0.988 (0.008)
MNIST	0.958 (0.002)	0.957 (0.001)	0.958 (0.002)	0.958 (0.001)	0.957 (0.001)
MNIST-Fashion	0.850 (0.002)	0.843 (0.004)	0.850 (0.003)	0.853 (0.001)	0.852 (0.002)
ISOLET	0.920 (0.003)	0.894 (0.014)	0.908 (0.009)	0.921 (0.003)	0.921 (0.003)
COIL-20	0.997 (0.004)	0.997 (0.004)	0.995 (0.006)	0.996 (0.005)	0.996 (0.004)
Activity	0.922 (0.005)	0.906 (0.015)	0.908 (0.012)	0.933 (0.010)	0.935 (0.007)

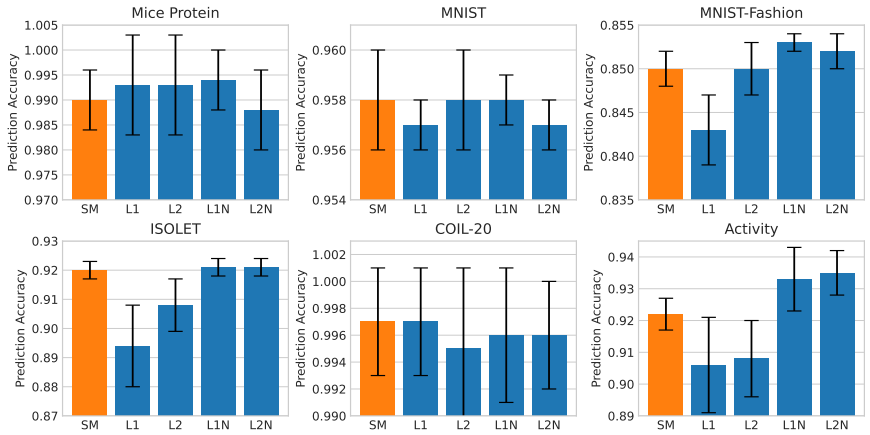

Figure 10: Accuracies achieved by Sequential Attention with various Hadamard product parameterization variations. Here, SM = softmax, L1 = ℓ_1, L2 = ℓ_2, L1N = ℓ_1 normalized, L2N = ℓ_2 normalized.

B.6 Approximation of marginal gains

In this section, we present our experimental results which show the correlations between weights computed by Sequential Attention and *marginal gains*.

Definition B.1 (Marginal gains)**.** Let $\ell : 2^{[d]} \to \mathbb{R}$ be a set function defined on the ground set $[d]$. Then, for a set $S \subseteq [n]$ and $i \notin S$, the *marginal gain of* i *with respect to* S is

$$\ell(S \cup \{i\}) - \ell(S).$$

In the setting of feature selection, marginal gains are often considered for measuring the importance of candidate features i given a set S of features that have already be selected, by using the set function ℓ corresponding to the model loss when trained on a subset of features. It is known that greedily selecting features based on the marginal gain performs well in both theory Das & Kempe (2011); Elenberg et al. (2018) and practice Das et al. (2022). However, these scores are extremely expensive to compute in practice, as they require training a model for every feature considered.

In this experiment, we first compute the top k features selected by Sequential Attention for $k \in \{0, 9, 49\}$ on the MNIST dataset. We then compute the marginal gains as well as the attention weights according to Sequential Attention conditioned on these features, and compare the two sets of scores. The Sequential Attention weights are computed by only applying the attention softmax mask over the $d - k$ unselected features, while the marginal gains are computed by explicitly training a model for each candidate feature to be added to the preselected k features. Because our Sequential Attention algorithm is motivated by an efficient implementation of the greedy selection algorithm that uses marginal gains (see Section 1), one may expect to find that these two sets of scores are similar, or correlated in some sense. We show this by plotting the top scores according to the two sets of scores, and by computing the Spearman correlation between the two sets of scores.

In the first and second rows of Figure 11, we see that the top 50 pixels according to the marginal gains and attention weights are visually similar, avoiding previously selected regions and finding new areas which are now important. In the third row, we quantify the similarity via the Spearman correlation between the feature rankings. Although the correlations degrade as we select more features, this is expected, as one expects marginal gains to become similar among remaining features, after removing the most important features.

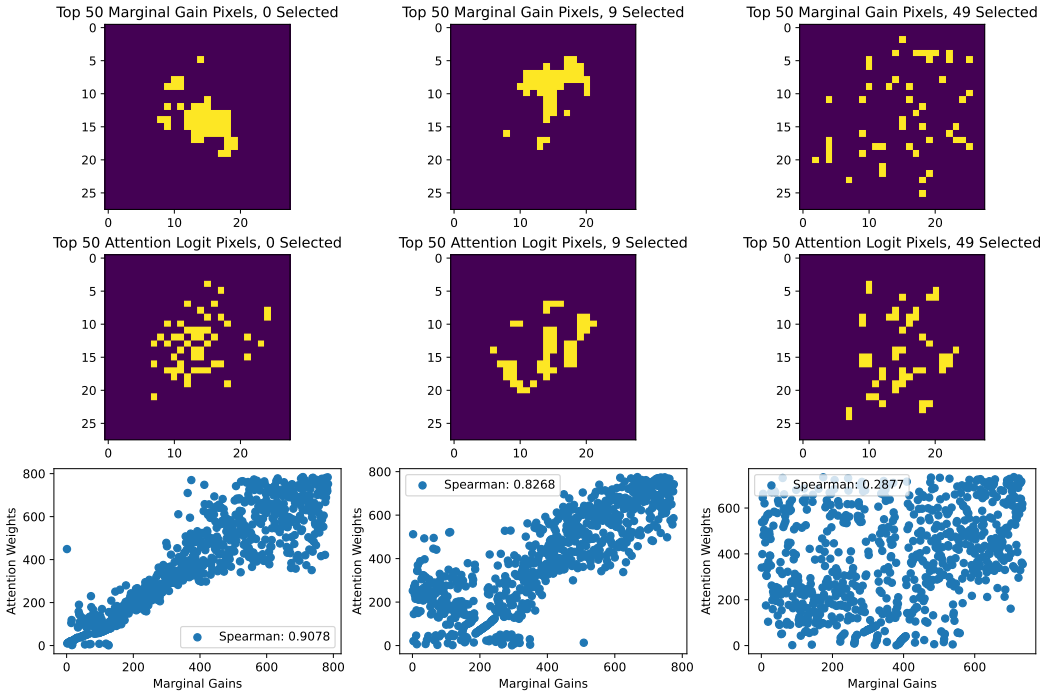

Figure 11: **Marginal gain experiments.** The first and second rows show that top 50 features according to marginal gain and Sequential Attention, respectively. The third row shows the Spearman correlation between the computed sets of scores.