An Efficient Pruner for Large Language Model with Theoretical Guarantee

Canhong Wen [1] **Yihong Zuo** [2] **Wenliang Pan** [3]

Abstract

Large Language Models (LLMs) have showcased remarkable performance across a range of tasks but are hindered by their massive parameter sizes, which impose significant computational and storage demands. Pruning has emerged as an effective solution to reduce model size, but traditional methods often involve inefficient retraining or rely on heuristic-based one-shot approaches that lack theoretical guarantees. In this paper, we reformulate the pruning problem as an ℓ_0-penalized optimization problem and propose a monotone accelerated Iterative Hard Thresholding (mAIHT) method. Our approach combines solid theoretical foundations with practical effectiveness, offering a detailed theoretical analysis that covers convergence, convergence rates, and risk upper bounds. Through extensive experiments, we demonstrate that mAIHT outperforms state-of-the-art pruning techniques by effectively pruning the LLaMA-7B model across various evaluation metrics.

1. Introduction

Large Language Models (LLMs) have demonstrated exceptional capabilities in tasks such as reasoning, question answering, text generation, and sentiment analysis (Kojima et al., 2022; Wei et al., 2022; Achiam et al., 2023). However, the growing size of these models, with ever-increasing parameter counts, imposes substantial demands on the computational and storage resources of hardware devices. For example, running the LLaMA 3.1 405B model (Dubey et al., 2024) requires a minimum of 486GB of GPU memory in 8-bit mode, demanding at least eight 80GB A100 GPUs.

[1]Department of Statistics and Finance, School of Management, University of Science and Technology of China, Hefei, China [2]School of Gifted Young, University of Science and Technology of China, Hefei, China [3]State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China. Correspondence to: Wenliang Pan <panwliang@amss.ac.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

Network pruning (LeCun et al., 1989; Hassibi & Stork, 1992) is a well-established method for addressing these challenges by removing redundant parameters while preserving model performance. Early pruning techniques often involved retraining after pruning (Han et al., 2015; Liu et al., 2018), a process that became increasingly inefficient as model size grew. Moreover, efficient fine-tuning methods like LoRA (Hu et al., 2021) cannot be easily applied to pruned sparse models.

To mitigate these issues, recent research has shifted towards one-shot pruning methods, which eliminate parameters based on calibration data while retaining performance. These methods often view pruning as a layer-wise subset selection problem, leading to the development of various optimization-based approaches (Frantar & Alistarh, 2022; Benbaki et al., 2023). However, the sheer scale of modern LLMs makes many traditional pruning methods impractical, and most current approaches rely on heuristic methods. For instance, SparseGPT (Frantar & Alistarh, 2023) leverages the Optimal Brain Surgeon (OBS) framework (Hassibi & Stork, 1992) and relies on heuristic approximations of the loss change, while Wanda (Sun et al., 2023) uses the product of weight magnitude and input activation norms to guide pruning. Although these methods are efficient, they lack theoretical guarantees of optimality due to their reliance on intuition and approximations.

Some recent studies have also explored efficient optimization-based approaches like ADMM (Alternating Direction Method of Multipliers)-based pruning (Boža, 2024; Meng et al., 2024a) and FISTA (Fast Iterative Shrinkage-Thresholding Algorithm)-based Pruning (Zhao et al., 2024). However, these approaches still lack theoretical guarantees or introduce bias by relaxing the ℓ_1-norm problem (Bertsimas et al., 2016).

In this paper, we introduce mAIHT Pruner, a novel one-shot layer-wise pruning method based on ℓ_0-penalized optimization. Our contribution are as follows:

1. We reformulate the pruning problem as an ℓ_0-penalized optimization problem and propose a monotone accelerated iterative hard thresholding (mAIHT) algorithm to solve it. This technique not only accelerates the convergence of the traditional iterative hard threshold-

ing (IHT) method but also significantly improves the solution quality in practice. Furthermore, we design an adaptive selection of the penalty coefficient, allowing our algorithm to control the sparsity of the solution precisely.

2. We conduct rigorous theoretical analysis to explore the convergence and statistical properties of our algorithms. Specifically, We establish both the convergence and the rate of convergence. Furthermore, we analyze the risk upper bounds of the algorithms, bridging the gap between the experimental results and the theoretical analysis.

3. In our experiments, we benchmark mAIHT against the latest state-of-the-art methods through the pruning of the LLaMA-7B model (Touvron et al., 2023). The findings indicate that mAIHT outperforms its counterparts, delivering superior pruning performance across low to moderate sparsity levels.

Notations. Denote by $[n] = \{1, \ldots, n\}$ for a positive integer n. Denote the cardinality of a set S by $|S|$. We use bold uppercase and lowercase letters to represent matrices and vectors, respectively. For a vector $\boldsymbol{a}$, denote the ℓ_p norm of $\boldsymbol{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$ as $\|\boldsymbol{a}\|_p = (\sum_{i=1}^n a_i^p)^{\frac{1}{p}}$, where $0 \leq p \leq \infty$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{m\times n}$, we denote the ℓ_0 pseudo-norm as $\|\mathbf{A}\|_0$, which is equal to the number of non-zero entries, Frobenius norm as $\|\mathbf{A}\|_F = (\sum_{ij} a_{ij}^2)^{\frac{1}{2}}$, the spectral norm as $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})$, where $\sigma_{\max}(\mathbf{A})$ is the largest singular value of matrix $\mathbf{A}$, and ℓ_1 norm as $\|\mathbf{A}\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^m |a_{ij}|$. The support of $\mathbf{A}$ is denoted by $\mathrm{Supp}(\mathbf{A})$ which is the index set $\{(i,j) : a_{ij} \neq 0\}$. Given $i \in [m], j \in [n]$ and a set $S \subset [m] \times [n]$, $\mathbf{A}_{ij}$ or $\mathbf{A}_{i,j}$ denote a_{ij}. $\mathbf{A}_S$ and $\mathrm{P}_S(\mathbf{A})$ denote projection of $\mathbf{A}$ onto S, which means retaining the components of $\mathbf{A}$ indexed by S and zeroing out the remaining components of $\mathbf{A}$. Denote $\mathbf{I}_n$ as the identity matrix of dimension n. Let $\mathbf{A} \in \mathbb{R}^{m_a\times n_a}$ and $\mathbf{B} \in \mathbb{R}^{m_b\times n_b}$. We define the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$ as $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{m_a m_b \times n_a n_b}$ such that $(\mathbf{A} \otimes \mathbf{B})_{(i_a-1)m_b+i_b,(j_a-1)n_b+j_b}$, is equal to $\mathbf{A}_{i_a j_a}\mathbf{B}_{i_b j_b}$ with $i_a \in [m_a], j_a \in [n_a], i_b \in [m_b], j_b \in [n_b]$. For two positive sequences $\{a_n\}$ and $\{b_n\}$, write $a_n = O(b_n)$ if there exists some constant $C > 0$ such that $a_n \leq Cb_n$ for all n. Write $a_n = o(b_n)$ if $a_n/b_n \to 0$ as $n \to \infty$.

2. Method and Algorithm

2.1. Model formulation

Post-training compression is typically achieved by dividing the full-model compression problem into smaller, layer-wise subproblems. Given a calibration input, we evaluate the quality of the solution using the residual sum of squares between the pruned and original outputs. To be specific, let $\widehat{\mathbf{W}}_l \in \mathbb{R}^{d_1\times d_2}$ denote the pre-trained weight matrix of layer ℓ, where d_1 and d_2 denote the input and output dimension of layer ℓ. For simplicity, we write $\widehat{\mathbf{W}}_l$ as $\widehat{\mathbf{W}}$ in what follows. Given a calibration input $\mathbf{X} \in \mathbb{R}^{N\times d_1}$ of size N and a pre-specified sparsity level k, the layer-wise pruning problem can be formulated as a ℓ_0-constrained optimization problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{d_1\times d_2}} \quad \frac{1}{2}\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}\|_F^2, \quad s.t. \quad \|\mathbf{W}\|_0 \leq k, \tag{1}$$

where $\|\mathbf{W}\|_0 = \sum_{ij} I(|w_{ij}| > 0)$ is the ℓ_0 norm counting the number of nonzero elements in $\mathbf{W} = (w_{ij})$. This formulation is first introduced in (Meng et al., 2024a) and an operator-splitting technique is employed to solve the optimization problem.

Rather than directly optimizing problem equation 1, we consider the associated Lagrangian formulation as follows

$$\min_{\mathbf{W}\in\mathbb{R}^{d_1\times d_2}} \quad \mathcal{L}(\mathbf{W}) := \frac{1}{2}\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_0, \tag{2}$$

where $\lambda \geq 0$ is the regularization parameter controlling the sparsity of the solution.

2.2. Naive iterative hard-thresholding algorithm

To solve the optimization problem equation 2, the most tricky component is to deal with the ℓ_0-regularized term, a non-convex and non-smooth function. To overcome the computational difficulty, we consider using the proximal gradient method (Parikh et al., 2014), which updates the weight matrix via a proximal operator. In particular, for any function $g(\cdot)$ defined on $\mathbf{W} \in \mathbb{R}^{d_1\times d_2}$, the proximal operator of $g(\cdot)$ is defined by

$$\mathrm{Prox}_{g(\cdot)}(\mathbf{W}) = \arg\min_{\mathbf{V}\in\mathbb{R}^{d_1\times d_2}} \left\{ g(\mathbf{V}) + \frac{1}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 \right\}.$$

Denote $f(\mathbf{W}) = \frac{1}{2}\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}\|_F^2$ and $h(\mathbf{W}) = \lambda\|\mathbf{W}\|_0$, then the objective function in equation 2 can be rewritten as $\mathcal{L}(\mathbf{W}) = f(\mathbf{W}) + h(\mathbf{W})$. The main challenge in applying the proximal gradient method in solving equation 2 is handling the function $h(\cdot)$. Therefore, we first present the proximal operator of $h(\mathbf{W}) = \lambda\|\mathbf{W}\|_0$ in the following proposition.

Proposition 2.1. *For any nonnegative real number λ, the proximal operator of $h(\mathbf{W}) = \lambda\|\mathbf{W}\|_0$ can be expressed as*

$$\mathrm{H}_{\sqrt{2\lambda}}(\mathbf{W}) \in \mathrm{Prox}_{\lambda\|\cdot\|_0}(\mathbf{W}),$$

where $\mathrm{H}_{\sqrt{2\lambda}}(\cdot)$ is the element wise hard thresholding operator, i.e., $\mathrm{H}_{\sqrt{2\lambda}}(\mathbf{W})_{ij} = w_{ij}$ if $|w_{ij}| > \sqrt{2\lambda}$ and $\mathrm{H}_{\sqrt{2\lambda}}(\mathbf{W})_{ij} = 0$ otherwise.

Based on Proposition 2.1, given the current estimate $\mathbf{W}^k$ at the k-th iteration, we can derive the proximal gradient update of $\mathcal{L}(\mathbf{W})$ as

$$\begin{aligned}\mathbf{W}^{k+1} &= \mathrm{H}_{\sqrt{2\alpha\lambda}}(\mathbf{W}^k - \alpha\nabla f(\mathbf{W}^k)) \\ &= \mathrm{H}_{\sqrt{2\alpha\lambda}}(\mathbf{W}^k + \alpha\mathbf{X}^{\mathrm{T}}\mathbf{X}(\widehat{\mathbf{W}} - \mathbf{W}^k)),\end{aligned} \tag{3}$$

where α denotes the step size. The update scheme in (3) is related to the well-known iterative hard-thresholding (IHT) algorithm (Blumensath & Davies, 2008; 2009).

Remark 2.2. By setting $\mathbf{W}^0 = \widehat{\mathbf{W}}$, we have $\mathbf{W}^1 = \mathrm{H}_{\sqrt{2\alpha^k\lambda}}(\mathbf{W}^0)$, which is the magnitude pruning proposed by (Han et al., 2015), a well-estabilishing pruning technique in neural networks. (Frantar & Alistarh, 2023) finds that magnitude pruning fails dramatically on LLMs even with relatively low levels of sparsity.

2.3. Monotone accelerated iterative hard-thresholding algorithm

In practice, the convergence rate of the naive IHT algorithm is relatively slow, leading to significant computational overhead when pruning LLMs. To address this, we employ the monotone accelerated technique (Li & Lin, 2015) by introducing a momentum term. Following (Beck & Teboulle, 2009), we define the momentum term as a very specific linear combination between the previous two points and the corresponding proximal gradient map. At the $(k+1)$-th iteration, this momentum term is added to the current solution $\mathbf{W}^k$ before performing the proximal gradient descent step. Furthermore, to determine when to accelerate and guarantee the loss function does not increase at the $(k+1)$-th iteration, we use a proximal gradient step as a monitoring variable, denoted as $\mathbf{V}^{k+1}$. The acceleration step is accepted only when $\mathcal{L}(\mathbf{Z}^{k+1}) \leq \mathcal{L}(\mathbf{V}^{k+1})$. This monitoring mechanism ensures the sufficient descent property, meaning there exists a small constant δ such that $\mathcal{L}(\mathbf{W}^{k+1}) \leq \mathcal{L}(\mathbf{V}^{k+1}) \leq \mathcal{L}(\mathbf{W}^k) - \delta\|\mathbf{W}^k - \mathbf{V}^{k+1}\|_F^2$. We summarise this monotone accelerated update for the $(k+1)$-th iteration as follows:

$$\begin{aligned}&\mathbf{Y}^k = \mathbf{W}^k + \frac{t^{k-1}}{t^k}(\mathbf{Z}^k - \mathbf{W}^k) + \frac{t^{k-1}-1}{t^k}(\mathbf{W}^k - \mathbf{W}^{k-1}), \\ &\mathbf{Z}^{k+1} = \mathrm{H}_{\sqrt{2\alpha_1\lambda}}(\mathbf{Y}^k - \alpha_1\nabla f(\mathbf{Y}^k)), \\ &\mathbf{V}^{k+1} = \mathrm{H}_{\sqrt{2\alpha_2\lambda}}(\mathbf{W}^k - \alpha_2\nabla f(\mathbf{W}^k)), \\ &t^{k+1} = \frac{\sqrt{4(t^k)^2+1}+1}{2}, \\ &\mathbf{W}^{k+1} = \left\{ \begin{array}{ll} \mathbf{Z}^{k+1}, & \text{if } \mathcal{L}(\mathbf{Z}^{k+1}) \leq \mathcal{L}(\mathbf{V}^{k+1}), \\ \mathbf{V}^{k+1}, & \text{otherwise.} \end{array} \right\}\end{aligned} \tag{4}$$

Here $\mathbf{Y}^k$ represents the momentum term, $\mathbf{Z}^{k+1}$ denotes the proximal gradient map of the momentum term, and $\mathbf{V}^{k+1}$ is the monitoring variable. Since the proximal gradient step essentially serves as a hard-threholsing rule, we refer to this proposed algorithm as the monotone accelerated iterative hard-thresholding (mAIHT) algorithm.

2.4. Adaptive IHT/mAIHT algorithm

Due to the non-convexity of problems in (1) and (2), they are not equivalent. However, in the context of one-shot pruning for neural networks, our primary target is to directly control the sparsity of the solution, as formulated in problem (1). To address this, we propose an adaptive method for determining an optimal λ corresponding to a pre-specified sparsity level. To be specific, we increase λ when the sparsity of the current solution, $\mathbf{W}^k$, is below the desired sparsity, and decrease it when the sparsity exceeds the target. Additionally, the magnitude of increase or decrease is designed to be positively correlated with the difference between the current and desired sparsity levels. At the $(k+1)$-th iteration, the update rule for λ is given by

$$\lambda^{k+1} = \lambda^k\left(1 + \frac{\|\mathbf{W}^k\|_0 - s}{d_1 d_2}\right),$$

where s denotes the target sparsity level. The initial value of λ can be set to retain 99% of the elements, i.e. $\lambda_{\text{init}} = Q^2_{0.01}(|\widehat{\mathbf{W}}|)/2\alpha$, where $Q_{0.01}(|\widehat{\mathbf{W}}|)$ represents the 0.01 quantile of the absolute values of all elements in $\widehat{\mathbf{W}}$ and α is the input step size. With this initialization and adaptive updating process, a gradual and controlable pruning can be achieved.

Upon terminating the mAIHT or IHT iteration, we obtain the selected support through a single projection operation $\mathrm{P}_s(\mathbf{W}^k)$, where $\mathrm{P}_s(\mathbf{W})$ represents the operation of retaining the largest s elements of the absolute values from $\mathbf{W}$ and setting the remaining elements to zero. At this stage, the optimal strategy at this point is to backsolve for the least squares solution on the current support. However, solving this exactly is computational expensive, as it requires computing the inverse of the $S \times S$ submatrix of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ for each column support S of $\mathbf{W}^k$. This results in a time complexity of $O(d_2 d_1^3)$. To mitigate this issue, we adopt the projected gradient descent algorithm (Frantar & Alistarh, 2023) to refine the final solution. This strategy proves to be highly efficient, as the proximal gradient process already provides a solution that is close to the optimum.

The detailed steps of the adaptive algorithm are outlined in Algorithm 1, which summarizes the process described above.

Remark 2.3. (Stability improvement) In practical applications, we observe that modifying the problem (2) to the

Algorithm 1 Adaptive IHT/mAIHT algorithm for layer-wise pruning

Input: step size $\alpha > 0$, λ_{init}, sparsity level s, maximum iteration time $k^1_{\max}$, $k^2_{\max}$.
Initialize: $\mathbf{W}^0 = \mathbf{W}^1 = \mathbf{Z}^1 = \widehat{\mathbf{W}}$, $t^1 = 1, t^0 = 0$, $\lambda^0 = \lambda_{\text{init}}$.
for $k = 1, 2, \cdots, k^1_{\max} - 1$ **do**
  $\lambda^{k+1} \leftarrow \lambda^k(1 + \frac{\|\mathbf{W}^k\|_0 - s}{d_1 d_2})$.
  Update $\mathbf{W}^{k+1}$ by (3) or (4) with $\lambda = \lambda^{k+1}$.
end for
$T \leftarrow \text{Supp}(\text{P}_s(\mathbf{W}^{t_1}))$.
for $k = 0, 1, 2, \cdots, k^2_{\max} - 1$ **do**
  $\mathbf{W}^{t_1+k+1} \leftarrow \text{P}_T(\mathbf{W}^{t_1+k} - \alpha\nabla f(\mathbf{W}^{t_1+k}))$.
end for
Output: $\mathbf{W}^{k^1_{\max}+k^2_{\max}}$

following formulation

$$\min_{\mathbf{W}\in\mathbb{R}^{d_1\times d_2}} \quad \mathcal{L}(\mathbf{W}) := \frac{1}{2}\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}\|_F^2 + \frac{\mu}{2}\|\widehat{\mathbf{W}} - \mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_0 \tag{5}$$

can improve the quality of the solution, where μ is a small positive constant. The additional ℓ_2 regularization term can adjust the trade-off between bias and variance, which can enhance predictive accuracy by preventing overfitting. Moreover, this modified problem is equivalent to the original problem (2) taking $\mathbf{X} \leftarrow (\mathbf{X}^{\text{T}}\mathbf{X} + \mu\mathbf{I})^{\frac{1}{2}}$, and therefore the theoretical properties of the problem (2) can be directly transferred to this modified problem.

3. Theoretical Analysis

In the following discussion, we consider the step size α and λ to be fixed across the implementation. Let $\alpha < 1/L$, where $L = \|\mathbf{X}\|_2^2$ is the Lipschitz constant of the gradient of f. We use IHT representing the iteration (3) and mAIHT representing the iteration (4).

3.1. Convergence and convergence rate

Intuitively, motivated by equation (3), we define $\mathbf{W}$ as an α-fixed point if $\mathbf{W}^k = \mathbf{W}$ leads to $\mathbf{W}^{k+1} = \mathbf{W}$. This can be regarded as a characterization of the stationarity of Problem (2). The formal definition of an α-fixed point is presented as follows.

Definition 3.1. Given an $\alpha > 0$, the matrix $W \in \mathbb{R}^{d_1\times d_2}$ is said to be an α-fixed point of Problem (2) if it satisfies the following fixed point equation

$$\mathbf{W} \in \text{Prox}_{\alpha\lambda\|\cdot\|_0}(\mathbf{W} + \alpha\mathbf{X}^{\text{T}}(\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W})).$$

We denote the set of all α-fixed points as $F(\alpha)$.

It is important to note that $\mathbf{W}$ is an α-fixed point if and only if

$$\begin{cases} (\mathbf{X}^{\text{T}}\mathbf{X}(\widehat{\mathbf{W}} - \mathbf{W}))_{ij} = 0\ ,\ |\mathbf{W}_{ij}| \geq \sqrt{2\alpha\lambda}, \\ \qquad\qquad\qquad\qquad \text{for } (i,j) \in \text{Supp}(\mathbf{W}), \\ |(\mathbf{X}^{\text{T}}\mathbf{X}(\widehat{\mathbf{W}} - \mathbf{W}))_{ij}| \leq \sqrt{\frac{2\lambda}{\alpha}}, \\ \qquad\qquad\qquad\qquad \text{for } (i,j) \notin \text{Supp}(\mathbf{W}). \end{cases}$$

Then for any $0 \leq \alpha_1 \leq \alpha_2$, we have

$$F(\alpha_1) \supset F(\alpha_2).$$

This implies that a larger step size can reduce the size of the fixed point set, leading to a more concentrated accumulation points. As a result, the algorithm may converge to a more stable solution, as the fixed points will be less spread out. Based on the above definition, we establish the convergence properties of the two algorithms. We first present a lemma to highlight the key descent property of the algorithms.

Lemma 3.2. *Let $\alpha, \alpha_1, \alpha_2 < \frac{1}{L}$ and $\{\mathbf{W}^k\}_{k=0}^{\infty}$ be the sequence generated by IHT or mAIHT. Then $\mathcal{L}(\mathbf{W}^k)$ is a descent sequence.*

Now we present the convergence results for both IHT and mAIHT algorithms.

Theorem 3.3 (Convergence Theorem)**.**

1. *Let $\alpha < \frac{1}{L}$ and $\{\mathbf{W}^k\}_{k=0}^{\infty}$ be the sequence generated by IHT. Then there is a α-fixed point $\mathbf{W}^*$, such that $\mathbf{W}^k \to \mathbf{W}^*$.*

2. *Let $\alpha_1, \alpha_2 < \frac{1}{L}$. $\{\mathbf{W}^k\}_{k=0}^{\infty}$ and $\{\mathbf{V}^k\}_{k=0}^{\infty}$ generated by mAIHT are bounded and any accumulation point of $\{\mathbf{W}^k\}_{k=0}^{\infty}$ is an α_2-fixed point.*

Li & Lin (2015) proved that for the general mAIHT algorithm, any accumulation point $\mathbf{W}^*$ of $\{\mathbf{W}^k\}_{k=0}^{\infty}$ will satisfy $0 \in \partial\mathcal{L}(\mathbf{W}^*)$, where $\partial\mathcal{L}$ denotes the sub-gradient set of $\mathcal{L}$. The condition $0 \in \partial\mathcal{L}(\mathbf{W}^*)$ is equivalent to $(\mathbf{X}^{\text{T}}\mathbf{X}(\widehat{\mathbf{W}} - \mathbf{W}^*))_{ij} = 0$ for $(i,j) \in \text{Supp}(\mathbf{W}^*)$. By comparing this condition with ours, we observe that the set composed of all α_2-fixed points is a subset of the points whose subgradients contain 0. This indicates that our result cannot be simply viewed as a special case of theirs, but rather a stronger result.

Next, we present the convergence rates of the two algorithms. According to Lemma 3.2, the sequences $\{\mathcal{L}(W^k)\}_{k=0}^{\infty}$ generated by both algorithms are decreasing sequences, and thus have limits $\mathcal{L}^*$. Therefore, we proceed to analyze the convergence rates of these sequences.

Theorem 3.4 (Convergence rate)**.**

1. *Let $\mathbf{W}^*$ be the limit of $\{\mathbf{W}^k\}_{k=0}^{\infty}$ generated by IHT, then $\mathcal{L}(\mathbf{W}^k)$ is a descent sequence with limit $\mathcal{L}^*$ and there exists a k_1 such that for all $k > k_1$, we have*

$$\mathcal{L}(\mathbf{W}^k) - \mathcal{L}^* \leq \frac{\|\mathbf{W}^{k_1} - \mathbf{W}^*\|_F^2}{2\alpha(k - k_1)}.$$

2. *Let $\{\mathbf{W}^k\}_{k=0}^{\infty}$ be generated by mAIHT, then $\mathcal{L}(\mathbf{W}^k)$ is a descent sequence that converges to its limit $\mathcal{L}^*$ at least in the sub-linear convergence rate after finite iterations, i.e. there exist $k_2 > 0$ and a constant C, such that for all $k > k_2$, we have*

$$\mathcal{L}(\mathbf{W}^k) - \mathcal{L}^* \leq \left(\frac{C}{(k - k_2)d(1 - 2\theta)}\right)^{\frac{1}{1-2\theta}},$$

where $\theta \in (0, \frac{1}{2})$ and $d = \min\left\{\frac{\alpha_2(1-L\alpha_2)}{4C(1+L\alpha_2)^2}, \frac{C}{1-2\theta}\left(2^{\frac{2\theta-1}{2\theta-2}} - 1\right)(\mathcal{L}(\mathbf{V}^1) - \mathcal{L}^*)^{2\theta-1}\right\}$.

Theorem 3.4 states that the theoretical asymptotic convergence rate of IHT is $O(1/k)$, while that of mAIHT is $O((1/k)^{\beta})$, where $\beta \in (1, \infty)$ is a constant. This shows that the theoretical convergence rate of mAIHT is not be worse than that of IHT. In addition, Theorem 3.4 suggests that a larger step size can lead to a faster convergence rate. As mentioned earlier, a larger step size can help reduce the size of the fixed point set. Combining these insights, we can conclude that a larger step size will endow the stable points with better theoretical properties. Hence, it is necessary to choose a relatively large step size in practice to exploit these benefits and improve the overall performance of the algorithm. However, one must also consider practical constraints such as potential instability or overshooting when setting the step size too large.

3.2. Risk upper bounds

Since reconstruction error is a crucial metric in the pruning task, we define the population risk as $R(\mathbf{W}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_x}(\|\boldsymbol{x}^{\mathrm{T}}\mathbf{W} - \boldsymbol{x}^{\mathrm{T}}\widehat{\mathbf{W}}\|^2)$, where $\mathcal{D}_x$ is the distribution of the calibration data. The goal of this section is to study the population risk of $\mathbf{W}^k$ obtained by IHT or mAIHT with finite calibration samples. In the following discussion, we assume that $\mathcal{D}_x$ is a bounded distribution i.e. there exists a M_x such that $\mathbb{P}(\|\boldsymbol{x}\|_2 \leq M_x) = 1$. We first define the restricted isometry property (RIP) introduced by Candes & Tao (2005b).

Definition 3.5. (Candes & Tao, 2005b; Candes, 2008) For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, the Kth restricted isometry constant (RIC), denoted by δ_K, is the smallest number $\delta \geq 0$ such that

$$(1 - \delta)\|\boldsymbol{x}\|_2^2 \leq \|\mathbf{A}\boldsymbol{x}\|_2^2 \leq (1 + \delta)\|\boldsymbol{x}\|_2^2$$

holds for all K-sparse vector $\boldsymbol{x} \in \mathbb{R}^n$

We now present the risk upper bounds for the IHT and mAIHT methods under RIP assumption.

Theorem 3.6 (Risk upper bound for IHT)**.** *Let $\mathbf{X} = \frac{1}{\sqrt{n}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $\mathbb{P}(\|\boldsymbol{x}_1\|_2 \leq M_x) = 1$. For $\{\mathbf{W}^n\}$ generated by IHT with step size $\alpha < \frac{1}{L}$ and λ. Assume $\alpha = O(1)$, $\lambda = O(1)$ and δ satisfies $\log(1/\delta) = O(n)$. For integer s, assume with probability $1 - \zeta$: (1) $\mathbf{W}^* = \lim \mathbf{W}^n$, $s^* = \|\mathbf{W}^*\|_0$ and $s^* \geq s$; (2) the restricted isometry constant of the matrix $\mathbf{I}_{d_2} \otimes \sqrt{\alpha}\mathbf{X}$ satisfies $\delta_{s^*+s} \leq \tau < 0.5$. Then with probability at least $1 - \delta - \zeta$, there exists a K. For any $k > K$, it holds that,*

$$R(\mathbf{W}^k) \leq C^2 \min_{\mathbf{W}, \|\mathbf{W}\|_0 \leq s} R(\mathbf{W}) + O\left(\frac{1}{1 - 2\tau}\right)(2\delta_{s^*+s})^{k-K} + O\left(\sqrt{\frac{\log(2d_1^2/\delta)}{n(1 - 2\tau)^4}}\right),$$

where $C = 1 + \frac{2(1+\delta_{s^*+s})}{1-2\delta_{s^*+s}}$.

Corollary 3.7. *In Theorem 3.6, suppose $\log(1/\delta) = o(n)$, $\delta = o(1)$, $\zeta = o(1)$ and $\frac{1}{1-2\tau} = O(1)$, then for any $\varepsilon > 0$, after $O\left(K + \frac{\log((1-2\tau)\varepsilon)}{\log(2\delta_{s^*+s})}\right)$ iterations of IHT, $\mathbf{W}^k$ satisfies*

$$R(\mathbf{W}^k) \leq C^2 \min_{\mathbf{W}, \|\mathbf{W}\|_0 \leq s} R(\mathbf{W}) + \varepsilon + o(1)$$

with probability $1 - o(1)$.

Theorem 3.8 (Risk upper bound for mAIHT)**.** *Let $\mathbf{X} = \frac{1}{\sqrt{n}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $\mathbb{P}(\|\boldsymbol{x}_1\|_2 \leq M_x) = 1$. For $\{\mathbf{W}^n\}$ generated by mAIHT with step size $\alpha < \frac{1}{L}$ and λ. Assume $\alpha_1 = \alpha_2 = O(1)$, $\lambda = O(1)$ and δ satisfies $\log(1/\delta) = O(n)$. For integer s, assume with probability $1 - \zeta$:*

1. *all of $\{\mathbf{W}^n\}$'s accumulation points have support size of s^* and $s^* \geq s$;*
2. *the restricted isometry constant of the matrix $\mathbf{I}_{d_2} \otimes \sqrt{\alpha}\mathbf{X}$ satisfies $\delta_{s^*+s} \leq \tau < u \approx 0.3478$, where u is real root of equation $4u^3 + 4u^2 + u = 1$;*

Then with probability at least $1 - \delta - \zeta$, there exists a K. For any $k > K$, it holds that

$$R(\mathbf{W}^k) \leq C^2 \min_{\mathbf{W}, \|\mathbf{W}\|_0 \leq s} R(\mathbf{W}) + O\left(\frac{1}{1 - \rho_\tau}\right)\rho^{k-K} + O\left(\sqrt{\frac{\log(2d_1^2/\delta)}{n(1 - \rho_\tau)^4}}\right),$$

where $\rho = \frac{2\delta_{s^*+s}\sqrt{1+\delta_{s^*+s}}}{\sqrt{1-\delta_{s^*+s}}} < 1$, $\rho_\tau = \frac{2\tau\sqrt{1+\tau}}{\sqrt{1-\tau}} < 1$ and $C = 1 + \frac{\sqrt{1+\delta_{s^*+s}}(4+2\delta_{s^*+s})}{\sqrt{1-\delta_{s^*+s}}(1-\rho)}$.

Corollary 3.9. *In Theorem 3.8, suppose* $\log(1/\delta) = o(n)$, $\delta = o(1)$, $\zeta = o(1)$ *and* $\frac{1}{1-\rho_\tau} = O(1)$, *then for any* $\varepsilon > 0$, *after* $O\left(K + \frac{\log((1-\rho_\tau)\varepsilon)}{\log(\rho)}\right)$ *iterations of mAIHT,* $\mathbf{W}^k$ *satisfies*

$$R(\mathbf{W}^k) \leq C^2 \min_{\mathbf{W}, \|\mathbf{W}\|_0 \leq s} R(\mathbf{W}) + \varepsilon + o(1)$$

with probability $1 - o(1)$.

Under proper assumptions, the upper bounds of the excess risks $R(\mathbf{W}^k) - \min_{\mathbf{W}, \|\mathbf{W}\|_0 \leq s} R(\mathbf{W})$ for IHT and mAIHT are composed of three terms. The first term $(C^2 - 1)\min_{\mathbf{W}, \|\mathbf{W}\|_0 \leq s} R(\mathbf{W})$ represents the unavoided optimization error caused by the non-convexity of the problem. Luckily, this error is controlled by the optimal risk, i.e. if the model has an intrinsic sparse structure, this error would be small. In practice, if the underlying problem is sparse, the algorithm is more likely to find a solution close to the optimal one, and this error becomes negligible. The second term is a non-sufficient optimization error caused by the finite time iterations of the algorithm. The bounds for this error show that the convergence rate is asymptotically linear under stronger assumptions than those discussed in section 3.1. The last term is the standard generalization error, which reflects how well the learned model can be generalized to the unseen data.

These two theorems suggest that mAIHT shares a similar risk upper bound with IHT. Both of them can be controlled by a constant multiple of the optimal risk with high probability. The detailed proofs can be found in Appendix C.

4. Numerical Experiments

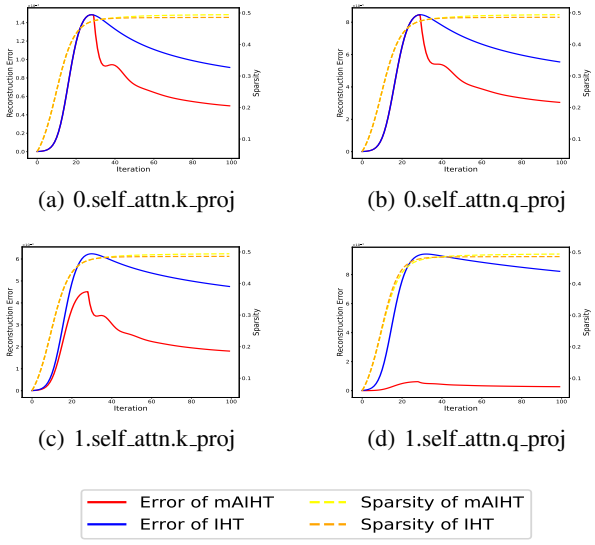
4.1. Experimental setup

Pre-processing. Like Wanda (Sun et al., 2023), we randomly utilize 128 calibration samples drawn from the C4 training dataset (Raffel et al., 2020). For each layer ℓ, the calibration data $\mathbf{X}$ for ℓ is the output of the previous $\ell - 1$ pruned layers. For better scaling, we normalize $\mathbf{X}$ before pruning (Meng et al., 2024a). Specifically, we let $\mathbf{E} = \mathrm{Diag}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-\frac{1}{2}}$, $\widehat{\mathbf{W}} \leftarrow \mathbf{E}^{-1}\widehat{\mathbf{W}}$ and $\mathbf{X} \leftarrow \mathbf{X}\mathbf{E}$. This technique can improve the pruning performance in practice. Notice that all the computations only need $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, so we only need to compute it once per layer and store it.

Hyperparameters choice. We set the step size $\alpha = \alpha_1 = \alpha_2 = 0.95/\|\mathbf{X}^{\mathrm{T}}\mathbf{X}\|_2$, the ℓ_2 penalty coefficient $\mu = 0.1$ (as defined in (5)), the number of mAIHT iterations $t_1 = 50$, and the number of weight refining iterations $t_2 = 30$.

4.2. Efficiency of the acceleration

We first use a single-layer experiment to demonstrate the efficiency of the acceleration, using reconstruction error as a metric. Specifically, we measure the reconstruction error as $\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}^k\|_F^2 / \|\mathbf{X}\widehat{\mathbf{W}}\|_F^2$, where $\widehat{\mathbf{W}}$ is the estimated weight matrix and $\mathbf{W}^k$ is the true weight matrix at the k-th iteration. We also report the estimated sparsity level in each iteration. We apply both IHT and mAIHT to several layers of the LLaMA-7B model. The results are shown in Figure 1. It can be observed that the iterative process of our algorithm is divided into two stages. In the first stage, the support gradually shrinks to the desired sparsity level, with the sparsity schedule naturally adopting an exponential shape. As shown in (Benbaki et al., 2023), such an exponential-shaped shrinking schedule is relatively advantageous. In the second stage, proximal gradient descent is used to optimize the reconstruction error. Comparing IHT and mAIHT, we can see that mAIHT achieves faster convergence and significantly improves the quality of the final stable solution.



(a) 0.self_attn.k_proj (b) 0.self_attn.q_proj

(c) 1.self_attn.k_proj (d) 1.self_attn.q_proj

Figure 1. Reconstruction Process for IHT and mAIHT in One Layer.

4.3. Pruning LLaMA-7B model

In this section, we compare different methods by pruning the LLaMA-7B model at various levels of sparsity.

First, we train the LLaMA-7B model on the WikiText2 dataset (Merity et al., 2016) and perform weight pruning using different methods. We compare our method with one-shot pruning methods for LLMs, including (i) Magnitude Pruning (MP) (Han et al., 2015), (ii)Wanda (Sun et al., 2023), (iii) SparseGPT (Frantar & Alistarh, 2023), and (iv)ADMM-based Pruning (Boža, 2024; Meng et al., 2024a). We use the original settings of these algorithms in the codebase released by Sun et al. (2023) and Boža (2024). For each method, we measure the perplexity, the exponential of the loss function. Table 1 showcases that mAIHT Pruner

outperforms all other methods in 0.2 to 0.5 sparsity. When the sparsity is 0.1, mAIHT is slightly inferior to ADMM but superior to other methods.

Table 1. Perplexity at different sparsity levels in WikiText2

	Sparsity level				
Method	0.1	0.2	0.3	0.4	0.5
MP	5.8061	6.0206	6.6685	8.6012	17.2857
Wanda	5.6962	5.8229	5.9951	6.3970	7.2588
Sparsegpt	5.6972	5.8084	5.9730	6.3415	7.2397
ADMM	**5.6925**	5.8045	5.9586	6.3268	7.0826
mAIHT	5.6928	**5.8042**	**5.9565**	**6.3240**	**7.0720**

Then we compare our proposal with other existing methods in the zero-shot tasks. We compare the methods on seven zero-shot tasks which are the same as those selected by (Sun et al., 2023). The tasks include BoolQ (Clark et al., 2019), RTE (Wang, 2018), HellaSWAG (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC easy and challenge (Clark et al., 2018), and OpenbookQA (Mihaylov et al., 2018) . Table 2 presents the accuracy of pruned models across various datasets, showcasing the performance of different pruning methods at different sparsity levels.

Our results reveal that within the low to moderate sparsity range (from 0.2 to 0.5), our method demonstrates accuracy among the top two on the majority of test datasets, and achieves the best mean accuracy across all sparsity levels.

5. Conclusion

In this paper, we introduce the mAIHT Pruner, an efficient layerwise one-shot pruning method for LLMs based on ℓ_0-penalized optimization. In our theoretical analysis, we prove the convergence of the algorithm and provide its convergence rate and risk upper bounds. The mAIHT Pruner is capable of finding high-quality solutions in practical pruning tasks. Through extensive experiments, we verify its superiore pruning performance at low to moderate sparsity levels.

Future works will consider extending this method to other various kinds of pruning strategies, such as structured pruning (Meng et al., 2024b) and nonuniform layerwise pruning (Yin et al., 2023), since they can all be formulated or approximately formulated as a variant of problem (1).

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. Wen's research is partially supported by National Key R&D Program of China (2024YFA1012200), the National Natural Science Foundation of China (12171449), and USTC Research Funds of the Double First-Class Initiative (YD2040002019). Pan's research was partially supported by the National Natural Science Foundation of China (12322113, 72495122, 12288201).

Impact Statement

The proliferation of Large Language Models (LLMs) has ushered in a new era of AI capabilities, but their immense parameter sizes pose challenges that hinder accessibility and sustainability. By addressing these issues, our mAIHT method offers a significant step forward in making LLMs more efficient and practical. With its solid theoretical foundations and demonstrated empirical success, mAIHT not only enables effective pruning with rigorous guarantees but also holds the potential to democratize the deployment of LLMs across industries.

However, this advancement also demands a focus on responsible AI practices. Reducing model size while maintaining performance raises essential questions about preserving fairness, robustness, and explainability in compressed models. As these technologies become more accessible, it becomes increasingly important to ensure they are leveraged ethically and inclusively.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Attouch, H. and Bolte, J. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116:5–16, 2009.

Beck, A. and Teboulle, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11):2419–2434, 2009.

Benbaki, R., Chen, W., Meng, X., Hazimeh, H., Ponomareva, N., Zhao, Z., and Mazumder, R. Fast as chita: Neural network pruning with combinatorial optimization. In *International Conference on Machine Learning*, pp. 2031–2049. PMLR, 2023.

Bertsimas, D., King, A., and Mazumder, R. Best subset selection via a modern optimization lens. 2016.

Blumensath, T. and Davies, M. E. Iterative Thresholding for Sparse Approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, December 2008. ISSN 1069-5869, 1531-5851. doi: 10.1007/s00041-008-9035-z.

Table 2. Zero-shot accuracies on various tasks for different sparsity levels. In each dataset, we bold the top two results, and for the average accuracy, we bold the best result.

Sparsity	Method	BoolQ	RTE	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Mean
0.5	MP	54.40	54.15	60.91	59.35	54.34	37.12	35.00	50.75
	Wanda	72.78	**54.51**	70.09	66.14	**65.03**	**40.27**	39.60	58.35
	SparseGPT	73.76	52.35	69.27	**70.09**	**65.53**	39.76	40.00	58.68
	ADMM	**74.34**	**54.51**	**70.49**	**69.22**	64.90	39.08	**40.80**	59.05
	mAIHT	**74.86**	**58.12**	**70.38**	68.90	64.69	**39.76**	**41.80**	**59.79**
0.4	MP	67.46	57.76	70.27	66.38	64.18	39.68	39.20	57.85
	Wanda	74.07	**61.37**	**73.77**	67.80	**70.08**	42.66	**43.00**	61.82
	SparseGPT	74.59	55.96	72.98	69.22	69.07	41.72	41.80	60.76
	ADMM	**75.57**	58.84	**73.77**	**69.53**	**69.91**	**43.34**	**42.40**	61.91
	mAIHT	**75.32**	**62.82**	73.70	**69.46**	69.61	**44.03**	42.20	**62.45**
0.3	MP	72.32	61.73	74.05	68.90	70.62	44.62	40.60	61.83
	Wanda	**76.30**	63.18	**75.63**	**69.93**	70.92	43.52	**43.20**	63.24
	SparseGPT	**76.00**	**63.90**	75.18	69.46	71.00	**45.22**	43.00	63.39
	ADMM	75.38	61.37	**75.48**	69.22	**71.17**	44.45	**43.60**	62.95
	mAIHT	75.78	**64.62**	75.42	**69.77**	**71.76**	**44.71**	**43.20**	**63.61**
0.2	MP	74.83	60.65	75.57	**70.32**	**72.10**	44.88	43.00	63.05
	Wanda	**76.02**	65.34	**76.14**	**69.77**	**72.10**	**45.31**	44.20	64.13
	SparseGPT	75.47	63.18	76.08	69.61	71.59	44.71	44.20	63.55
	ADMM	75.63	**65.70**	**76.23**	**69.77**	71.59	44.71	**45.00**	64.09
	mAIHT	**75.72**	**66.79**	76.13	69.46	71.89	**44.97**	**45.20**	**64.31**

Blumensath, T. and Davies, M. E. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

Boža, V. Fast and Effective Weight Update for Pruned Large Language Models, July 2024.

Candes, E. and Tao, T. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005a. doi: 10.1109/TIT.2005.858979.

Candes, E. J. The restricted isometry property and its implications for compressed sensing. *Comptes rendus. Mathematique*, 346(9-10):589–592, 2008.

Candes, E. J. and Tao, T. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005b.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Diao, H., Jayaram, R., Song, Z., Sun, W., and Woodruff, D. Optimal sketching for kronecker product regression and low rank approximation. *Advances in neural information processing systems*, 32, 2019.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Frantar, E. and Alistarh, D. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

Frantar, E. and Alistarh, D. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Gray, S., Radford, A., and Kingma, D. P. Gpu kernels for block-sparse weights. *arXiv preprint arXiv:1711.09224*, 3(2):2, 2017.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Hassibi, B. and Stork, D. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models, October 2021.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Li, H. and Lin, Z. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28, 2015.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Meng, X., Behdin, K., Wang, H., and Mazumder, R. ALPS: Improved Optimization for Highly Sparse One-Shot Pruning for Large Language Models, June 2024a.

Meng, X., Ibrahim, S., Behdin, K., Hazimeh, H., Ponomareva, N., and Mazumder, R. Osscar: One-shot structured pruning in vision and language models with combinatorial optimization. *arXiv preprint arXiv:2403.12983*, 2024b.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Parikh, N., Boyd, S., et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, A. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Wu, Y. N., Tsai, P.-A., Muralidharan, S., Parashar, A., Sze, V., and Emer, J. Highlight: Efficient and flexible dnn acceleration with hierarchical structured sparsity. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 1106–1120, 2023.

Yin, L., Wu, Y., Zhang, Z., Hsieh, C.-Y., Wang, Y., Jia, Y., Li, G., Jaiswal, A., Pechenizkiy, M., Liang, Y., et al. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. *arXiv preprint arXiv:2310.05175*, 2023.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zhao, P., Hu, H., Li, P., Zheng, Y., Wang, Z., and Yuan, X. A convex-optimization-based layer-wise post-training pruner for large language models. *arXiv preprint arXiv:2408.03728*, 2024.

A. Properties of the Optimization Function

In this section, we analyze the theoretical properties of the optimization function (2). Recall that we define $f(\mathbf{W}) = \frac{1}{2}\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}\|_F^2$, $h(\mathbf{W}) = \lambda\|\mathbf{W}\|_0$ and $\mathcal{L}(\mathbf{W}) = f(\mathbf{W}) + h(\mathbf{W})$, where $\widehat{\mathbf{W}} \in \mathbb{R}^{d_1 \times d_2}$, $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, and $\mathbf{X} \in \mathbb{R}^{N \times d_1}$.

A.1. Derivations of $f(\mathbf{W})$

We first introduce some definitions of tensor trick (Diao et al., 2019), an instrument to compute gradients in a clean and tractable fashion:

Definition A.1 (Vectorization)**.** For any matrix $\mathbf{X} \in R^{m \times n}$, we define $\underline{\mathbf{X}} := \mathrm{vec}(\mathbf{X}) \in R^{mn}$, such that $\mathbf{X}_{ij} = \underline{\mathbf{X}}_{(i-1)n+j}$ for all $i \in [m]$ and $j \in [n]$.

Lemma A.2 (Tensor Trick(Diao et al., 2019))**.** *For any* $\mathbf{A} \in \mathbb{R}^{m_a \times n_a}, \mathbf{B} \in \mathbb{R}^{m_b \times n_b}$ *and* $\mathbf{X} \in \mathbb{R}^{n_a \times n_b}$*, it holds* $\mathrm{vec}(\mathbf{A}\mathbf{X}\mathbf{B}^{\mathrm{T}}) = (\mathbf{A} \otimes \mathbf{B})\underline{\mathbf{X}} \in \mathbb{R}^{m_a m_b}$.

Hence, we can compute the gradient f as

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{W}} = \mathbf{X}^{\mathrm{T}}\mathbf{X}(\mathbf{W} - \widehat{\mathbf{W}}), \tag{6}$$

and the Hessian matrix of f is given by

$$\frac{\mathrm{d}^2 f}{\mathrm{d}\underline{\mathbf{W}}^2} = \frac{\mathrm{d}\,\mathrm{vec}(\mathbf{X}^{\mathrm{T}}\mathbf{X}(\mathbf{W} - \widehat{\mathbf{W}}))}{\mathrm{d}\underline{\mathbf{W}}} = \frac{\mathrm{d}((\mathbf{I}_{d_2} \otimes \mathbf{X}^{\mathrm{T}}\mathbf{X})(\underline{\mathbf{W}} - \widehat{\underline{\mathbf{W}}}))}{\mathrm{d}\underline{\mathbf{W}}} = \mathbf{I}_{d_2} \otimes \mathbf{X}^{\mathrm{T}}\mathbf{X}. \tag{7}$$

The second equation relies on the tensor trick (Lemma A.2). The Hessian matrix of f implies that the gradient of f is Lipschitz continuous with a Lipschitz constant of $L = \|\mathbf{X}^{\mathrm{T}}\mathbf{X}\|_2 = \|\mathbf{X}\|_2^2$, i.e. $\|\nabla f(\mathbf{W}) - \nabla f(\mathbf{V})\|_F \leq L\|\mathbf{W} - \mathbf{V}\|_F$.

A.2. Proof of Proposition 2.1

The proximal operator of $\lambda\|.\|_0$ is given by

$$\mathrm{Prox}_{\lambda\|.\|_0}(\mathbf{W}) = \arg\min_{\mathbf{V} \in \mathbb{R}^{d_1 \times d_2}} \left\{ \lambda\|\mathbf{V}\|_0 + \frac{1}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 \right\}$$

By taking condition on support size s, we can decompose the optimization as

$$\begin{aligned}
\min_{\mathbf{V} \in \mathbb{R}^{d_1 \times d_2}} \left\{ \lambda\|\mathbf{V}\|_0 + \frac{1}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 \right\} &= \min_{0 \leq s \leq d_1 d_2} \min_{\mathbf{V}, \|\mathbf{V}\|_0 = s} \left\{ \lambda\|\mathbf{V}\|_0 + \frac{1}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 \right\} \\
&= \min_{0 \leq s \leq d_1 d_2} \min_{\mathbf{V}, \|\mathbf{V}\|_0 = s} \left\{ \lambda s + \frac{1}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 \right\} \\
&= \min_{0 \leq s \leq d_1 d_2} \left\{ \lambda s + \frac{1}{2}\|\mathbf{W} - \mathrm{P}_s(\mathbf{W})\|_F^2 \right\},
\end{aligned}$$

where $\mathrm{P}_s(\mathbf{W})$ denotes the operation of retaining the largest s elements of the absolute values from $\mathbf{W}$ and setting the others to 0.

Consider arranging all entries of $\mathbf{W}$ in ascending order of their absolute values as $w_1, w_2, \cdots, w_{d_1 d_2}$. For a given support size s, denote

$$\mathcal{A}(s) = \lambda s + \frac{1}{2}\|\mathbf{W} - \mathrm{P}_s(\mathbf{W})\|_F^2 = \lambda s + \frac{1}{2}\sum_{i=1}^{d_1 d_2 - s} |w_i|^2$$

then

$$\mathcal{A}(s) - \mathcal{A}(s-1) = \lambda - \frac{1}{2}|w_{d_1 d_2 - s + 1}|^2$$

Hence, we have for $|w_{d_1 d_2 - s + 1}| \leq \sqrt{2\lambda}$, $\mathcal{A}(s) \geq \mathcal{A}(s-1)$, while for $|w_{d_1 d_2 - s + 1}| > \sqrt{2\lambda}$, $\mathcal{A}(s) < \mathcal{A}(s-1)$. Denote $s^* = |\,\mathrm{Supp}(\mathrm{H}_{\sqrt{2\lambda}}(\mathbf{W}))|$, then we have $|w_{d_1 d_2 - s^* + 1}| > \sqrt{2\lambda}$, and $|w_{d_1 d_2 - s^*}| \leq \sqrt{2\lambda}$. This suggests that

$$\mathcal{A}(s^*) < \mathcal{A}(s^* - 1) < \cdots < \mathcal{A}(0)$$

and

$$\mathcal{A}(s^*) \leq \mathcal{A}(s^*+1) \leq \cdots \leq \mathcal{A}(d_1 d_2).$$

Since $\mathrm{H}_{\sqrt{2\lambda}}(\mathbf{W}) = \mathrm{P}_{s^*}(\mathbf{W})$, then we have

$$\mathrm{H}_{\sqrt{2\lambda}}(\mathbf{W}) \in \mathrm{Prox}_{\lambda\|\cdot\|_0}(\mathbf{W}).$$

A.3. Kurdyka-Lojasiewicz (KL) properties

In the subsequent proof of our main results, we will leverage the Kurdyka-Lojasiewicz (KL) property of the optimization functions f and h. For details on the KL property, please refer to Bolte et al. (2014). Here, we will prove the KL property by showing that the optimization functions are semi-algebraic (Attouch & Bolte, 2009). We begin by defining the semi-algebraic function.

Definition A.3. (Bolte et al., 2014) A subset S of $\mathbb{R}^n$ is called the semialgebraic set if there exists a finite number of real polynomial functions g_{ij}, h_{ij} such that

$$S = \bigcup_j \bigcap_i \{\boldsymbol{u} \in \mathbb{R}^n : g_{ij}(\boldsymbol{u}) = 0, h_{ij}(\boldsymbol{u}) < 0\}.$$

A function $f(\boldsymbol{u})$ is called the semi-algebraic function if its graph $\{(\boldsymbol{u}, t) \in \mathbb{R}^n \times \mathbb{R}, t = f(\boldsymbol{u})\}$ is a semi-algebraic set.

Next, we will prove that f and h are semi-algebraic functions and provide some function properties that are needed in the subsequent proofs.

Proposition A.4.

1. *f and h are semi-algebraic function.*
2. *h is proper and lower semi-continuous.*
3. *$\mathcal{L}$ is coercive i.e., $\mathcal{L}$ is bounded from below and $\mathcal{L}(W) \to \infty$ when $\|\mathbf{W}\|_F^2 \to \infty$.*

Proof. The proofs of 2 and 3 are trivial, thus we only need to check 1:

For $f(\mathbf{W}) = \frac{1}{2}\|\mathbf{X}\widehat{\mathbf{W}} - \mathbf{X}\mathbf{W}\|_F^2$ is a real polynomial function, it is a semi-algebraic function.

For $h(\mathbf{W}) = \lambda\|\mathbf{W}\|_0$, the graph of h is:

$$G = \bigcup_{i=0}^{d_1 d_2} \{(\mathbf{W}, \lambda i) : \|\mathbf{W}\|_0 = i\} = \bigcup_{i=0}^{d_1 d_2} \bigcup_{S \in 2^{[d_1 d_2]}, \|S\|_0 = i} \{(\mathbf{W}, \lambda i) : \mathrm{Supp}(\mathbf{W}) = S\}.$$

Since $\{(\mathbf{W}, \lambda i) : \mathrm{Supp}(\mathbf{W}) = S\}$ is a semi-algebraic set, G is a semi-algebraic set and h is a semi-algebraic function. □

B. Proof of Optimization Results

B.1. Technical lemmas and some properties

We first define L-smooth functions.

Definition B.1. A function $f : \mathbb{R}^n \to \mathbb{R}$ is called the L-smooth function if f is differentiable and it satisfies

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

for every $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

Then we give some important lemmas for L-smooth functions and the proximal operator.

Lemma B.2 (Quadratic upper bound)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a L-smooth function. Then,*

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \nabla f(\boldsymbol{y}) \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \text{ for every } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

Lemma B.3 (Convergence rate for convex proximal gradient method)**.** *Let f be convex and L-smooth. Let g be convex. Assume then proximal gradient method applied to minimize $h = f + g$ with step size $\alpha \leq \frac{1}{L}$ generates $\{\boldsymbol{x}^t\}$. Then there exists a $\boldsymbol{x}^*$ such that $\boldsymbol{x}^t \to \boldsymbol{x}^*$ and*

$$h(\boldsymbol{x}^t) - h(\boldsymbol{x}^*) \leq \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2^2}{2\alpha(t-1)}.$$

Lemma B.4 (Sufficient descent lemma)**.** *Denote $f(\mathbf{W}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2$ and $h(\mathbf{W}) = \lambda\|\mathbf{W}\|_0$. Let $\alpha < \frac{1}{L}$. Then for any $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, and $\mathbf{V}$ defined by*

$$\mathbf{V} \in \mathrm{Prox}_{\alpha h(.)}(\mathbf{W} - \alpha\nabla f(\mathbf{W})).$$

We have

$$f(\mathbf{V}) + h(\mathbf{V}) \leq f(\mathbf{W}) + h(\mathbf{W}) - \frac{1}{2}\left(\frac{1}{\alpha} - L\right)\|\mathbf{V} - \mathbf{W}\|_F^2.$$

Proof. By the definition of the proximal operator, we can get

$$\mathbf{V} = \arg\min_{\mathbf{U} \in \mathbb{R}^{d_1 \times d_2}} \frac{1}{2}\|\mathbf{U} - \mathbf{W} + \alpha\nabla f(\mathbf{W})\|_F^2 + \alpha h(\mathbf{U}).$$

Taking $\mathbf{U} = \mathbf{W}$, we can get:

$$\frac{1}{2}\|\mathbf{V} - \mathbf{W} + \alpha\nabla f(\mathbf{W})\|_F^2 + \alpha h(\mathbf{V}) \leq \frac{1}{2}\|\alpha\nabla f(\mathbf{W})\|_F^2 + \alpha h(\mathbf{W}),$$
$$\frac{1}{2\alpha}\|\mathbf{V} - \mathbf{W}\|_F^2 + \langle \mathbf{V} - \mathbf{W}, \nabla f(\mathbf{W}) \rangle + h(\mathbf{V}) \leq h(\mathbf{W}).$$

By using the quadratic upper bound inequality (Lemma B.2) and the above inequality, we can obtain

$$\begin{aligned} f(\mathbf{V}) + h(\mathbf{V}) &\leq f(\mathbf{W}) + \langle \nabla f(\mathbf{W}), \mathbf{V} - \mathbf{W} \rangle + \frac{L}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 + h(\mathbf{V}) \\ &\leq f(\mathbf{W}) + \frac{L}{2}\|\mathbf{W} - \mathbf{V}\|_F^2 + h(\mathbf{W}) - \frac{1}{2\alpha}\|\mathbf{W} - \mathbf{V}\|_F^2 \\ &= f(\mathbf{W}) + h(\mathbf{W}) - \frac{1}{2}\left(\frac{1}{\alpha} - L\right)\|\mathbf{V} - \mathbf{W}\|_F^2. \end{aligned}$$

□

Based on this lemma, we give some direct properties of IHT and mAIHT.

Proposition B.5. *Let $\alpha^k = \alpha < \frac{1}{L}$ and $\{\mathbf{W}^k\}_{k=0}^{\infty}$ be the sequence generated in (3). Then we have*

(1) The sequence $\mathcal{L}(\mathbf{W}^k)$ satisfies

$$\mathcal{L}(\mathbf{W}^{k-1}) - \mathcal{L}(\mathbf{W}^k) \geq \frac{1}{2}(\frac{1}{\alpha} - L)\|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2,$$

and is decreasing and converges to a positive number $\mathcal{L}^*$.

(2) $\|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F \to 0$ as $k \to \infty$.

(3) For any positive integer M, we have:

$$\min_{k=1,\ldots,M} \quad \|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2 \leq \frac{2(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}^*)}{M(\frac{1}{\alpha} - L)}.$$

(4) There exists a positive integer K, such that for all integer $k > K$, $\mathbf{W}^k$ has the same support.

Proof. (1) By using Lemma B.4 and the iteration (3), for any integer $k > 0$, we have

$$\frac{1}{2}(\frac{1}{\alpha} - L)\|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2 \leq \mathcal{L}(\mathbf{W}^{k-1}) - \mathcal{L}(\mathbf{W}^k).$$

This indicates that sequence $\mathcal{L}(\mathbf{W}^k)$ is a positive decreasing sequence, so there exists a positive real number $\mathcal{L}^*$, such that $\lim_{k\to\infty} \mathcal{L}(\mathbf{W}^k) = \mathcal{L}^*$.

(2) For any $M \geq 0$, we have

$$\sum_{k=1}^{M} \|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2 \leq \frac{2(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}(\mathbf{W}^M))}{\frac{1}{\alpha} - L} \leq \frac{2\mathcal{L}(\mathbf{W}^0)}{\frac{1}{\alpha} - L}.$$

Let $M \to \infty$, this indicates that

$$\|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F \to 0 \text{ as } k \to \infty.$$

(3) For any positive integer M, we have

$$\min_{k=1,\ldots,M} \quad \|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2 \leq \frac{2(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}(\mathbf{W}^M))}{M(\frac{1}{\alpha} - L)} \leq \frac{2(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}^*)}{M(\frac{1}{\alpha} - L)}.$$

(4) We will prove (4) by contradiction. Suppose for any integer $K > 0$, there exists a $k > K$, such that $\mathbf{W}^k$ and $\mathbf{W}^{k-1}$ have different supports.

By the definition of $\mathbf{W}^k$, the absolute value of any of its non-zero elements is greater than $\sqrt{2\alpha\lambda}$, which implies that

$$\|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2 \geq 2\alpha\lambda, \tag{8}$$

if $\mathbf{W}^k$ and $\mathbf{W}^{k-1}$ have different supports.

From the assumption, we know that there are infinitely many k such that (8) holds, which implies that

$$\infty = \sum_{k=1}^{\infty} \|\mathbf{W}^k - \mathbf{W}^{k-1}\|_F^2 \leq \frac{2(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}^*)}{\frac{1}{\alpha} - L}.$$

This leads to a contradiction. So there exists a positive integer K, such that for all integers $k > K$, $\mathbf{W}^k$ has the same support. □

Proposition B.6. *Let $\alpha_1, \alpha_2 < \frac{1}{L}$ and $\{\mathbf{W}^k\}_{k=0}^{\infty}$ be the sequence generated in mAIHT(4). Then we have*

(1) The sequence $\mathcal{L}(\mathbf{W}^k)$ satisfies

$$\mathcal{L}(\mathbf{W}^{k+1}) \leq \mathcal{L}(\mathbf{V}^{k+1}) \leq \mathcal{L}(\mathbf{W}^k) - \frac{1}{2}(\frac{1}{\alpha_2} - L)\|V^{k+1} - \mathbf{W}^k\|_F^2.$$

and is decreasing and converges to a positive number $\mathcal{L}^*$.

(2) $\|\mathbf{V}^k - \mathbf{W}^{k-1}\|_F \to 0$ as $k \to \infty$.

(3) For any positive integer M, we have:

$$\min_{k=1,\ldots,M} \quad \|\mathbf{V}^k - \mathbf{W}^{k-1}\|_F^2 \leq \frac{2(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}^*)}{M(\frac{1}{\alpha_2} - L)}.$$

This proposition's proof is essentially the same as Proposition B.5, so we omit its proof.

B.2. Proof of Theorem 3.3

Proof. (1) After a finite number of steps, the support of $\mathbf{W}^k$ remains unchanged(see Theorem B.5(4)). Consequently, the algorithm's iteration is equivalent to performing projected gradient descent over the support set. Since projected gradient descent over a closed convex set converges when $\alpha^k < \frac{1}{L}$, there exists a $\mathbf{W}^*$ such that $\mathbf{W}^k \to \mathbf{W}^*$. Moreover, Since the absolute value of all nonzero entries of $\mathbf{W}^k$ is greater than $\sqrt{2\alpha\lambda}$, $\mathbf{W}^*$ retains the same support as $\mathbf{W}^k$ for sufficiently large k.

Hence, $\|\mathbf{W}^k\|_0 \to \|\mathbf{W}^*\|_0$ as $k \to \infty$. This implies that $\mathcal{L}(\mathbf{W}^k) \to \mathcal{L}(\mathbf{W}^*)$.

The remaining proof only needs to verify that $\mathbf{W}^*$ satisfies the equivalent condition for α-fixed points.

For $(i,j) \in \mathrm{Supp}(\mathbf{W}^*)$ and sufficiently large k, $\mathbf{W}_{ij}^k = \mathbf{W}_{ij}^{k-1} + (\alpha \mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^{k-1}))_{ij}$ and $|\mathbf{W}_{ij}^k| \geq \sqrt{2\alpha\lambda}$. Let $k \to \infty$, we can obtain

$$(\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^*))_{ij} = 0 \text{ and } |\mathbf{W}_{ij}^*| \geq \sqrt{2\alpha\lambda}.$$

For $(i,j) \notin \mathrm{Supp}(\mathbf{W}^*)$ and sufficiently large k, $|(\alpha \mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^{k-1}))_{ij}| < \sqrt{2\alpha\lambda}$. Let $k \to \infty$, we can get

$$|(\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^*))_{ij}| \leq \sqrt{\frac{2\lambda}{\alpha}}.$$

(2) Since $\mathcal{L}(\mathbf{W}^k) \leq \mathcal{L}(\mathbf{V}^k) \leq \mathcal{L}(\mathbf{W}^0)$ for $k = 1, 2, \cdots$ and $\mathcal{L}$ is coercive, $\{\mathbf{W}^k\}_{k=0}^{\infty}$ and $\{\mathbf{V}^k\}_{k=0}^{\infty}$ generated by (4) are bounded; otherwise, $\mathcal{L}(\mathbf{W}^k)$ or $\mathcal{L}(\mathbf{V}^k)$ will tend to positive infinity, which contradicts the boundedness of $\mathcal{L}(\mathbf{W}^k)$ or $\mathcal{L}(\mathbf{V}^k)$.

By proposition B.6 , we know that $\mathbf{V}^{k+1} - \mathbf{W}^k \to 0$ as $k \to \infty$. Suppose $\{\mathbf{W}^k\}_{k=0}^{\infty}$ has a accumulation point $\mathbf{W}^*$ and its sub-sequence $\{\mathbf{W}^{k_j}\}_{j=0}^{\infty}$ satisfy $\lim_{j\to\infty} \mathbf{W}^{k_j} = \mathbf{W}^*$. Then we have $\lim_{j\to\infty} \mathbf{V}^{k_j+1} = \mathbf{W}^*$. Since the absolute value of any non-zero elements of $\mathbf{V}^k, \mathbf{W}^k, k = 2, 3, \cdots$ is greater than $\min\{\sqrt{2\alpha_1\lambda}, \sqrt{2\alpha_2\lambda}\}$, there exists a $J > 0$, such that for every $j > J$, $\mathbf{W}^{k_j}$ and $\mathbf{V}^{k_j+1}$ have the same support as $\mathbf{W}^*$. Hence, similar to (1) above, the remaining proof only needs to verify that $\mathbf{W}^*$ satisfies the equivalent condition of a α_2-fixed point.

For $(i,j) \in \mathrm{Supp}(\mathbf{W}^*)$ and sufficiently large k, $\mathbf{V}_{ij}^{k_n+1} = \mathbf{W}_{ij}^{k_n} + (\alpha_2 \mathbf{X}^{\mathrm{T}}(Y - \mathbf{X}\mathbf{W}^{k_n}))_{ij}$ and $|\mathbf{V}_{ij}^{k_n}| \geq \sqrt{2\alpha_2\lambda}$. Let $n \to \infty$, We can obtain

$$(\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^*))_{ij} = 0 \text{ and } |\mathbf{W}_{ij}^*| \geq \sqrt{2\alpha_2\lambda}.$$

For $(i,j) \notin \mathrm{Supp}(\mathbf{W}^*)$ and sufficiently large n, $|(\alpha_2 \mathbf{X}^{\mathrm{T}}(Y - \mathbf{X}\mathbf{W}^{k_n}))_{ij}| < \sqrt{2\alpha_2\lambda}$. Let $k \to \infty$, we can get

$$|(\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^*))_{ij}| \leq \sqrt{\frac{2\lambda}{\alpha_2}}.$$

This completes the proof.

□

B.3. Proof of Theorem 3.4

(1) From proposition B.5, the support of $\mathbf{W}^k$ will stabilize after finite iterations. Suppose after k_1's iteration the support stabilizes, denoted as S. Then the iteration is equivalent to

$$\mathbf{W}^k = \mathrm{P}_S(\mathbf{W}^{k-1} - \alpha^k \nabla f(\mathbf{W}^{k-1})),$$

where $\mathrm{P}_S(\mathbf{W})$ project $\mathbf{W}$ to S. By using Lemma B.3 with $f = f(\mathbf{W})$ and

$$g = I_S(\mathbf{W}) = \begin{cases} 0, & \mathrm{Supp}(\mathbf{W}) \subset S \\ \infty, & \mathrm{Supp}(\mathbf{W}) \not\subset S. \end{cases}$$

We have

$$\mathcal{L}(\mathbf{W}^k) - \mathcal{L}^* \leq \frac{\|\mathbf{W}^{k_1} - \mathbf{W}^*\|_F^2}{2\alpha(k-k_1)}.$$

(2) We first illustrate that our optimization function satisfies the KL property through the following theorem:

Theorem B.7 ((Bolte et al., 2014; Attouch & Bolte, 2009)). *For a semi-algebraic function $f(\boldsymbol{x})$, if it is a proper and lower semicontinuous, then f satisfies the KL property and the desingularising function has the form of $\varphi(s) = cs^{1-\theta}$ where c is positive real number and $\theta \in [0, 1)$.*

The proof of Theorem 3.4 is a direct application of the following theorem established by Li & Lin (2015).

Theorem B.8 ((Li & Lin, 2015)). *Let f be a proper function with Lipschitz continuous gradients and h be proper and lower semicontinuous. Assume that $\mathcal{L}$ is coercive, f and h satisfy the KL property and the desingularising function has the form of $\varphi(t) = \frac{C}{\theta}t^\theta$ for some $C > 0, \theta \in (0, 1]$, then the monotone accelerated proximal gradient methods satisfies:*

1. *If $\theta = 1$, then there exists k_1 such that $\mathcal{L}(W^k) = \mathcal{L}^*$ for all $k > k_1$ and the algorithm terminates in finite steps.*
2. *If $\theta \in [\frac{1}{2}, 1)$, then there exists k_2 such that for all $k > k_2$,*

$$\mathcal{L}(W^k) - \mathcal{L}^* \leq \left(\frac{d_1 C^2}{1 + d_1 C^2}\right)^{k-k_2} r_{k_2}.$$

3. *If $\theta \in (0, \frac{1}{2})$, then there exists k_3 such that for all $k > k_3$,*

$$\mathcal{L}(W^k) - \mathcal{L}^* \leq \left(\frac{C}{(k-k_3)d_2(1-2\theta)}\right)^{\frac{1}{1-2\theta}},$$

where $\mathcal{L}^*$ is the same function value at all the accumulation points of $\{W^k\}, r_k = \mathcal{L}(V^k) - \mathcal{L}^*$, $d_1 = \left(\frac{1}{\alpha_2} + L\right)^2 / \left(\frac{1}{2\alpha_2} - \frac{L}{2}\right)$ and $d_2 = \min\left\{\frac{1}{2d_1 C}, \frac{C}{1-2\theta}\left(2^{\frac{2\theta-1}{2\theta-2}} - 1\right) r_1^{2\theta-1}\right\}$

Theorem 3.4.2 is a direct consequence of combining Theorem B.7, Theorem B.8, and Proposition A.4.

C. Proof of Statistical Results

C.1. Technical lemmas

Recall that we denote $R(\mathbf{W}) = \mathbb{E}(\|\boldsymbol{x}^{\mathrm{T}}\mathbf{W} - \boldsymbol{x}^{\mathrm{T}}\widehat{\mathbf{W}}\|^2) = \mathrm{Tr}((\mathbf{W} - \widehat{\mathbf{W}})^{\mathrm{T}}\boldsymbol{\Sigma}(\mathbf{W} - \widehat{\mathbf{W}}))$, where $\boldsymbol{\Sigma} = \mathbb{E}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})$. We denote the empirical risk as $\widetilde{R}(\mathbf{W}) = \|\mathbf{X}\mathbf{W} - \mathbf{X}\widehat{\mathbf{W}}\|_F^2 = \mathrm{Tr}((\mathbf{W} - \widehat{\mathbf{W}})^{\mathrm{T}}\widetilde{\boldsymbol{\Sigma}}(\mathbf{W} - \widehat{\mathbf{W}}))$, where $\mathbf{X} = \frac{1}{\sqrt{n}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_3 \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $\widetilde{\boldsymbol{\Sigma}} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$. Then we offer a lemma to give a generalization risk bound.

Lemma C.1. *Let $\mathcal{D}_x$ be a distribution over $x \in \mathbb{R}^{d_1}$ which is bounded, i.e. there exits a M_x, such that $\mathbb{P}_{x\sim\mathcal{D}_x}(\|\boldsymbol{x}\|_F \leq M_x) = 1$. Assume $\mathbf{X} = \frac{1}{\sqrt{n}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_3 \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$.Then for any $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, we have*

$$|R(\mathbf{W}) - \widetilde{R}(\mathbf{W})| \leq \sqrt{\frac{2M_x^4 \|\mathbf{W} - \widehat{\mathbf{W}}\|_1^4 d_2^2 \log(\frac{2d_1^2}{\delta})}{n}}$$

with probability at least $1 - \delta$,

Proof. By using Hoeffding's inequality, for any $j, k \in 1, \cdots, d_1$ we can get

$$|\boldsymbol{\Sigma}_{jk} - \widetilde{\boldsymbol{\Sigma}}_{jk}| = |\frac{1}{n}\sum_{i=1}^{n} x_{ij}x_{ik} - \mathbb{E}(x_{1j}x_{1k})| \leq \sqrt{\frac{2M_x^4 \log(\frac{2d_1^2}{\delta})}{n}}$$

with probability at least $1 - \delta/d_1^2$.

Denote $\Delta = \sqrt{\frac{2M_x^4 \log(\frac{2d_1^2}{\delta})}{n}}$. Then we have,

$$\mathbb{P}(\|\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}\|_\infty \leq \Delta) = 1 - \mathbb{P}(\bigcup_{j,k=1}^{d_1} \{|\mathbf{\Sigma}_{jk} - \widetilde{\mathbf{\Sigma}}_{jk}| \geq \Delta\}) \geq 1 - \sum_{j,k=1}^{d_1} \mathbb{P}(|\mathbf{\Sigma}_{jk} - \widetilde{\mathbf{\Sigma}}_{jk}|) \geq \Delta) \geq 1 - \delta$$

Hence, with probability at least $1-\delta$, for any $\mathbf{W}$, we have

$$\begin{aligned}
|R(\mathbf{W}) - \widetilde{R}(\mathbf{W})| &= |\operatorname{Tr}((\mathbf{W} - \widehat{\mathbf{W}})^{\mathrm{T}}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})(\mathbf{W} - \widehat{\mathbf{W}}))| \\
&\leq \sum_{i=1}^{d_2}\sum_{j=1}^{d_1}\sum_{k=1}^{d_1} |((\mathbf{W} - \widehat{\mathbf{W}})^{\mathrm{T}})_{ij}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})_{jk}(\mathbf{W} - \widehat{\mathbf{W}})_{ki}| \\
&\leq \Delta \sum_{i=1}^{d_2}\sum_{j=1}^{d_1}\sum_{k=1}^{d_1} |(\mathbf{W} - \widehat{\mathbf{W}})_{ji}(\mathbf{W} - \widehat{\mathbf{W}})_{ki}| \\
&\leq \Delta d_2 \|\mathbf{W} - \widehat{\mathbf{W}}\|_1^2.
\end{aligned}$$

□

We then give a useful lemma for RIP and extend it to a matrix version.

Lemma C.2. *(Candes & Tao, 2005a) Suppose matrix $\mathbf{A}$ satisfies the RIP of order k. Given a vector $\boldsymbol{u} \in \mathbb{R}^n$ and a set $\Omega \in [N]$, one has*

1. $\left\|\left(\left(\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)\boldsymbol{u}\right)_\Omega\right\|_2 \leq \delta_t \|\boldsymbol{u}\|_2$ *if* $|\Omega \cup \operatorname{Supp}(\boldsymbol{u})| \leq t$.
2. $\left\|(\mathbf{A}^T\boldsymbol{u})_\Omega\right\|_2 \leq \sqrt{1+\delta_t}\|\boldsymbol{u}\|_2$ *if* $|\Omega| \leq t$.

Corollary C.3. *For a matrix $\mathbf{A} \in \mathbb{R}^{n\times d_1}$ and matrix $\mathbf{W} \in \mathbb{R}^{d_1\times d_2}$, suppose matrix $\mathbf{I}_{d_2} \otimes \mathbf{A}$ satisfy the RIP of order k. Given a and a set $\Omega \in [d_1] \times [d_2]$, one has*

1. $\left\|\left(\left(\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)\mathbf{W}\right)_\Omega\right\|_F \leq \delta_t \|\mathbf{W}\|_F$ *if* $|\Omega \cup \operatorname{Supp}(\mathbf{W})| \leq t$.
2. $\left\|(\mathbf{A}^T\mathbf{W})_\Omega\right\|_F \leq \sqrt{1+\delta_t}\|\mathbf{W}\|_F$ *if* $|\Omega| \leq t$.

C.2. Proof of Theorem 3.6

Notice that by taking $\mathbf{X}^{'} = \sqrt{\alpha}\mathbf{X}, \alpha^{'} = 1, \lambda^{'} = \alpha\lambda.$, the IHT iteration remains the same. Without loss of generality, we first consider $\alpha = 1$.

Since from Proposition B.5, there exists a K, such that for any $k \geq K$, $\mathbf{W}^k$ have the same support as $\mathbf{W}^*$. Denote the support as S^*, and by the asumption, $|S^*| = s^*$. For $k > K$ and any $\mathbf{W}$ such that $\|\mathbf{W}\|_0 \leq s$, we denote $\mathbf{Y} = \mathbf{X}\widehat{\mathbf{W}}$, $\mathbf{\Phi} = \mathbf{Y} - \mathbf{X}\mathbf{W}$, $S = \operatorname{Supp}(\mathbf{W})$ and $\mathbf{G}^k = \mathbf{W}^{k-1} + \mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{W}^{k-1})$. Then by triangular inequality, we have

$$\begin{aligned}
\|\mathbf{W}^k - \mathbf{W}\|_F &= \|(\mathbf{G}^k)_{S^*} - \mathbf{W}_{S\cup S^*}\|_F \\
&\leq \|(\mathbf{G}^k)_{S^*} - (\mathbf{G}^k)_{S^*\cup S}\|_F + \|(\mathbf{G}^k)_{S^*\cup S} - \mathbf{W}_{S\cup S^*}\|_F \\
&\leq 2\|(\mathbf{G}^k)_{S^*\cup S} - \mathbf{W}_{S\cup S^*}\|_F.
\end{aligned} \tag{9}$$

The second inequality is due to $(\mathbf{G}^k)_{S^*}$ is the best s^*-sparse approximation of $(\mathbf{G}^k)_{S^*\cup S}$ and $\|\mathbf{W}\|_0 \leq s \leq s^*$.

$$\begin{aligned}
\|\mathbf{W}^k - \mathbf{W}\|_F &\leq 2\|(\mathbf{W}^{k-1} + \mathbf{X}^{\mathrm{T}}(\mathbf{\Phi} + \mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W}^{k-1}))_{S\cup S^*} - \mathbf{W}_{S\cup S^*}\|_F \\
&\leq 2\|((\mathbf{I} - \mathbf{X}^{\mathrm{T}}\mathbf{X})(\mathbf{W}^{k-1} - \mathbf{W}))_{S^*\cup S}\|_F + 2\|(\mathbf{X}^{\mathrm{T}}\mathbf{\Phi})_{S^*}\|_F \\
&\leq 2\delta_{s^*+s}\|\mathbf{W}^{k-1} - \mathbf{W}\|_F + 2\sqrt{1+\delta_{s^*+s}}\|\mathbf{\Phi}\|_F.
\end{aligned} \tag{10}$$

The third inequality is from Corollary C.3. Then by induction, it's easy to get

$$\|\mathbf{W}^k - \mathbf{W}\|_F \leq (2\delta_{s^*+s})^{K-k}\|\mathbf{W}^K - \mathbf{W}\|_F + \frac{2\sqrt{1+\delta_{s^*+s}}}{1-2\delta_{s^*+s}}\|\mathbf{\Phi}\|_F.$$

From Proposition B.5, we have $\mathcal{L}(\mathbf{W}^K) \leq \mathcal{L}(\mathbf{W}^0) = \mathcal{L}(\widehat{\mathbf{W}}) \leq \lambda d_1 d_2$. Hence, we have

$$\begin{aligned}\sqrt{1-\delta_{s^*+s}}\|\mathbf{W} - \mathbf{W}^K\|_F &\leq \|\mathbf{X}(\mathbf{W} - \mathbf{W}^K) + \mathbf{\Phi}\|_F + \|\mathbf{\Phi}\|_F \\ &\leq \|\mathbf{Y} - \mathbf{X}\mathbf{W}^K\|_F + \|\mathbf{\Phi}\|_F \\ &\leq \sqrt{2\mathcal{L}(\mathbf{W}^K)} + \|\mathbf{\Phi}\|_F \\ &\leq \sqrt{2\lambda d_1 d_2} + \|\mathbf{\Phi}\|_F\end{aligned} \tag{11}$$

and

$$\|\mathbf{Y} - \mathbf{X}\mathbf{W}^k\|_F \leq \|\mathbf{\Phi}\|_F + \|\mathbf{X}(\mathbf{W} - \mathbf{W}^k)\|_F \leq \|\mathbf{\Phi}\|_F + \sqrt{1+\delta_{s^*+s}}\|\mathbf{W}^k - \mathbf{W}\|_F \tag{12}$$

Hence by combining (10), (11) and (12), we have

$$\begin{aligned}\widetilde{R}(\mathbf{W}^k) &= \|\mathbf{Y} - \mathbf{X}\mathbf{W}^k\|_F^2 \\ &\leq ((1 + \frac{2(1+\delta_{s^*+s})}{1-2\delta_{s^*+s}})\|\mathbf{\Phi}\|_F + \sqrt{1+\delta_{s^*+s}}(2\delta_{s^*+s})^{k-K}\|\mathbf{W}^K - \mathbf{W}\|_F)^2 \\ &\leq ((1 + \frac{2(1+\delta_{s^*+s})}{1-2\delta_{s^*+s}})\|\mathbf{\Phi}\|_F + \sqrt{3}(2\delta_{s^*+s})^{k-K}(\sqrt{2\lambda d_1 d_2} + \|\mathbf{\Phi}\|_F))^2 \\ &= \left(C\sqrt{\widetilde{R}(\mathbf{W})} + \sqrt{3}(\sqrt{2\lambda d_1 d_2} + \|\mathbf{\Phi}\|_F)(2\delta_{s^*+s})^{k-K}\right)^2\end{aligned} \tag{13}$$

where $C = 1 + \frac{2(1+\delta_{s^*+s})}{1-2\delta_{s^*+s}}$. It's easy to see that $C \leq 1 + \frac{4}{1-2\tau} = O\left(\frac{1}{1-2\tau}\right)$.

Denote $\mathcal{B}(\mathbf{W}) = \sqrt{\frac{2M_x^4\|\mathbf{W}-\widehat{\mathbf{W}}\|_1^4 d_2^2 \log(\frac{2d_1^2}{\delta})}{n}}$. We choose a fixed $\mathbf{W}^+ \in \arg\min_{\mathbf{W},\|\mathbf{W}\|_0 \leq s} \mathbb{E}(\|\boldsymbol{x}^{\mathrm{T}}\mathbf{W} - \boldsymbol{x}^{\mathrm{T}}\widehat{\mathbf{W}}\|^2)$ and let $\mathbf{W}$ in (13) equals to $\mathbf{W}^+$. Then $\mathcal{B}(\mathbf{W}^+) = O(\sqrt{\frac{\log(2d_2^2/\delta)}{n}})$. Since we assume $\frac{\log(1/\delta)}{n} = O(1)$, with probability at least $1-\delta$, $|\widetilde{R}(\mathbf{W}^+) - R(\mathbf{W}^+)| \leq O(1)$. Hence we have $\sqrt{3}(\sqrt{2\lambda d_1 d_2} + \|\mathbf{\Phi}\|_F) = O(\sqrt{\lambda} + \sqrt{\widetilde{R}(\mathbf{W}^+)}) \leq O(\sqrt{\lambda} + 1)$. Similar to (11), we also have

$$\|\mathbf{W}^+ - \mathbf{W}^k\|_F \leq \frac{\sqrt{2\lambda d_1 d_2} + \sqrt{\widetilde{R}(\mathbf{W}^+)}}{\sqrt{1-\delta_{s^*+s}}} \leq \frac{\sqrt{2\lambda d_1 d_2} + \sqrt{R(\mathbf{W}^+) + O(1)}}{\sqrt{1/2}} = O(\sqrt{\lambda} + 1). \tag{14}$$

Hence, we have

$$\begin{aligned}\|\mathbf{W}^k - \widehat{\mathbf{W}}\|_1 &\leq \|\mathbf{W}^k - \mathbf{W}^+\|_1 + \|\widehat{\mathbf{W}} - \mathbf{W}^+\|_1 \\ &\leq \sqrt{d_1}\|\mathbf{W}^k - \mathbf{W}^+\|_F + \|\widehat{\mathbf{W}} - \mathbf{W}^+\|_1 \\ &= O(\sqrt{\lambda} + 1)\end{aligned}$$

This shows that

$$\mathcal{B}(\mathbf{W}^k) = O\left(\sqrt{\frac{(\sqrt{\lambda}+1)^4 \log(2d_1^2/\delta)}{n}}\right). \tag{15}$$

Then we have

$$\begin{aligned}
R(\mathbf{W}^k) &\leq \widetilde{R}(\mathbf{W}^k) + \mathcal{B}(\mathbf{W}^k) \\
&\leq \left(C\sqrt{\widetilde{R}(\mathbf{W}^+)} + O\left(\sqrt{\lambda}+1\right)(2\delta_{s^*+s})^{k-K}\right)^2 + \mathcal{B}(\mathbf{W}^k) \\
&\leq \left(C\sqrt{R(\mathbf{W}^+) + \mathcal{B}(\mathbf{W}^+)} + O\left(\sqrt{\lambda}+1\right)(2\delta_{s^*+s})^{k-K}\right)^2 + \mathcal{B}(\mathbf{W}^k) \\
&= C^2 R(\mathbf{W}^+) + O\left(C(\sqrt{\lambda}+1)\right)(2\delta_{s^*+s})^{k-K} + O\left((\sqrt{\lambda}+1)^2\right)(2\delta_{s^*+s})^{2(k-K)} \\
&\quad + C^2\mathcal{B}(\mathbf{W}^+) + \mathcal{B}(\mathbf{W}^k) \\
&\leq C^2 R(\mathbf{W}^+) + O\left(\frac{\sqrt{\lambda}+1}{1-2\tau} + (\sqrt{\lambda}+1)^2\right)(2\delta_{s^*+s})^{k-K} + O\left(\frac{(\sqrt{\lambda}+1)^2}{(1-2\tau)^2}\sqrt{\frac{\log(2d_1^2/\delta)}{n}}\right)
\end{aligned} \tag{16}$$

Hence, for any $\alpha \leq 1/L$, when $\mathbf{I}_{d_2} \otimes \sqrt{\alpha}\mathbf{X}$ satisfies $\delta_{s+s^*} \leq \tau$, take λ in (16) equal to $\alpha\lambda$. Since we assume $\alpha = O(1)$ and $\lambda = O(1)$, we have

$$R(\mathbf{W}^k) \leq C^2 R(\mathbf{W}^+) + O(\frac{1}{1-2\tau})(2\delta_{s^*+s})^{k-K} + O\left(\frac{1}{(1-2\tau)^2}\sqrt{\frac{\log(2d_1^2/\delta)}{n}}\right).$$

with probability $1-\delta-\xi$. This completes the proof.

C.3. Proof of Theorem 3.8

Since we assume that all accumulation points of $\{\mathbf{W}^k\}$ have the same support size s^*, there exists a K, such that for $k > K$, $\|\mathbf{W}^k\|_0 = s^*$, $\|\mathbf{V}^{k+1}\|_0 = s^*$. Otherwise, for any $N \in \mathbb{N}$, there exists a k_N, such that $\|\mathbf{W}^{k_N}\|_0 \neq s^*$ or $\|\mathbf{V}^{k_N}\|_0 \neq s^*$. From Proposition B.6 and Theorem 3.3, we know that $\mathbf{V}^k$ and $\mathbf{W}^k$ are bounded and share the same accumulation points. . Since the absolute value of any non-zero entries of $\mathbf{W}^k$ and $\mathbf{V}^k$ is greater than $\sqrt{2\lambda\min\{\alpha_1,\alpha_2\}}$, we conclude that there exists an accumulation point $\mathbf{W}^*$ of either $\mathbf{W}^{k_N}$ or $\mathbf{V}^{k_N}$, and we have $\|\mathbf{W}^*\|_0 \neq s^*$, which leads to a contradiction.

Denote $\alpha = \alpha_1 = \alpha_2$. Notice that by taking $\mathbf{X}^{'} = \sqrt{\alpha}\mathbf{X}, \alpha^{'} = 1, \lambda^{'} = \alpha\lambda,$, the mAIHT iteration also remains the same. Without loss of generality, we first consider $\alpha = 1$. For any $\mathbf{W}$ such that $\|\mathbf{W}\|_0 \leq s$, we denote $\mathbf{Y} = \mathbf{X}\widehat{\mathbf{W}}$, $\mathbf{\Phi} = \mathbf{Y} - \mathbf{X}\mathbf{W}$, $S^k = \text{Supp}(\mathbf{V}^k)$ and $S = \text{Supp}(\mathbf{W})$. Then for any $k > K$, we have

$$\begin{aligned}
\|\mathbf{Y} - \mathbf{X}\mathbf{W}^{k+1}\|_F &= \|\mathbf{X}(\mathbf{W} - \mathbf{W}^{k+1}) + \mathbf{\Phi}\|_F \\
&\geq \sqrt{1-\delta_{s^*+s}}\|\mathbf{W} - \mathbf{W}^{k+1}\|_F - \|\mathbf{\Phi}\|_F.
\end{aligned} \tag{17}$$

Similar to (9), we also have

$$\begin{aligned}
\|\mathbf{V}^{k+1} - \mathbf{W}\|_F &\leq 2\|(\mathbf{W}^k - \mathbf{W} + \mathbf{X}^{\mathrm{T}}(\mathbf{\Phi} + \mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W}^k))_{S^{k+1}\cup S}\|_F \\
&\leq 2\|((\mathbf{I} - \mathbf{X}^{\mathrm{T}}\mathbf{X})(\mathbf{W}^k - \mathbf{W}))_{S^{k+1}\cup S}\|_F + 2\|(\mathbf{X}^{\mathrm{T}}\mathbf{\Phi})_{S^{k+1}\cup S}\|_F \\
&\leq 2\delta_{s^*+s}\|\mathbf{W}^k - \mathbf{W}\|_F + 2\sqrt{1+\delta_{s^*+s}}\|\mathbf{\Phi}\|_F.
\end{aligned} \tag{18}$$

The second inequality is due to Corollary C.3. By repeatedly applying the definition of RIP, we have

$$\begin{aligned}
\sqrt{1-\delta_{s^*+s}}\|\mathbf{W} - \mathbf{W}^{k+1}\|_F &\leq \|\mathbf{X}(\mathbf{W} - \mathbf{W}^{k+1})\|_F \\
&\leq \|\mathbf{Y} - \mathbf{X}\mathbf{W}^{k+1}\|_F + \|\mathbf{\Phi}\|_F \\
&\leq \|\mathbf{Y} - \mathbf{X}\mathbf{V}^{k+1}\|_F + \|\mathbf{\Phi}\|_F \\
&\leq \|\mathbf{X}(\mathbf{W} - \mathbf{V}^{k+1})\|_F + 2\|\mathbf{\Phi}\|_F \\
&\leq \sqrt{1+\delta_{s^*+s}}\|\mathbf{W} - \mathbf{V}^{k+1}\|_F + 2\|\mathbf{\Phi}\|_F
\end{aligned} \tag{19}$$

The third inequality is due to $\mathcal{L}(\mathbf{W}^{k+1}) \leq \mathcal{L}(\mathbf{V}^{k+1})$ and $\|\mathbf{W}^{k+1}\|_0 = \|\mathbf{V}^{k+1}\|_0 = s^*$. Combine (18) and (19), we have

$$\sqrt{1-\delta_{s^*+s}}\|\mathbf{W}-\mathbf{W}^{k+1}\|_F \leq 2\sqrt{1+\delta_{s^*+s}}\delta_{s^*+s}\|\mathbf{W}-\mathbf{W}^{k}\|_F + (4+2\delta_{s^*+s})\|\mathbf{\Phi}\|_F.$$

We denote $\rho = 2\frac{\sqrt{1+\delta_{s^*+s}}\delta_{s^*+s}}{\sqrt{1-\delta_{s^*+s}}}$. Since we assume that $\delta_{s^*+s} \leq \tau < u \approx 0.34781$, where u is the real root of $4u^3+4u^2+u=1$, we have $\rho \leq \rho_\tau < 1$, where $\rho_\tau = \frac{2\sqrt{1+\tau}\tau}{\sqrt{1-\tau}}$. Then we can get

$$\|\mathbf{W}^k - \mathbf{W}\|_F \leq \rho^{k-K}\|\mathbf{W}^K - \mathbf{W}\|_F + \frac{4+2\delta_{s^*+s}}{\sqrt{1-\delta_{s^*+s}}(1-\rho)}\|\mathbf{\Phi}\|_F.$$

Similar to analysis of IHT, by (12) and (11), we have

$$\begin{aligned}
\widetilde{R}(\mathbf{W}^k) &= \|\mathbf{Y}-\mathbf{X}\mathbf{W}^k\|_F^2 \\
&\leq (\sqrt{1+\delta_{s+s^*}}\|\mathbf{W}^k-\mathbf{W}\|_F + \|\mathbf{\Phi}\|_F)^2 \\
&\leq ((1+\frac{\sqrt{1+\delta_{s^*+s}}(4+2\delta_{s^*+s})}{\sqrt{1-\delta_{s^*+s}}(1-\rho)})\|\mathbf{\Phi}\|_F + \sqrt{1+\delta_{s^*+s}}\rho^{k-K}\|\mathbf{W}^K-\mathbf{W}\|_F)^2 \\
&\leq (C\|\mathbf{\Phi}\|_F + \frac{2(\sqrt{2\lambda d_1 d_2}+\|\mathbf{\Phi}\|_F)}{\sqrt{1-\tau}}\rho^{k-K})^2
\end{aligned} \tag{20}$$

where $C = 1+\frac{\sqrt{1+\delta_{s^*+s}}(4+2\delta_{s^*+s})}{\sqrt{1-\delta_{s^*+s}}(1-\rho)}$. It's easy to see that

$$C \leq C_\tau = 1 + \frac{\sqrt{1+\tau}(3+\tau)}{\sqrt{1-\tau}(1-\rho_\tau)} = O(\frac{1}{1-\rho_\tau})$$

Denote $\mathcal{B}(\mathbf{W}) = \sqrt{\frac{2M_x^4\|\mathbf{W}-\widehat{\mathbf{W}}\|_1^4 d_2^2 \log(\frac{2d_1^2}{\delta})}{n}}$. We choose a fixed $\mathbf{W}^+ \in \arg\min_{\mathbf{W},\|\mathbf{W}\|_0\leq s}\mathbb{E}(\|\boldsymbol{x}^{\mathrm{T}}\mathbf{W}-\boldsymbol{x}^{\mathrm{T}}\widehat{\mathbf{W}}\|^2)$. Take $\mathbf{W}$ in (20) equal to $\mathbf{W}^+$. Since we assume $\frac{\log(1/\delta)}{n} = O(1)$, from Lemma C.1, with probability at least $1-\delta$, $\widetilde{R}(\mathbf{W}^+) \leq R(\mathbf{W}^+) + O(1)$. Hence we have $2(\sqrt{2\lambda d_1 d_2}+\|\mathbf{\Phi}\|_F) = O(\sqrt{\lambda d_1 d_2} + \sqrt{\widetilde{R}(\mathbf{W}^+)}) \leq O(\sqrt{\lambda}+1)$ and $\mathcal{B}(\mathbf{W}^k) \leq O\left(\sqrt{\frac{(\sqrt{\lambda}+1)^4\log(2d_1^2/\delta)}{n}}\right)$(similar to (15)). Then we have

$$\begin{aligned}
R(\mathbf{W}^k) &\leq \widetilde{R}(\mathbf{W}^k) + \mathcal{B}(\mathbf{W}^k) \\
&\leq \left(C\sqrt{\widetilde{R}(\mathbf{W}^+)} + O\left(\sqrt{\lambda}+1\right)\rho^{k-K}\right)^2 + \mathcal{B}(\mathbf{W}^k) \\
&\leq \left(C\sqrt{R(\mathbf{W}^+)+\mathcal{B}(\mathbf{W}^+)} + O\left(\sqrt{\lambda}+1\right)\rho^{k-K}\right)^2 + \mathcal{B}(\mathbf{W}^k) \\
&\leq C^2 R(\mathbf{W}^+) + O\left(C_\tau(\sqrt{\lambda}+1)\right)\rho^{k-K} + O\left((\sqrt{\lambda}+1)^2\right)\rho^{2(k-K)} + C_\tau^2\mathcal{B}(\mathbf{W}^+) + \mathcal{B}(\mathbf{W}^k) \\
&\leq C^2 R(\mathbf{W}^+) + O(\frac{(\sqrt{\lambda}+1)^2}{1-\rho_\tau})\rho^{k-K} + O\left((\sqrt{\lambda}+1)^2\sqrt{\frac{\log(2d_1^2/\delta)}{n(1-\rho_\tau)^4}}\right).
\end{aligned} \tag{21}$$

Hence, for any $0 < \alpha \leq 1/L$, when $\mathbf{I}_{d_2} \otimes \sqrt{\alpha}\mathbf{X}$ satisfies $\delta_{s+s^*} \leq \tau$, since we assume $\alpha = O(1), \lambda = O(1)$, we have

$$\begin{aligned}
R(\mathbf{W}^k) &\leq C^2 R(\mathbf{W}^+) + O(\frac{(\sqrt{\alpha\lambda}+1)^2}{1-\rho_\tau})\rho^{k-K} + O\left((\sqrt{\alpha\lambda}+1)^2\sqrt{\frac{\log(2d_1^2/\delta)}{n(1-\rho_\tau)^4}}\right) \\
&= C^2 \min_{\mathbf{W},\|\mathbf{W}\|_0\leq s} R(\mathbf{W}) + O(\frac{1}{1-\rho_\tau})\rho^{k-K} + O\left(\sqrt{\frac{\log(2d_1^2/\delta)}{n(1-\rho_\tau)^4}}\right),
\end{aligned}$$

with probability $1-\delta-\xi$.

This completes the proof.

D. Additional Experiments

In this section, we present experiments comparing the pruning time and performance of mAIHT with Wanda and SparseGPT. We also evaluate how the number of iterations in mAIHT affects both runtime and perplexity, showing its flexibility in balancing speed and accuracy.

D.1. Results on Larger Model

We conducted additional experiments on the LLaMA-13B model with 50% sparsity, evaluating its performance on Wikitext-2 perplexity as well as zero-shot tasks. The results are summarized in the tables below:

Table 3. Perplexity on Wikitext-2 (LLaMA-13B, 50% Sparsity)

Method	Perplexity
SparseGPT	6.2535
mAIHT	**6.1336**

Table 4. Zero-Shot Downstream Task Accuracy (%, LLaMA-13B, 50% Sparsity)

Method	BoolQ	RTE	HellaSwag	ARC-e	ARC-c	WinoGrande	OBQA	Mean
SparseGPT	76.06	**60.28**	74.00	67.55	41.89	71.98	44.20	62.28
mAIHT	**75.47**	**60.28**	**75.08**	**70.03**	**44.88**	**71.42**	**44.80**	**63.14**

These results indicate that mAIHT outperforms SparseGPT both in terms of Wikitext-2 perplexity and the average performance across downstream tasks. mAIHT also achieves higher performance on individual tasks, demonstrating its effectiveness for model pruning.

D.2. Runtime Comparison

We compare the pruning times (including activation collection) for the LLaMA-7B model across three methods. All experiments were performed with an A100 GPU.

Table 5. Pruning Time on LLaMA-7B (in Seconds)

Method	Time (s)
Wanda	148.55
SparseGPT	609.04
mAIHT (50 iters)	1370.79

Although mAIHT incurs higher pruning times due to its iterative gradient-based optimization process, it remains significantly more efficient than full model fine-tuning. For instance, the Wanda paper (Sun et al., 2023) reports that LoRA fine-tuning on LLaMA-7B takes approximately 24 hours on a single V100 GPU, and full fine-tuning can take up to 24 days. In contrast, mAIHT completes pruning in under 30 minutes on an A100 GPU, offering a practical and efficient alternative to fine-tuning.

Importantly, mAIHT also provides a controllable trade-off between pruning time and performance. As discussed in Remark 2.2, a single iteration of mAIHT is equivalent to magnitude pruning, and with pre-pruning normalization, this reduces to Wanda. Thus, mAIHT can be seen as a natural generalization of one-shot methods like Wanda and SparseGPT. By increasing the number of iterations, mAIHT improves performance progressively while keeping the computational cost manageable.

The following table illustrates how performance improves with more iterations of mAIHT when pruning LLaMA-7B to 50% sparsity:

Table 6. mAIHT Runtime and Perplexity at 50% Sparsity on LLaMA-7B

Method	Wanda (1 iter)	20 iters	50 iters	100 iters
Perplexity	7.2588	7.1876	7.0720	7.0765
Time (s)	148	860	1370	2286

These results show that mAIHT steadily improves performance as the number of iterations increases, reaching the best perplexity at 50 iterations. The slight increase in perplexity at 100 iterations is likely due to overfitting to the calibration data. Runtime scales approximately linearly with the number of iterations, enabling users to balance speed and accuracy based on their computational budget. Therefore, mAIHT provides a flexible and effective pruning strategy that offers much of the performance benefits of fine-tuning with only a fraction of the computational cost, and clear advantages over static one-shot methods.

D.3. Support for Structured Pruning

The mAIHT framework can be extended to support structured sparsity constraints. Specifically, n:m sparsity can be incorporated by replacing the $\lambda\|W\|_0$ term in the optimization objective with an indicator function $I_S(W)$, where S denotes the set of matrices that satisfy the n:m sparsity pattern. The indicator function I_S takes a value of 0 when $W \in S$ and $+\infty$ otherwise. In this case, the proximal step in mAIHT reduces to a projection onto S, replacing the standard hard-thresholding operator.

Empirical results on LLaMA-7B with 2:4 sparsity are provided in the following tables. As shown, mAIHT outperforms SparseGPT in terms of Wikitext-2 perplexity and achieves competitive or superior performance across zero-shot tasks.

Table 7. Perplexity on Wikitext-2 (LLaMA-7B, 2:4 Sparsity)

Method	Perplexity
SparseGPT	7.2933
mAIHT	**7.2606**

Table 8. Zero-Shot Downstream Task Accuracy (%, LLaMA-7B, 2:4 Sparsity)

Method	BoolQ	RTE	HellaSwag	ARC-e	ARC-c	WinoGrande	OBQA	Mean
SparseGPT	**73.79**	54.51	69.28	**66.75**	39.33	**68.42**	38.60	58.67
mAIHT	72.96	**60.28**	**69.56**	64.89	**40.27**	67.48	**39.40**	**59.26**

Additionally, the mAIHT framework is compatible with various other structured sparsity patterns, including hierarchical sparsity (Wu et al., 2023), block sparsity (Gray et al., 2017), and row sparsity (Meng et al., 2024b). For each of these sparsity patterns, the optimization objective is modified by replacing the ℓ_0 term with the indicator function $I_S(W)$, and the projection step is adapted accordingly.

Although the current theoretical analysis of mAIHT is focused on unstructured sparsity and does not directly extend to structured sparsity due to the geometric and combinatorial differences in the sparsity set S, future work will aim to extend these theoretical guarantees to structured sparsity.

D.4. Effect of Gradient-Based Refinement

We conducted an ablation study on LLaMA-7B at 50% sparsity to evaluate the effect of the gradient-based refinement step on Wikitext-2 perplexity. The results are shown below:

Table 9. Ablation Study on Refinement (Wikitext-2 PPL, LLaMA-7B, 50% Sparsity)

Method	Perplexity
SparseGPT	7.2397
mAIHT (w/o Refinement)	7.0843
mAIHT (Full)	**7.0720**

Even without explicit refinement, mAIHT achieves lower perplexity than SparseGPT, indicating that the proximal gradient steps implicitly refine the retained weights. The explicit refinement step further improves performance, confirming the effectiveness of the gradient-based refinement mechanism in mAIHT.