On Computational Limits of Modern Hopfield Models: A Fine-Grained Complexity Analysis

Jerry Yao-Chieh Hu [*1] **Thomas Lin** [*2] **Zhao Song** [3] **Han Liu** [4]

Abstract

We investigate the computational limits of the memory retrieval dynamics of modern Hopfield models from the fine-grained complexity analysis. Our key contribution is the characterization of a phase transition behavior in the efficiency of all possible modern Hopfield models based on the norm of patterns. Specifically, we establish an upper bound criterion for the norm of input query patterns and memory patterns. Only below this criterion, sub-quadratic (efficient) variants of the modern Hopfield model exist, assuming the Strong Exponential Time Hypothesis (SETH). To showcase our theory, we provide a formal example of efficient constructions of modern Hopfield models using low-rank approximation when the efficient criterion holds. This includes a derivation of a lower bound on the computational time, scaling linearly with $\max\{$# of stored memory patterns, length of input query sequence$\}$. In addition, we prove its memory retrieval error bound and exponential memory capacity.

1. Introduction

We investigate the computational limits of modern Hopfield models (Wu et al., 2024a;b; Hu et al., 2024a;b; 2023; Ramsauer et al., 2021) from a fine-grained complexity analysis, and characterize a norm-based phase transition for all possible efficient modern Hopfield model. This analysis holds practical significance. Modern Hopfield models are a type of associative memory model compatible with deep learning. More precisely, their deep learning derivatives offer robust alternatives to attention mechanisms in various transformer- and Hopfield-based methods (Hofmann et al., 2024; Xu et al., 2024; Wu et al., 2024a;b; Hu et al., 2024a; Schimunek et al., 2023; Fürst et al., 2022; Paischer et al., 2022; Seidl et al., 2022; Widrich et al., 2020). However, these models currently lack efficient implementations for large-scale applications (Hu et al., 2023, Section C.2). This issue becomes more relevant with the rise of Large Foundation Models (Bommasani et al., 2021), where expansive attention-based architectures, pre-trained on vast datasets, are pivotal across multiple scientific fields, including natural language processing (Brown et al., 2020; Floridi and Chiriatti, 2020), financial analytics (Wu et al., 2023), genomic research (Zhou et al., 2024; 2023; Ji et al., 2021), medical science (Thirunavukarasu et al., 2023; Singhal et al., 2023; Moor et al., 2023) and more. This work makes a timely theoretical analysis of their computational limits, aimed at advancing (Hopfield-based) large foundation models.

Let $\mathbf{x} \in \mathbb{R}^d$ be the input query pattern. The memory patterns are stored in a matrix $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_M] \in \mathbb{R}^{d\times M}$. Hopfield models are energy-based associative memory models. These models store memory patterns $\boldsymbol{\Xi}$ on the local minima of their energy landscapes, i.e. energy functions E. For any input query $\mathbf{x}$, they retrieve its closest memory pattern through some energy minimization algorithms, i.e. retrieval dynamics $\mathcal{T}$, initialized at $\mathbf{x}$.

Ramsauer et al. (2021) propose the Modern Hopfield Model with a specific set of energy function E and memory retrieval dynamics $\mathcal{T}$, and integrate it into deep learning architectures via its connection with the transformer attention (Vaswani et al., 2017), offering enhanced performance, and theoretically guaranteed exponential memory capacity. Specifically, they introduce the energy function:

$$E(\mathbf{x}) = -\operatorname{lse}(\beta, \boldsymbol{\Xi}^\mathsf{T}\mathbf{x}) + \frac{1}{2}\langle \mathbf{x}, \mathbf{x}\rangle, \tag{1.1}$$

where the retrieval dynamics is given by

$$\mathbf{x}^{\text{new}} = \mathcal{T}_{\text{Dense}}(\mathbf{x}) = \boldsymbol{\Xi}\cdot \operatorname{Softmax}(\beta\boldsymbol{\Xi}^\mathsf{T}\mathbf{x}). \tag{1.2}$$

The function $\operatorname{lse}(\beta, \mathbf{z}) := \log\left(\sum_{\mu=1}^M \exp\{\beta z_\mu\}\right)/\beta$ is the log-sum-exponential for any given vector $\mathbf{z} \in \mathbb{R}^M$ and $\beta >$

[*]Equal contribution [1]Department of Computer Science, Northwestern University, Evanston, IL, USA [2]Department of Physics, National Taiwan University, Taipei, Taiwan [3]Adobe Research, Seattle, WA, USA [4]Department of Statistics and Data Science, Northwestern University, Evanston, IL, USA. Correspondence to: Jerry Yao-Chieh Hu <jhu@u.northwestern.edu>, Thomas Lin <b12202026@ntu.edu.tw>, Zhao Song <zsong@adobe.com>, Han Liu <hanliu@northwestern.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

0. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L] \in \mathbb{R}^{d \times L}$ be a sequence of input queries, such that (1.2) becomes $\mathbf{Z} := [\mathbf{x}_1^{\text{new}}, \ldots, \mathbf{x}_L^{\text{new}}] = \mathcal{T}_{\text{Dense}}(\mathbf{X})$, and hence

$$\mathcal{T}_{\text{Dense}}(\mathbf{X}) = \overbrace{\mathbf{\Xi}}^{\in \mathbb{R}^{d \times M}} \cdot \operatorname{Softmax}(\beta \overbrace{\underbrace{\mathbf{\Xi}^{\mathsf{T}}}_{\in \mathbb{R}^{M \times d}} \underbrace{\mathbf{X}}_{\in \mathbb{R}^{d \times L}}}^{\in \mathbb{R}^{M \times L}}) \in \mathbb{R}^{d \times L},$$

where the $\operatorname{Softmax}(\cdot)$ applies column-wise normalization[1]. Here we assume $d = L^{o(1)}$, i.e., the growth rate of this function is sub-polynomial concerning L.

To motivate the study of possible efficient implementations, we make the following observation on (1.1):

> The bottleneck of Hopfield-based methods is the time to perform matrix multiplication in memory retrieval: $\mathcal{O}(dML)$. Namely, (1.1) is inefficient with $M = \Omega(e^d)$ (large memory set) and $L = \Omega(e^d)$ (long query sequences).

Explicitly, if the associative space is d-dimensional, this necessitates d multiplication operations for the inner products of $\{\mathbf{x}\}$ and $\{\boldsymbol{\xi}\}$. Consequently, the complexity of computing a dot product is $\mathcal{O}(d)$. Each pattern in $\mathbf{Z}$ must associate with every pattern in $\mathbf{\Xi}$. Therefore, the time complexity for sequences of length L and M with a pattern dimension of d is $\mathcal{O}(dML)$. In this regard, this work aims to characterize the fundamental limits on improving $\mathcal{O}(dML)$. Specifically, we ask the following questions:

Question 1. Is it possible to improve the time complexity $\mathcal{O}(dML)$ with a controllable approximation error?

Question 2. More aggressively, is it possible to perform memory retrieval computations in almost linear time $L^{1+o(1)}$ or $M^{1+o(1)}$ or $(L+M)^{1+o(1)}$?

To address these questions, we explore approximate retrieval computations with precision guarantees. We aim to find a surrogate $\mathcal{T}_{\text{approx.}}$ (also denoted as $\widetilde{\mathcal{T}}_{\text{Dense}}$) for $\mathcal{T}_{\text{Dense}}$ such that

$$\left\| \mathcal{T}_{\text{approx.}} - \mathcal{T}_{\text{Dense}} \right\|_{\max} \leq \delta_{\text{approx.}},$$

for some $\delta_{\text{approx.}} > 0$, where $\|\mathbf{A}\|_{\max} := \max_{i,j} |a_{ij}|$.

To be concrete, we study the following approximation problem with the realistic setting $\delta_{\text{approx.}} = 1/\text{poly}(L)$.

Problem 1 (Approximate Modern Hopfield Memory Retrieval Dynamics $\texttt{AHop}(d, M, L, \beta, B, \delta_H)$)**.** Let $\delta_H > 0$. Given $\mathbf{\Xi} \in \mathbb{R}^{d \times M}$ and $\mathbf{X} \in \mathbb{R}^{d \times L}$ such that $\|\mathbf{\Xi}\|_{\max} \leq B$ and $\|\mathbf{X}\|_{\max} \leq B$. We aim to study an approximation problem $\texttt{AHop}(d, M, L, \beta, B, \delta_H)$, that approximates $\mathbf{Z}$ with a matrix $\widetilde{\mathbf{Z}} := \widetilde{\mathcal{T}}_{\text{Dense}}(\mathbf{X})$ such that

$$\left\| \widetilde{\mathbf{Z}} - \mathbf{\Xi}\mathbf{D}^{-1}\mathbf{A} \right\|_{\max} \leq \delta_H,$$

where $\mathbf{\Xi}\mathbf{D}^{-1}\mathbf{A} = \mathbf{Z}$ with

$$\mathbf{A} = \exp\{\beta \mathbf{\Xi}^{\mathsf{T}} \mathbf{X}\}, \quad \mathbf{D} = \operatorname{diag}(\mathbf{A}\mathbf{1}_M).$$

[1] Many existing works denote $\mathbf{Z}$ by $\mathbf{X}^{\text{new}}$.

In this work, we aim to investigate the computational limits and potential efficient algorithms of $\texttt{AHop}$.

Contributions. Our contributions are threefold:

- **Computational Limits.** We answer Question 1 by identifying a phase transition behavior on the norm of query and memory patterns assuming the Strong Exponential Time Hypothesis (SETH). Explicitly, let $\tau = \max\{M, L\}$ be the upper bound of the patterns' lengths. We prove an upper bound criterion $B^\star = \Theta(\sqrt{\log \tau})$ for $\|\mathbf{\Xi}\|_{\max}$ and $\|\mathbf{X}\|_{\max}$ such that, only below which, solving $\texttt{AHop}$ in $\tau^{2-\Omega(1)}$ (sub-quadratic) time is possible.
- **Efficient Model.** We answer Question 2 by providing an efficient algorithm for $\texttt{AHop}$ based on low-rank approximation: an almost linear time modern Hopfield model. Explicitly, we prove that the algorithm, under realistic settings, performs the computation in almost linear time $\tau^{1+o(1)}$.
- **Exponential Memory Capacity.** Focusing on the almost-linear-time modern Hopfield model, we derive its retrieval error bound and show that this model achieves almost-linear-time efficiency while maintaining the exponential memory capacity characteristic of modern Hopfield models.

Background and Related Works

Modern Hopfield Models for Deep Learning. Classical Hopfield models (Hopfield, 1984; 1982; Krotov and Hopfield, 2016) emulate human brain associative memory by focusing on storing and retrieving memory patterns. In machine learning community, a noticeable interest in these models arises from (i) improved memory storage capacities (from linear to polynomial (Krotov and Hopfield, 2016), to exponential (Demircigil et al., 2017) and to kernelized (Wu et al., 2024a)), (ii) novel architectures (Hoover et al., 2023; Seidl et al., 2022; Fürst et al., 2022), and (iii) their biological plausibility (Kozachkov et al., 2022; Krotov and Hopfield, 2021). Notably, the modern Hopfield models (Hu et al., 2024a;b; 2023; Wu et al., 2024a;b; Burns and Fukai, 2023; Brandstetter, 2021; Ramsauer et al., 2021) offer fast convergence and expanded memory capacity. Importantly, they serve as advanced extensions of attention mechanisms to Transformer architecture. They have extensive applications in diverse fields like tabular learning (Xu et al., 2024), drug discovery (Schimunek et al., 2023), immunology (Widrich et al., 2020), time series forecasting (Wu et al., 2024b; Auer et al., 2024), reinforcement learning (Paischer et al., 2022), and large foundation models (Hu et al., 2024a; Fürst et al., 2022).

Theory of Modern Hopfield Models. Besides empirical successes, Modern Hopfield Models provide a model-based theoretical framework for analyzing transformer attention and Transformer architectures. (Hu et al., 2023) and Wu et al. (2024b) propose a unified framework to analyze and derive modern Hopfield models via entropic regularizers. Significantly, their work presents sparse variants (sparse and generalized sparse models) and incorporates the standard modern Hopfield model (Ramsauer et al., 2021) as a particular example in their framework. Yet, they also note that the modern Hopfield paradigm is incomplete and lacks efficient implementations or variants (Hu et al., 2023, Section E). Extending this foundation, Hu et al. (2024b) introduces a principled construction of efficient variants from the non-parametric perspective, including linear, top-K, and random feature modern Hopfield models. This study aims to refine this research direction towards efficient models. We believe that this study is critical in guiding future research toward a Hopfield-driven design paradigm, especially for large-scale models.

Fine-Grained Complexity. Much of fine-grained complexity theory relies on hypotheses concerning the time complexity of three problems: Conjunctive Normal Form Satisfiability (CNFSAT), All-Pairs Shortest Paths (APSP), and 3-SUM (Williams, 2018). Impagliazzo and Paturi (2001) introduce the Strong Exponential Time Hypothesis (SETH) to address the complexity of CNF-SAT. SETH is a stronger form of the $\mathtt{P} \neq \mathtt{NP}$ conjecture, suggesting that our current best $\mathtt{SAT}$ algorithms are optimal. It states as follows:

Hypothesis 1 (SETH). For every $\epsilon > 0$, there is a positive integer $k \geq 3$ such that k-$\mathtt{SAT}$ on formulas with n variables cannot be solved in $\mathcal{O}(2^{(1-\epsilon)n})$ time, even by a randomized algorithm.

SETH is a popular conjecture for proving fine-grained lower bounds for a wide variety of algorithmic problems, such as k-Hitting Set and k-NAE-SAT (Cygan et al., 2016). See Williams (2018) for a comprehensive review. Along this line, we utilize the fine-grained reduction under SETH to analyze the computational limits. In previous fine-grained reduction works, Backurs et al. (2017) analyze the computational complexity for multiple Empirical Risk Minimization problems, such as kernel SVMs and kernel ridge. Alman et al. (2020) study the applicability of efficient spectral graph theory on geometric graphs under SETH. Aggarwal and Alman (2022) focus on the computational limits of Batch Gaussian Kernel Density Estimation problems. Alman et al. (2023) utilize the weight-data correlation in a tree data structure for fast neural network training. Alman and Song (2023; 2024b) extend the previous work to transformer attention and introduce a tensor generalization. Compared to existing works, this work is, to the best of our knowledge, the first analysis of computational limits for modern Hopfield (associative memory) models (Hu et al., 2024a;b; Wu et al., 2024a;b; Hu et al., 2023; Ramsauer et al., 2021). In addition, it offers a more general characterization, encompassing computational analyses of self-attention (Alman and Song, 2024b; 2023) and cross-attention as special cases.

Notations. We denote (column) vectors by lower case bold letters, and matrices by upper case bold letters. We write $\langle \mathbf{a}, \mathbf{b} \rangle := \mathbf{a}^\mathsf{T}\mathbf{b}$ as the inner product for vectors $\mathbf{a}, \mathbf{b}$. Let $\mathbf{a}[i]$ denotes the i-th component of vector $\mathbf{a}$. The index set $\{1, \cdots, I\}$ is denoted by $[I]$, where $I \in \mathbb{N}_+$. Let $\|\mathbf{A}\|_{\max} := \max_{i,j} |\mathbf{A}_{ij}|$ for any matrix $\mathbf{A}$. We denote the memory patterns by $\boldsymbol{\xi} \in \mathbb{R}^d$ and the query pattern by $\mathbf{x} \in \mathbb{R}^d$, and $\boldsymbol{\Xi} := [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_M] \in \mathbb{R}^{d \times M}$ as shorthand for stored memory patterns $\{\boldsymbol{\xi}_\mu\}_{\mu \in [M]}$. We denote $\{\underline{\boldsymbol{\tau}}_1, \cdots, \underline{\boldsymbol{\tau}}_d\} \subset \mathbb{R}^{1 \times n}$ for each row in the matrix $\mathbf{Z} \in \mathbb{R}^{d \times n}$.

2. Computational Limits

In this section, we characterize the computational limits of all possible efficient variants of modern Hopfield models, i.e. $\mathtt{AHop}$, via fine-grained reduction. Our primary technique involves casting the $\mathtt{AHop}$ problem (Problem 1) as a subroutine in the Approximate Nearest Neighbor Search Problem and deducing the hardness through reduction.

2.1. Background: Approximate Nearest Neighbor Search Problem

Approximate Nearest Neighbor Search ($\mathtt{ANNS}$) problem (Indyk and Motwani, 1998; Arya et al., 1998; Muja and Lowe, 2014; Li et al., 2019) shares the same objective with the $\mathtt{AHop}$ problem of identifying a pattern closely resembling a query pattern as a memory retrieval process. Furthermore, the $\mathtt{ANNS}$ problem, which is particularly useful in high-dimensional spaces, seeks an approximate nearest neighbor within acceptable bounds to avoid the prohibitive computational costs of finding the exact nearest neighbor (Indyk and Motwani, 1998; Muja and Lowe, 2014). In this work, we observe that $\mathtt{ANNS}$ aligns with the goal of memory retrieval to efficiently find and recall the most relevant memory pattern in response to a specific input query. In our context, this translates to approximating the largest entry of $\mathrm{Softmax}(\boldsymbol{\Xi}^\mathsf{T}\mathbf{x})$ in (1.2) for each query $\mathbf{x}$, while maintaining a bounded error.

In $\mathtt{ANNS}$, one is given as input n vectors of dimension d, and an error parameter $\delta > 0$, and the goal is to find a pair of vectors whose distance is at most $(1+\delta)$ times the *minimum* distance between any pair of the vectors. The straightforward algorithm for ANNS runs in quadratic time, and it is known that it is impossible to solve ANNS in truly sub-quadratic time assuming SETH (Rubinstein, 2018).

To be concrete, we state the ANNS problem considered in this work as follows.

Definition 2.1 (Approximate Nearest Neighbor Search ANNS)**.** Given $\delta > 0$, $(1+\delta)$-ANNS for sets $A, B \subset \{0,1\}^d$, with $|A| = |B| = n$ requires finding $\mathbf{a}^* \in A, \mathbf{b}^* \in B$ such that:

$$\|\mathbf{a}^* - \mathbf{b}^*\|_2^2 \leq (1+\delta) \min_{\mathbf{a}\in A, \mathbf{b}\in B} \|\mathbf{a} - \mathbf{b}\|_2^2. \tag{2.1}$$

Next, we present the hardness results from Rubinstein (2018) as an auxiliary lemma for later use. Specifically, Rubinstein (2018) show that no sub-quadratic-time algorithms exist for the ANNS.

Lemma 2.1 (Hardness for ANNS, Theorem 4.1 of (Rubinstein, 2018))**.** Assuming Hypothesis 1, for every $q > 0$, there exist $\delta \in (0, 0.1)$ and $C > 0$ such that $(1+\delta)$-ANNS with dimension $d = C \log n$ requires $\Omega(n^{2-q})$ time.

2.2. Fine-Grained Reduction for AHop

To study the computational limits, our proof strategy involves connecting AHop to the hardness of ANNS (see Lemma 2.1) through a fine-grained reduction. We do this by introducing a decision problem Gap-ANNS as a $(1+\delta)$-gap reduction (Demaine, 2014) of the ANNS optimization problem (2.1), making the analysis more tractable while maintaining the same level of hardness. To be more precise, if we prove AHop is a reduction of Gap-ANNS, we also prove AHop is a $(1+\delta)$-gap reduction of ANNS. We start with Gap-ANNS in below.

Definition 2.2 (Gap Approximate Nearest Neighbor Search Gap-ANNS(d, n, t, δ))**.** Given two sets of n input vectors $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subset \{0,1\}^d$ and $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \{0,1\}^d$, the Gap-ANNS$(d, n, t, \delta)$ problem requires, for each $i \in [n]$, distinguish between the following two cases:

- **Case 1:** There exists at least one pair $(\mathbf{a}_i, \mathbf{b}_j) \in A \times B$ such that $\|\mathbf{a}_i - \mathbf{b}_j\|_2^2 < t$.
- **Case 2:** For all $\mathbf{b}_j \in B$, it holds that $\|\mathbf{a}_i - \mathbf{b}_j\|_2^2 \geq (1+\delta)\cdot t$.

An algorithm for Gap-ANNS(d, n, t, δ) with $\log(nd)$ time can binary search the answer of ANNS (Williams, 2018).

Then, we show that AHop serves as a subroutine within Gap-ANNS, thereby establishing a connection between the computational complexities of both problems.

Theorem 2.1 (Reduction from ANNS to AHop)**.** Consider Gap-ANNS with two sets of n input vectors, for every $q > 0$, for any chosen constants $C, C_0 > 0$, there exist $\delta \in (0, 0.1)$ and constants $C_\alpha, C_\beta > 0$ such that: Gap-ANNS$(d = C\log n, n, t = C_0 \log n, \delta)$ requires $\mathcal{O}(T + n^{2-q})$ time if AHop$(2d, M = 2n, L = 2n, \beta = 1/2d, B = C_\beta\sqrt{\log n}, \delta_H = n^{-C_\alpha})$ requires time T.

Proof Sketch. To solve Gap-ANNS, we employ different approaches for two scenarios, either through

- **Scenario 1:** a brute-force approach, or
- **Scenario 2:** reducing Gap-ANNS to an AHop problem, and translating AHop's solution to Gap-ANNS's solution (i.e. distinguish the 2 cases in Definition 2.2).

The proof of **Scenario 1** employs a brute-force algorithm for Gap-ANNS. This algorithm iterates over vectors within a Hamming distance of t from each input vector and checks for a match in the target set. It results in a manageable time complexity $\mathcal{O}(n^{2-q})$.

The proof for **Scenario 2** adopts a complex strategy. Initially, an AHop instance is formulated to encompass the Gap-ANNS problem. This formulation involves selecting specific parameters to ensure that resolving the AHop problem concurrently addresses the Gap-ANNS challenge. Next, we introduce $\widetilde{t}$, a threshold exceeding the AHop algorithm's error bound, to effectively bridge the conditions of the Gap-ANNS problem with the compound inequality derived from AHop. Finally, by considering an illustrative set of input vectors under the premise of a uniform distribution, the method for resolving the Gap-ANNS is elucidated with the established value of $\widetilde{t}$. This approach simplifies the decision-making process in solving Gap-ANNS. □

Main Proof. Here is the main proof of Theorem 2.1.

Proof. Let $\{\mathbf{a}_1, \cdots, \mathbf{a}_n, \mathbf{b}_1, \cdots, \mathbf{b}_n\} \subseteq \{0,1\}^d$ denote the input vectors of Gap-ANNS$(d = C\log n, n, t, \delta)$. For any given c satisfying

$$\begin{cases} c\,(\log C + 1) \leq 1 - q \\ 0 < c \leq \frac{1}{2}C, \end{cases} \tag{2.2}$$

we categorize two scenarios based on whether $t < c\log n$.

Scenario 1: $t < c\log n$.

The brute-force algorithm is described below:

1. For each $i \in [n]$, iterate over all vectors $\mathbf{b}' \in \{0,1\}^d$ which have Hamming distance at most t from $\mathbf{a}_i$.
2. Check whether $\mathbf{b}' \in \{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$.

Since $t < \frac{1}{2}C\log n < d$, there are $\binom{d}{t}$ choices for the vector $\mathbf{b}'$, so the algorithm takes $\mathcal{O}(n\cdot\binom{d}{t})$ time. We know:

$$n\cdot\binom{d}{t} \le n\cdot\binom{C\log n}{c\log n} \le \left(e\frac{C}{c}\right)^{c\log n} \le n^{1+c\log(Ce)}.$$

Therefore, if we choose constant c satisfying (2.2), the algorithm requires $\mathcal{O}(n^{2-q})$ time.

Scenario 2: $t \ge c\log n$.

Scenario 2 - Part 1. This part shows the associated `AHop` problem. Our objective is to construct an instance of the `AHop` problem in such a way that solving it also addresses the `Gap-ANNS` problem. To this end, we configure the `AHop`$(\widetilde{d},\widetilde{n},\widetilde{n},\beta,B,\delta_H)$ problem by selecting a specific set of parameters:

$$\widetilde{d} := 2d, \quad \widetilde{n} := 2n, \quad \beta := 1/\widetilde{d},$$

$$C_\beta > 2\sqrt{C/(C_0\delta)}, \quad C_\alpha > \frac{C_\beta^2}{4}(3+C_0/C)+1, \tag{2.3}$$

$$B := C_\beta\sqrt{\log n}, \quad \delta_H := n^{-C_\alpha}, \tag{2.4}$$

Note that δ_H is dependent on but not equal to δ.

We parametrize `AHop`'s input, Ξ and $\mathbf{X}$, with `Gap-ANNS` input $\{\mathbf{a}_1,\cdots,\mathbf{a}_n,\mathbf{b}_1,\cdots,\mathbf{b}_n\}$:

$$\Xi := B\cdot\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n & \mathbf{0}_d & \mathbf{0}_d & \cdots & \mathbf{0}_d\\ \mathbf{1}_d & \mathbf{1}_d & \cdots & \mathbf{1}_d & \mathbf{1}_d & \mathbf{1}_d & \cdots & \mathbf{1}_d\end{bmatrix} \in \mathbb{R}^{\widetilde{d}\times\widetilde{n}},$$

$$\mathbf{X} := B\cdot\begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_n & \mathbf{0}_d & \mathbf{0}_d & \cdots & \mathbf{0}_d\\ \mathbf{0}_d & \mathbf{0}_d & \cdots & \mathbf{0}_d & \mathbf{1}_d & \mathbf{1}_d & \cdots & \mathbf{1}_d\end{bmatrix} \in \mathbb{R}^{\widetilde{d}\times\widetilde{n}}.$$

By construction, we have $\|\Xi\|_{\max} \le B$ and $\|\mathbf{X}\|_{\max} \le B$. This follows that:

$$\left\|\beta\Xi^\top\mathbf{X}\right\|_{\max} \le \beta B^2\widetilde{d} = B^2.$$

Consider the matrix $\mathbf{A} := \exp\{\beta\Xi^\top\mathbf{X}\} \in \mathbb{R}^{\widetilde{n}\times\widetilde{n}}$:

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\\ \mathbf{A}_3 & \mathbf{A}_4\end{bmatrix}, \tag{2.5}$$

where $\mathbf{A}_1,\mathbf{A}_2,\mathbf{A}_3,\mathbf{A}_4 \in \mathbb{R}^{n\times n}$:

$$\begin{aligned}\mathbf{A}_1 &:= \left[\exp\{\beta B^2\langle\mathbf{a}_i,\mathbf{b}_j\rangle\}\right]_{i\in[1,n],j\in[1,n]},\\ \mathbf{A}_2 &:= \left[\exp\{B^2\}\right]_{i\in[1,n],j\in[n+1,2n]},\\ \mathbf{A}_3 &:= [0]_{i\in[n+1,2n],j\in[1,n]},\\ \mathbf{A}_4 &:= \left[\exp\{B^2\}\right]_{i\in[n+1,2n],j\in[n+1,2n]}.\end{aligned}$$

We provide the explicit form of (2.5) in (B.1).

For each $(i,j)\in[n]\times[n]$, it holds

$$\begin{aligned}\mathbf{A}_{i,j} &= \exp\{\beta B^2\langle\mathbf{a}_i,\mathbf{b}_j\rangle\}\\ &\le \exp\left\{\beta B^2\widetilde{d}\|\mathbf{a}_i\|_{\max}\|\mathbf{b}_j\|_{\max}\right\} \le \exp\{B^2\}.\end{aligned}$$

Thus,

$$0 \le \mathbf{A}_{i,j} \le \exp\{B^2\}.$$

Since $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_{\widetilde{n}})$, for each $i\in[\widetilde{n}]$, we get

$$n\exp\{B^2\} \le \mathbf{D}_{i,i} \le 2n\exp\{B^2\}. \tag{2.6}$$

Scenario 2 - Part 2. This part shows the `Gap-ANNS` is a part of the associated `AHop` problem. Given input matrices $\mathbf{D}\in\mathbb{R}^{\widetilde{n}\times\widetilde{n}}$, $\mathbf{A}\in\mathbb{R}^{\widetilde{n}\times\widetilde{n}}$, if we have an algorithm `AHop`$(\widetilde{d},\widetilde{n},\widetilde{n},\beta,B,\delta_H)$ such that its output $\widetilde{\mathbf{Z}}$ satisfies

$$\left\|\widetilde{\mathbf{Z}} - \Xi\mathbf{D}^{-1}\mathbf{A}\right\|_{\max} \le \delta_H. \tag{2.7}$$

To connect (2.7) to `Gap-ANNS`, we define $\widetilde{t}$ as

$$\widetilde{t} := \frac{1}{3}\frac{\exp\{\frac{1}{4}B^2(1-t/d)\}}{2n\exp\{B^2\}}.$$

It follows that

$$\begin{aligned}\widetilde{t} &= \frac{1}{6n}\exp\left\{-\frac{3}{4}B^2 - \frac{1}{4}B^2 t/d\right\}\\ &= \frac{1}{6n}\exp\left\{-\frac{3}{4}B^2 - \frac{1}{4}B^2 C_0/C\right\}\\ &= \frac{1}{6}\exp\left\{-\frac{3}{4}C_\beta^2\log n - \frac{1}{4}\frac{C_0}{C}C_\beta^2\log n - \log n\right\} &&\text{(By (2.4))}\\ &= \frac{1}{6}n^{-\frac{3}{4}C_\beta^2-\frac{1}{4}\frac{C_0}{C}C_\beta^2-1} \ge n^{-C_\alpha} = \delta_H.\end{aligned}$$

Since $\widetilde{t}\ge\delta_H$, the last row vector of $\widetilde{\mathbf{Z}}$, i.e $\underline{\widetilde{\mathbf{z}}}_{\widetilde{d}}\in\mathbb{R}^{1\times\widetilde{n}}$ for all $j\in[\widetilde{n}]$, satisfying

$$\left|\underline{\widetilde{\mathbf{z}}}_{\widetilde{d}}[j] - \left(\underline{\boldsymbol{\xi}}_{\widetilde{d}}\mathbf{D}^{-1}\mathbf{A}\right)[j]\right| \le \widetilde{t}, \tag{2.8}$$

where $\underline{\boldsymbol{\xi}}_{\widetilde{d}} = \mathbf{1}_{\widetilde{n}}^\top$ is the last row of Ξ.

Scenario 2 - Part 3. This part shows how to distinguish the 2 cases in the `Gap-ANNS` with $\underline{\widetilde{\mathbf{z}}}_{\widetilde{d}}$ constructed in the previous `AHop`$(\widetilde{d},\widetilde{n},\widetilde{n},\beta,B,\delta_H)$ problem.

For the sake of convenience, we assume each input vector has an equal probability of being either 0 or 1, that is,

$$\begin{cases}\|\mathbf{a}_i\|_2^2 = d/2, & \forall i\in[n],\\ \|\mathbf{b}_j\|_2^2 = d/2, & \forall j\in[n].\end{cases}$$

Hence, for each $(i,j) \in [n] \times [n]$,

$$\beta B^2 \langle \mathbf{a}_i, \mathbf{b}_j \rangle = \frac{B^2}{4d}(\|\mathbf{a}_i\|_2^2 + \|\mathbf{b}_j\|_2^2 - \|\mathbf{a}_i - \mathbf{b}_j\|_2^2) = \frac{B^2}{4d}\left(d - \|\mathbf{a}_i - \mathbf{b}_j\|_2^2\right). \tag{2.9}$$

Our goal of solving $\texttt{Gap-ANNS}(d,n,t,\delta)$ is to determine, for each $i \in [n]$, whether there is a $j \in [n]$ such that $\|\mathbf{a}_i - \mathbf{b}_j\|_2^2 \le t$, or whether $\|\mathbf{a}_i - \mathbf{b}_j\|_2^2 > (1+\delta)t$ for all $j \in [n]$.

Case 1: If there exists an $(i,j) \in [n] \times [n]$ such that $\|\mathbf{a}_i - \mathbf{b}_j\|_2^2 \le t$, then

$$\beta B^2 \langle \mathbf{a}_i, \mathbf{b}_j \rangle \ge \frac{1}{4} B^2 (1 - t/d). \qquad \text{(By (2.9))}$$

In this case,

$$\begin{aligned}
\widetilde{\mathbf{z}}_{\widetilde{d}}[j] &\ge \sum_{\iota}^{n} \mathbf{D}_{\iota,\iota}^{-1} \mathbf{A}_{\iota,j} - \widetilde{t} && \text{(By (2.8))}\\
&\ge \mathbf{D}_{i,i}^{-1} \exp\{\beta B^2 \langle \mathbf{a}_i, \mathbf{b}_j \rangle\} - \widetilde{t} \\
&\ge \frac{\exp\{\frac{1}{4}B^2(1-t/d)\}}{2n\exp\{B^2\}} - \widetilde{t} && \text{(By (2.6))}\\
&= 2\widetilde{t}.
\end{aligned}$$

Case 2: If $\|\mathbf{a}_i - \mathbf{b}_j\|_2^2 > (1+\delta)t$ for all $(i,j) \in [n] \times [n]$, then

$$\beta B^2 \langle \mathbf{a}_i, \mathbf{b}_j \rangle < \frac{1}{4} B^2 (1 - (1+\delta)t/d). \qquad \text{(By (2.9))}$$

In this case,

$$\begin{aligned}
\widetilde{\mathbf{z}}_{\widetilde{d}}[j] &\le \sum_{\iota}^{n} \mathbf{D}_{\iota,\iota}^{-1} \mathbf{A}_{\iota,j} + \widetilde{t} && \text{(By (2.8))}\\
&= \sum_{\iota}^{n} \mathbf{D}_{\iota,\iota}^{-1} \exp\{\beta B^2 \langle \mathbf{a}_\iota, \mathbf{b}_j \rangle\} + \widetilde{t} \\
&< \frac{n\exp\{\frac{1}{4}B^2(1-(1+\delta)t/d)\}}{n\exp\{B^2\}} + \widetilde{t} && \text{(By (2.6))}\\
&= \frac{\exp\{\frac{1}{4}B^2(1-t/d)\}}{2n\exp\{B^2\}} \frac{2n}{\exp\{\frac{\delta}{4}B^2 t/d\}} + \widetilde{t} \\
&= 3\widetilde{t} \cdot \frac{2n}{\exp\left\{\frac{\delta}{4}C_\beta^2 \log n C_0/C\right\}} + \widetilde{t} && \text{(By (2.3))}\\
&< 2\widetilde{t}.
\end{aligned}$$

Therefore, by determining whether $\widetilde{\mathbf{z}}_{\widetilde{d}}[j] \ge 2\widetilde{t}$, we distinguish the two cases, or solve the $\texttt{Gap-Ann}(n,d,t,\delta)$. Furthermore, the entire algorithm take T time, the same as the time required for $\texttt{AHop}(\widetilde{d}, \widetilde{n}, \widetilde{n}, \beta, B, \delta_H)$. ☐

Corollary 2.1.1. Assuming Hypothesis 1, for every $q > 0$, for any chosen $C, C_0 > 0$, there exist $\delta \in (0, 0.1)$ and $C_\alpha, C_\beta > 0$ satisfying (2.3) such that $\texttt{AHop}(2d, M = 2n, L = 2n, \beta = 1/2d, B = C_\beta\sqrt{\log n}, \delta_H = n^{-C_\alpha})$ requires $\Omega(n^{2-q})$ time.

Proof. By Lemma 2.1, suppose $\delta \in (0, 0.1)$, $(1+\delta)$-$\texttt{ANNS}$ with dimension $d = C\log n$ requires $\Omega(n^{2-q})$ time. By Theorem 2.1, $\texttt{Gap-ANNS}$ requires $\mathcal{O}(T + n^{2-q})$ time with T being the computation time of $\texttt{AHop}(d, M, L, \beta, B, \delta_H)$. For $\texttt{Gap-ANNS}$ to have the same precision δ as $(1+\delta)$-$\texttt{ANNS}$, we need $\mathcal{O}(T + n^{2-q}) = \Omega(n^{2-q})$. Consequently, $\texttt{AHop}(d, M, L, \beta, B, \delta_H)$ requires $T = \Omega(n^{2-q})$ time. This completes the proof. ☐

Interestingly, Corollary 2.1.1 characterizes a phase transition behavior in $\texttt{AHop}$ problems assuming Hypothesis 1. To extend the applicability of this corollary beyond the specific case where $M = L = n$, we introduce $\tau := \max\{M, L\}$ to capture the larger dimension. That is, regardless of whether M or L is larger, τ ensures that the hardness result considers the worst-case scenario (i.e. extending the shorter one). To sum up, we establish a criterion $B^\star = \Theta(\sqrt{\log \tau})$ for $\|\boldsymbol{\Xi}\|_{\max}$ and $\|\mathbf{X}\|_{\max}$ such that, only below which, solving $\texttt{AHop}$ in $\tau^{2-\Omega(1)}$ (sub-quadratic) time is possible.

3. An Almost Linear Modern Hopfield Model

To showcase our theory, this section presents an example of an almost linear-time modern Hopfield model using low-degree polynomial approximation. We show its almost linear lower bound on computational time in Section 3.2 and its upper bound on memory retrieval error in the same section. Additionally, we show that this model possesses a marginally smaller, yet still exponential-in-d memory capacity in Section 3.3, compared to standard modern Hopfield associative memory models (Wu et al., 2024b; Hu et al., 2023; Ramsauer et al., 2021).

3.1. Background: Polynomial Method for Low-Rank Approximation

Consider a matrix $\mathbf{A} \in \mathbb{R}^{p\times q}$, and a function $f : \mathbb{R} \to \mathbb{R}$. We define $f(\mathbf{A}) : \mathbb{R}^{p\times q} \to \mathbb{R}^{p\times q}$ as the matrix obtained by applying f entry-wise to $\mathbf{A}$. The polynomial method aims to find a low-rank approximation for $f(\mathbf{A})$. Under this method, if $\mathbf{A}$ possesses a low rank, and if function f can be well-approximated by a low-degree polynomial, then the matrix $f(\mathbf{A})$ can be approximated by a low-rank matrix. Furthermore, this low-rank approximation can be efficiently computed in terms of its low-rank decomposition.

Aggarwal and Alman (2022) provide the bounds on the degrees of the polynomial required for low-rank approximation of $f(\mathbf{A})$, particularly when f is the exponential function. Leveraging these results, we construct a low-rank approximation for $\mathrm{Softmax}(\beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X})$ in (1.1), satisfying the following definition:

Definition 3.1 ((δ_A, r) Low-Rank Approximation)**.** Let $r \in \mathbb{N}_+ \geq 1$ and $\delta_A \in (0, 0.1)$. For a given $\mathbf{A} \in \mathbb{R}^{p \times q}$, we say $\widetilde{\mathbf{A}} \in \mathbb{R}^{p \times q}$ is an (δ_A, r)-approximation of $\mathbf{A}$ if

- $\widetilde{\mathbf{A}} = \mathbf{U}\mathbf{V}^\mathsf{T}$ with $\mathbf{U} \in \mathbb{R}^{p \times r}$ and $\mathbf{V} \in \mathbb{R}^{q \times r}$, and
- $\left| \widetilde{\mathbf{A}}_{ij} - \mathbf{A}_{ij} \right| \leq \delta_A \cdot \mathbf{A}_{ij}$ for each $i \in [p]$ and $j \in [q]$.

3.2. Low-Rank Matrix Approximation for `AHop`

This section includes our linear time result for `AHop` via low-rank approximation. Let $\|\mathbf{X}\|_{\max} \leq B$ and $\|\boldsymbol{\Xi}\|_{\max} \leq B$. Let $\mathtt{T}_{\mathrm{mat}}(a, b, c)$ denote the time required for multiplication between an $\mathbb{R}^{a \times b}$ matrix and an $\mathbb{R}^{b \times c}$ matrix. In fact, $\mathtt{T}_{\mathrm{mat}}(a, b, c) \leq \mathcal{O}(abc)$.

We compute $\widetilde{\mathbf{A}}$ as a (δ_A, r)-approximation of $\mathbf{A}$:

Lemma 3.1. Suppose $B > 1$ and matrices $\boldsymbol{\Xi} \in \mathbb{R}^{d \times M}$, $\mathbf{X} \in \mathbb{R}^{d \times L}$ have $\|\mathbf{X}\|_{\max} \leq B$ and $\|\boldsymbol{\Xi}\|_{\max} \leq B$. Given $\mathbf{A} = \exp\{\beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X}\} \in \mathbb{R}^{M \times L}$, for $\delta_A \in (0, 0.1)$, there is a positive integer g upper bounded by

$$g = \mathcal{O}\left(\max \left\{ B^2 \beta d, \frac{\log(1/\delta_A)}{\log\left[1/(B^2 \beta d) \cdot \log(1/\delta_A)\right]} \right\} \right),$$

and a $r \in \mathbb{N}_+$ upper bounded by $r \leq \binom{2(g+d)}{2g}$ such that: There is a matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{M \times L}$ that is an (δ_A, r)-approximation of $\mathbf{A} \in \mathbb{R}^{M \times L}$. Furthermore, the matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ defining $\widetilde{\mathbf{A}}$ can be computed in $\mathcal{O}(\max\{M, L\} \cdot rg)$ time.

Proof. For each $(i, j) \in [M] \times [L]$, we have

$$\left| (\boldsymbol{\Xi}^\mathsf{T} \mathbf{X})_{i,j} \right| = \left| \sum_{l=1}^{d} \boldsymbol{\Xi}_{l,i} \mathbf{X}_{l,j} \right| \leq \|\boldsymbol{\Xi}\|_{\max} \|\mathbf{X}\|_{\max} d \leq B^2 d.$$

Thus, the entries of the exp in $\mathbf{A}$ have upper bound:

$$\left\| \beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X} \right\|_{\max} \leq B^2 \beta d.$$

Applying Lemma A.2 with bound $B^2 \beta d$, there is a polynomial function $P(x)$ of degree g such that:

$$\sup_{\boldsymbol{\Xi}, \mathbf{X}} \left| P((\beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X})_{ij}) - \mathbf{A}_{ij} \right| < \delta_A \cdot \mathbf{A}_{ij}.$$

Applying Lemma A.3 with $\boldsymbol{\Xi}$, $\mathbf{X}$, there exists an algorithm constructing $\mathbf{U}_1, \mathbf{U}_2$ in $\mathcal{O}(\max\{M, L\} \cdot rg)$ time such that $P(\beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X}) = \mathbf{U}_1 \mathbf{U}_2^\mathsf{T}$.

Therefore, by Definition 3.1, $\widetilde{\mathbf{A}} := P(\beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X})$ is an (δ_A, r)-approximation of $\mathbf{A}$. □

Prior to solving `AHop`, we compute the approximation error bound for $\widetilde{\mathbf{Z}}$ by utilizing a low-rank approximation (Lemma 3.1) applied to Problem 1.

Lemma 3.2 (Approximation Error)**.** Let $\delta_A \in (0, 0.1)$, $\beta > 0$, $B > 0$, $\|\mathbf{X}\|_{\max} \leq B$, and $\|\boldsymbol{\Xi}\|_{\max} \leq B$. Let $\mathbf{A} = \exp\{\beta \boldsymbol{\Xi}^\mathsf{T} \mathbf{X}\} \in \mathbb{R}^{M \times L}$, and let $\widetilde{\mathbf{A}} \in \mathbb{R}^{M \times L}$ such that, for each $(l, j) \in [M] \times [L]$,

$$\left| \widetilde{\mathbf{A}}_{l,j} - \mathbf{A}_{l,j} \right| \leq \delta_A \cdot \mathbf{A}_{l,j}. \tag{3.1}$$

Let $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_M) \in \mathbb{R}^{M \times M}$ and $\widetilde{\mathbf{D}} = \mathrm{diag}(\widetilde{\mathbf{A}}\mathbf{1}_M) \in \mathbb{R}^{M \times M}$, for each $l \in [M]$, we have

$$\left| \widetilde{\mathbf{D}}_{l,l} - \mathbf{D}_{l,l} \right| \leq \delta_A \cdot \mathbf{D}_{l,l}. \tag{3.2}$$

Hence, we have

$$\left\| \widetilde{\mathbf{Z}} - \mathbf{Z} \right\|_{\max} = \left\| \boldsymbol{\Xi} \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbf{A}} - \boldsymbol{\Xi} \mathbf{D}^{-1} \mathbf{A} \right\|_{\max} \leq 2MB\delta_A. \tag{3.3}$$

Proof. To see (3.2), we observe

$$\begin{aligned}
\left| \widetilde{\mathbf{D}}_{l,l} - \mathbf{D}_{l,l} \right| &= \left| \sum_{j=1}^{n} \widetilde{\mathbf{A}}_{l,j} - \sum_{j=1}^{n} \mathbf{A}_{l,j} \right| \\
&\leq \sum_{j=1}^{n} \left| \widetilde{\mathbf{A}}_{l,j} - \mathbf{A}_{l,j} \right| \\
&\leq \sum_{j=1}^{n} \delta_A \cdot \mathbf{A}_{l,j} = \delta_A \cdot \mathbf{D}_{l,l}. \quad \text{(By (3.1))}
\end{aligned}$$

By triangle inequality, we have

$$\begin{aligned}
&\left\| \boldsymbol{\Xi} \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbf{A}} - \boldsymbol{\Xi} \mathbf{D}^{-1} \mathbf{A} \right\|_{\max} \\
\leq &\underbrace{\left\| \boldsymbol{\Xi} \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbf{A}} - \boldsymbol{\Xi} \mathbf{D}^{-1} \widetilde{\mathbf{A}} \right\|_{\max}}_{\text{(I)}} + \underbrace{\left\| \boldsymbol{\Xi} \mathbf{D}^{-1} \widetilde{\mathbf{A}} - \boldsymbol{\Xi} \mathbf{D}^{-1} \mathbf{A} \right\|_{\max}}_{\text{(II)}}.
\end{aligned}$$

Consider the (I) term; for each $(i, j) \in [d] \times [L]$, we have

$$\begin{aligned}
&\left| \left(\boldsymbol{\Xi} \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbf{A}} - \boldsymbol{\Xi} \mathbf{D}^{-1} \widetilde{\mathbf{A}} \right)_{i,j} \right| \\
= &\left| \sum_{l=1}^{M} \boldsymbol{\Xi}_{i,l} \left(\widetilde{\mathbf{D}}_{l,l}^{-1} - \mathbf{D}_{l,l}^{-1} \right) \widetilde{\mathbf{A}}_{l,j} \right| \\
\leq &\sum_{l=1}^{M} \left| \left(\widetilde{\mathbf{D}}_{l,l}^{-1} - \mathbf{D}_{l,l}^{-1} \right) \widetilde{\mathbf{A}}_{l,j} \right| \cdot \|\boldsymbol{\Xi}\|_{\max} \\
= &\sum_{l=1}^{M} \left| \frac{\mathbf{D}_{l,l} - \widetilde{\mathbf{D}}_{l,l}}{\mathbf{D}_{l,l} \widetilde{\mathbf{D}}_{l,l}} \widetilde{\mathbf{A}}_{l,j} \right| \cdot \|\boldsymbol{\Xi}\|_{\max} \\
\leq &\delta_A B \sum_{l=1}^{M} \widetilde{\mathbf{D}}_{l,l}^{-1} \widetilde{\mathbf{A}}_{l,j} \quad \text{(By } \|\boldsymbol{\Xi}\|_{\max} \leq B \text{ and (3.2))} \\
\leq &\delta_A B \cdot M.
\end{aligned}$$

Algorithm 1 The algorithm to solve AHop

Input: matrices $\Xi \in \mathbb{R}^{d\times M}, \mathbf{X} \in \mathbb{R}^{d\times L}$, with β, d, M, L, B, and error margin δ_A. Let $\tau := \max\{M, L\}$.

1: $g \leftarrow \mathcal{O}\left(\max\left\{B^2\beta d, \frac{\log(1/\delta_A)}{\log[1/(B^2\beta d)\cdot\log(1/\delta_A)]}\right\}\right)$ by Lemma 3.1

2: $r \leftarrow \binom{2(g+d)}{2g}$ by Lemma 3.1

3: Compute $\mathbf{U}_1 \in \mathbb{R}^{M\times r}, \mathbf{U}_2 \in \mathbb{R}^{L\times r}$ by Lemma A.3 // Time: $\mathcal{O}(\tau r g)$

4: Compute $\widetilde{\mathbf{D}}^{-1} = \mathrm{diag}\left(\mathbf{U}_1(\mathbf{U}_2^T\mathbf{1}_L)\right) \in \mathbb{R}^{M\times M}$ // Time: $\mathcal{O}(\tau r)$

5: $\widetilde{\mathbf{Z}} \leftarrow \Xi\widetilde{\mathbf{D}}^{-1}\mathbf{U}_1\mathbf{U}_2^\mathsf{T} \in \mathbb{R}^{d\times M}$ // Time: $\mathcal{O}(\tau r d)$

return $\widetilde{\mathbf{Z}}$

Consider the (II) term; for each $(i, j) \in [d] \times [L]$, we have

$$
\begin{aligned}
&\left|\left(\Xi\widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}} - \Xi\mathbf{D}^{-1}\mathbf{A}\right)_{i,j}\right| \\
=&\left|\sum_{l=1}^{M}\Xi_{i,l}\mathbf{D}_{l,l}^{-1}\left(\widetilde{\mathbf{A}}_{l,j} - \mathbf{A}_{l,j}\right)\right| \\
\leq&\sum_{l=1}^{M}\left|\mathbf{D}_{l,l}^{-1}\right|\cdot\left|\left(\widetilde{\mathbf{A}}_{l,j} - \mathbf{A}_{l,j}\right)\right|\cdot\|\Xi\|_{\max} \\
\leq&\,\delta_A B\sum_{l=1}^{M}\mathbf{D}_{l,l}^{-1}\mathbf{A}_{l,j} \qquad (\text{By } \|\Xi\|_{\max} \leq B \text{ and } (3.1)) \\
\leq&\,\delta_A B\cdot M.
\end{aligned}
$$

Combining (I) and (II), we obtain

$$
\left\|\Xi\widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}} - \Xi\mathbf{D}^{-1}\mathbf{A}\right\|_{\max} \leq 2MB\delta_A.
$$

This completes the proof of (3.3). □

Lemma 3.2 states that the controllable approximation error in Problem 1 takes the form of $\delta_H = 2MB\delta_A$ by low-rank approximation. Here M is the size of stored memory set Ξ, δ_A is the precision of low-rank approximation and B is the upper bound of $\|\mathbf{X}\|_{\max}$ and $\|\Xi\|_{\max}$.

Next, we show that AHop utilizing (δ_A, r)-approximation requires only almost linear computational time.

Theorem 3.1 (Almost Linear AHop, Algorithm 1)**.** Let $\tau := \max\{M, L\}$ and $\delta_H := 2MB\delta_A$. For $\beta > 0$, $d, M, L \in \mathbb{N}_+$, $\delta_A > 0$, $\|\mathbf{X}\|_{\max} \leq B$ and $\|\Xi\|_{\max} \leq B$ with $B \geq 1$, there are $g = \mathcal{O}\left(\max\left\{B^2\beta d, \frac{\log(1/\delta_A)}{\log[1/(B^2\beta d)\cdot\log(1/\delta_A)]}\right\}\right) \in \mathbb{N}_+$ and $r = \binom{2(g+d)}{2g} \in \mathbb{N}_+$ such that: There exists an Algorithm 1 that runs in $\mathcal{O}(\tau r g + \tau r d)$ time to solve $\mathtt{AHop}(d, M, L, \beta, B, \delta_H)$. Thus, under realistic settings where $d = \mathcal{O}(\log\tau), \beta = \Theta(1/d), \delta_H = MB/poly(\tau)$, if $B = o(\sqrt{\log\tau})$, Algorithm 1 requires time $\tau^{1+o(1)}$.

Proof. In Algorithm 1, step 3 requires $\mathcal{O}(\tau r g)$ time by Lemma 3.1; step 4 requires $\mathtt{T}_{\mathrm{mat}}(r, L, 1) + \mathtt{T}_{\mathrm{mat}}(M, r, 1) = \mathcal{O}(\tau r)$ time; step 5 requires $dM + \mathtt{T}_{\mathrm{mat}}(d, M, r) + \mathtt{T}_{\mathrm{mat}}(d, r, L) = \mathcal{O}(\tau r d)$ time. Thus, Algorithm 1 requires $\mathcal{O}(\tau r g + \tau r d)$ time.

If the parameters satisfy $d = \mathcal{O}(\log\tau), \beta = \Theta(1/d), B = o(\sqrt{\log\tau})$, and $\delta_A = 1/poly(\tau) = \tau^{-\mathcal{O}(1)}$, we have

$$
\begin{aligned}
g &= \mathcal{O}\left(\max\left\{B^2\beta d, \frac{\log(1/\delta_A)}{\log\left[1/(B^2\beta d)\cdot\log(1/\delta_A)\right]}\right\}\right) \\
&= \mathcal{O}\left(\max\left\{o(\log\tau), \frac{\log\tau}{\log(\log\tau)}\right\}\right) = o(\log\tau).
\end{aligned}
$$

We write g as $\log\tau/f$ with any $f = \omega(1)$, then

$$
\begin{aligned}
r &= \binom{2(d+g)}{2g} \leq \left(\frac{e(d+g)}{g}\right)^{2g} = 2^{\mathcal{O}(g\log((d+g)/g))} \\
&\leq 2^{\mathcal{O}(g\log(\log\tau/g))} = 2^{\mathcal{O}(\log\tau\log f/f)} \\
&< 2^{o(\log\tau)} < \tau^{o(1)}.
\end{aligned}
$$

We know $(\log\tau)^{\mathcal{O}(1)} \leq \tau^c$ for all $a, c > 0$ and $b > 1$, so

$$
\mathcal{O}(\tau^a(\log\tau)^b) \leq \tau^a\cdot\tau^{o(1)} = \tau^{a+o(1)},
$$

where $\tau^{a+o(1)}$ means $\tau^{a+o(1)}$ grows slightly larger than τ^a. Since $d, r, g = \mathcal{O}(\log\tau)$, there exists some constant K such that $d, r, g \leq K\log\tau$. Thus, Algorithm 1 requires time:

$$
\mathcal{O}(\tau r d + \tau r g) \leq \mathcal{O}\left(\tau(\log\tau)^2\right) \leq \tau^{1+o(1)}.
$$

This completes the proof. □

Theorem 3.1 provides a formal example of efficient computation Algorithm 1 for AHop using low-rank approximation (Lemma 3.1) within a controllable approximation error (Lemma 3.2). This corresponds to Corollary 2.1.1 when the efficient criterion holds. Specifically, to achieve efficient computation under realistic settings, we require $B = o(\sqrt{\log\tau})$, leading to almost linear running time $\tau^{1+o(1)}$.

3.3. Memory Retrieval Error Bound

Considering the standard modern Hopfield retrieval dynamics with length-L query sequences from (1.1):

$$
\mathbf{Z} = \Xi\,\mathrm{Softmax}\left(\beta\Xi^\mathsf{T}\mathbf{X}\right).
$$

Let $\widetilde{\mathbf{Z}} \in \mathbb{R}^{d \times L}$ be the output of the *efficient* memory retrieval dynamics by Algorithm 1 retrieving $\mathbf{X}^{\text{new}}$ from stored memory set $\boldsymbol{\Xi} \in \mathbb{R}^{d \times M}$ based on given query $\mathbf{X} \in \mathbb{R}^{d \times L}$.

To see how this approximate model stores and retrieves memory patterns, we first introduce the following definitions.

Definition 3.2. Given a function $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$. A generalized fixed point of $\mathcal{T}$ is a point $\mathbf{x} \in \mathbb{R}^d$ for which $\mathbf{x} \in \mathcal{T}(\mathbf{x})$.

Definition 3.3 (Memory Storage and Retrieval)**.** For each $\mu \in [M]$, let $R := \frac{1}{2} \operatorname{Min}_{\mu,\nu \in [M];\mu \neq \nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$ be the finite radius of each sphere $\mathcal{S}_\mu$ centered at memory pattern $\boldsymbol{\xi}_\mu$. We say $\boldsymbol{\xi}_\mu$ is *stored* if all $\mathbf{x} \in \mathcal{S}_\mu$ are generalized fixed points of $\mathcal{T}$, $\mathbf{x}^\star_\mu \in \mathcal{S}_\mu$, and $\mathcal{S}_\mu \cap \mathcal{S}_\nu = \emptyset$ for $\mu \neq \nu$. We say $\boldsymbol{\xi}_\mu$ is ϵ*-retrieved* by $\mathcal{T}$ with $\mathbf{x}$ for an error ϵ, if $\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \epsilon$.

Remark 3.1. A direct implication from Definition 3.3 is that the approximation error of `AHop` (see Equation (2.7)) must satisfy $\delta_H = 2MB\delta_A < R$ for successful memory retrieval (and storage).

Additionally, we recall the following definition regarding the separation between memory patterns.

Definition 3.4 (Separation of Patterns)**.** The separation of a memory pattern $\boldsymbol{\xi}_\mu$ from all other memory patterns $\boldsymbol{\Xi}$ is defined as its minimal inner product difference to any other patterns: $\Delta_\mu := \operatorname{Min}_{\nu,\nu \neq \mu} [\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu \rangle - \langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle]$.

Next, we present the retrieval error bound of $\widetilde{\mathbf{Z}}$.

Theorem 3.2 (Retrieval Error)**.** Let $\overline{\boldsymbol{\Xi}}$ be the ground truth memory sequence corresponding to $\mathbf{X}$. Suppose $\mathbf{x}_l \in S_\mu$ with some $\mu \in [M]$ for each $l \in [L]$, it holds

$$\left\|\widetilde{\mathbf{Z}} - \overline{\boldsymbol{\Xi}}\right\|_{\max} \leq 2B(M-1)e^{-\beta\left(\langle \boldsymbol{\xi}_\mu, \mathbf{x}\rangle - \operatorname{Max}_{\nu \in [M]} \langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle\right)} + 2MB\delta_A. \tag{3.4}$$

Proof. We first decompose the RHS of (3.4) as

$$\left\|\widetilde{\mathbf{Z}} - \overline{\boldsymbol{\Xi}}\right\|_{\max} = \Big\| \underbrace{\left(\widetilde{\mathbf{Z}} - \mathbf{Z}\right)}_{\text{Approximation Error}} + \underbrace{\left(\mathbf{Z} - \overline{\boldsymbol{\Xi}}\right)}_{\text{Retrieval Error}} \Big\|_{\max}. \tag{3.5}$$

Then, we bound the approximation error with Lemma 3.2 and bound the retrieval error with (Hu et al., 2023, eqn. 2.7). By triangle inequality, we complete the proof. $\square$

Remark 3.2. By definition of $\|\cdot\|_{\max}$, this bound also holds for retrieval based on single pattern $\mathbf{x}$.

Remark 3.3. Similar to standard results of modern Hopfield models (Wu et al., 2024a;b; Hu et al., 2023; Ramsauer et al., 2021), (3.4) indicates that with sufficiently large Δ_μ and sufficiently small approximation error, Algorithm 1 retrieves memory patterns in a single *iteration*. This allows this efficient modern Hopfield model to serve as a network layer with a single activation, enabling its integration into deep learning, similar to (Hu et al., 2024a; Xu et al., 2024; Schimunek et al., 2023; Hoover et al., 2023; Seidl et al., 2022; Fürst et al., 2022; Paischer et al., 2022).

Surprisingly, this model achieves almost linear time efficiency while maintaining the exponential memory capacity characteristic of modern Hopfield models.

Corollary 3.2.1 (Capacity Lower Bound, Informal)**.** Suppose all memory patterns are sampled from a sphere of radius m. This efficient modern Hopfield (approximate (1.1) with Algorithm 1) exhibits a *exponential-in-d* lower bound M on the number of patterns it can store and retrieve.

Proof Sketch. We first derive the necessary condition for a pattern to be stored and retrieved in the model, i.e., the well-separation condition. Next, we combine it with the separation analysis of random patterns (Hu et al., 2023). See Appendix B.3 for a formal version and a detailed proof. $\square$

Remark 3.4. While the capacity M is slightly smaller than those of (Wu et al., 2024b; Hu et al., 2023; Ramsauer et al., 2021), it still scales exponentially in pattern dimension d. Namely, `AHop` as per Algorithm 1 achieves almost linear computation time with only a marginal sacrifice in memory capacity.

4. Discussion and Conclusion

We apply the fine-grained reduction under the SETH hypothesis to study the computational limits of the retrieval dynamics of modern Hopfield associative memory models (Hu et al., 2024a;b; Wu et al., 2024a;b; Hu et al., 2023; Ramsauer et al., 2021). This work holds practical significance because of the robust link between transformer attention mechanisms and modern Hopfield models. We make a key observation by framing associative memory retrieval as an Approximate Nearest Neighbor Search (ANNS) problem, enabling the application of fine-grained reduction. This allows us to identify a phase transition behavior on the efficiency of all possible variants of modern Hopfield models (Corollary 2.1.1) by tuning the norm bound of queries $\mathbf{X}$ and memories $\boldsymbol{\Xi}$. In addition, we showcase our theory with an almost linear time variant of modern Hopfield models (Theorem 3.1). We show this efficient model inherits the defining characteristic of modern Hopfield models: exponential memory capacity (Corollary 3.2.1 and Theorem B.1).

Limitation. By the formal nature of this work, our results do not lead to practical implementations. However, we anticipate that our findings will offer valuable insights for future efficient Hopfield-centric and transformer-based foundation models and deep learning implementations.

Impact Statement

This theoretical work, as outlined in the introduction and related works, aims to elucidate the foundations of large Hopfield- and transformer-based foundation models and is not expected to have negative social impacts.

Acknowledgments

JH would like to thank Feng Ruan, Dino Feng and Andrew Chen for enlightening discussions on related topics, and the Red Maple Family for support. TL would like to thank Gamma Paradigm Research and NTU ABC-Labs for support. The authors would like to thank the anonymous reviewers and program chairs for constructive comments. JH is partially supported by the Walter P. Murphy Fellowship. HL is partially supported by NIH R01LM1372201, NSF CAREER1841569, DOE DE-AC02-07CH11359, DOE LAB 20-2261 and a NSF TRIPODS1740735. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

References

Amol Aggarwal and Josh Alman. Optimal-degree polynomial approximations for exponentials and gaussian kernel density estimation. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 9783959772419. doi: 10.4230/LIPIcs.CCC.2022.22. URL https://doi.org/10.4230/LIPIcs.CCC.2022.22.

Josh Alman and Zhao Song. Fast attention requires bounded entries. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=KOVWXcrFIK.

Josh Alman and Zhao Song. The fine-grained complexity of gradient computation for training large language models. *arXiv preprint arXiv:2402.04497*, 2024a. URL https://arxiv.org/abs/2402.04497.

Josh Alman and Zhao Song. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b. URL https://openreview.net/forum?id=v0zNCwwkaV.

Josh Alman, Timothy Chu, Aaron Schild, and Zhao Song. Algorithms and hardness for linear algebra on geometric graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 541–552. IEEE, 2020. URL https://arxiv.org/abs/2011.02466.

Josh Alman, Jiehao Liang, Zhao Song, Ruizhe Zhang, and Danyang Zhuo. Bypass exponential time preprocessing: Fast neural network training via weight-data correlation preprocessing. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=ZqSx5vXOgC.

Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

Andreas Auer, Martin Gauch, Daniel Klotz, and Sepp Hochreiter. Conformal prediction for time series with modern hopfield networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. URL https://arxiv.org/abs/2303.12783.

Arturs Backurs, Piotr Indyk, and Ludwig Schmidt. On the fine-grained complexity of empirical risk minimization: Kernel methods and neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. URL https://arxiv.org/abs/1704.02958.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. URL https://arxiv.org/abs/2108.07258.

Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. *arXiv preprint arXiv:2304.02207*, 2023. URL https://arxiv.org/abs/2304.02207.

Johannes Brandstetter. Blog post: Hopfield networks is all you need, 2021. URL https://ml-jku.github.io/hopfield-layers/. Accessed: April 4, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems (NeurIPS)*, 33:1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

Thomas F Burns and Tomoki Fukai. Simplicial hopfield networks. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=_QLsH8gatwx.

Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as cnf-sat. *ACM Transactions on Algorithms (TALG)*, 12(3):1–24, 2016. URL https://arxiv.org/abs/1112.2275.

Erik Demaine. Algorithmic lower bounds: Fun with hardness proofs, 2014.

Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168:288–299, 2017. URL https://arxiv.org/abs/1702.01929.

Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arXiv preprint arXiv:2304.04397*, 2023. URL https://arxiv.org/abs/2304.04397.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet T Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto, et al. Cloob: Modern hopfield networks with infoloob outperform clip. *Advances in neural information processing systems (NeurIPS)*, 35:20450–20468, 2022. URL https://arxiv.org/abs/2110.11316.

Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in llm based on tensor and svm trick, and solving it in matrix multiplication time. *arXiv preprint arXiv:2309.07418*, 2023a. URL https://arxiv.org/abs/2309.07418.

Yeqi Gao, Zhao Song, and Shenghao Xie. In-context learning for attention scheme: from single softmax regression to multiple softmax regression via a tensor trick. *arXiv preprint arXiv:2307.02419*, 2023b. URL https://arxiv.org/abs/2307.02419.

Jiuxiang Gu, Chenyang Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Exploring the frontiers of softmax: Provable optimization, applications in diffusion model, and beyond. *arXiv preprint arXiv:2405.03251*, 2024a. URL https://arxiv.org/abs/2405.03251.

Jiuxiang Gu, Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, and Junze Yin. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *arXiv preprint arXiv:2405.05219*, 2024b. URL https://arxiv.org/abs/2405.05219.

Jiuxiang Gu, Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Tensor attention training: Provably efficient learning of higher-order transformers. *arXiv preprint arXiv:2405.16411*, 2024c. URL https://arxiv.org/abs/2405.16411.

Yuzhou Gu, Zhao Song, Junze Yin, and Lichen Zhang. Low rank matrix completion via robust alternating minimization in nearly linear time. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024d. URL https://openreview.net/forum?id=N0gT4A0jNV.

Claus Hofmann, Simon Schmid, Bernhard Lehner, Daniel Klotz, and Sepp Hochreiter. Energy-based hopfield boosting for out-of-distribution detection. *arXiv preprint arXiv:2405.08766*, 2024.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed J Zaki, and Dmitry Krotov. Energy transformer. *arXiv preprint arXiv:2302.07253*, 2023. URL https://arxiv.org/abs/2302.07253.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10):3088–3092, 1984.

Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2309.12673.

Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Robin Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. In *Forty-first International Conference on Machine Learning (ICML)*, 2024a. URL https://arxiv.org/abs/2404.03828.

Jerry Yao-Chieh Hu, Bo-Yu Chen, Dennis Wu, Feng Ruan, and Han Liu. Nonparametric modern hopfield models. *arXiv preprint arXiv:2404.03900*, 2024b. URL https://arxiv.org/abs/2404.03900.

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Leo Kozachkov, Ksenia V Kastanenka, and Dmitry Krotov. Building transformers from neurons and astrocytes. *bioRxiv*, pages 2022–10, 2022.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016. URL https://arxiv.org/abs/1606.01164.

Dmitry Krotov and John J. Hopfield. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=X4y_10OX-hX.

Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. Approximate nearest neighbor search on high dimensional data-experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1475–1488, 2019. URL https://arxiv.org/abs/1610.02455.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.

Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11): 2227–2240, 2014.

Frank WJ Olver, Daniel W Lozier, Ronald F Boisvert, and Charles W Clark. *NIST handbook of mathematical functions hardback and CD-ROM*. Cambridge university press, 2010.

Fabian Paischer, Thomas Adler, Vihang Patil, Angela Bitto-Nemling, Markus Holzleitner, Sebastian Lehner, Hamid Eghbal-Zadeh, and Sepp Hochreiter. History compression via language models in reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 17156–17185. PMLR, 2022. URL https://arxiv.org/abs/2205.12258.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

Aravind Reddy, Zhao Song, and Lichen Zhang. Dynamic tensor product regression. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=hUjMhflYvGc.

Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing (STOC)*, pages 1260–1268, 2018. URL https://arxiv.org/abs/1803.00904.

Johannes Schimunek, Philipp Seidl, Lukas Friedrich, Daniel Kuhn, Friedrich Rippmann, Sepp Hochreiter, and Günter Klambauer. Context-enriched molecule representations improve few-shot drug discovery. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=XrMWUuEevr.

Philipp Seidl, Philipp Renz, Natalia Dyubankova, Paulo Neves, Jonas Verhoeven, Jorg K Wegner, Marwin Segler, Sepp Hochreiter, and Gunter Klambauer. Improving few- and zero-shot reaction template prediction using modern hopfield networks. *Journal of chemical information and modeling*, 62(9):2111–2120, 2022.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620 (7972):172–180, 2023. URL https://arxiv.org/abs/2212.13138.

Zhao Song, David Woodruff, Zheng Yu, and Lichen Zhang. Fast sketching of polynomial kernels of polynomial degree. In *International Conference on Machine Learning (ICML)*, pages 9812–9823. PMLR, 2021. URL https://arxiv.org/abs/2108.09420.

Zhao Song, Xin Yang, Yuanyuan Yang, and Lichen Zhang. Sketching meets differential privacy: fast algorithm for dynamic kronecker projection maintenance. In *International Conference on Machine Learning (ICML)*, pages 32418–32462. PMLR, 2023. URL https://arxiv.org/abs/2210.11542.

Zhao Song, Junze Yin, and Lichen Zhang. Solving attention kernel regression problem via pre-conditioner. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 208–216. PMLR, 02–04 May 2024a. URL https://proceedings.mlr.press/v238/song24a.html.

Zhao Song, Lichen Zhang, and Ruizhe Zhang. Training multi-layer over-parametrized neural network in subquadratic time. In *Innovations in Theoretical Computer*

Science (ITCS), 2024b. URL https://arxiv.org/abs/2112.07628.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

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 (NeurIPS)*, 30, 2017. URL https://arxiv.org/abs/1706.03762.

Michael Widrich, Bernhard Schäfl, Milena Pavlović, Hubert Ramsauer, Lukas Gruber, Markus Holzleitner, Johannes Brandstetter, Geir Kjetil Sandve, Victor Greiff, Sepp Hochreiter, et al. Modern hopfield networks and attention for immune repertoire classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18832–18845, 2020. URL https://arxiv.org/abs/2007.13505.

Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de janeiro 2018*, pages 3447–3487. World Scientific, 2018.

Dennis Wu, Jerry Yao-Chieh Hu, Teng-Yun Hsiao, and Han Liu. Uniform memory retrieval with larger capacity for modern hopfield models. In *Forty-first International Conference on Machine Learning (ICML)*, 2024a. URL https://arxiv.org/abs/2404.03827.

Dennis Wu, Jerry Yao-Chieh Hu, Weijian Li, Bo-Yu Chen, and Han Liu. Stanhop: Sparse tandem hopfield model for memory-enhanced time series prediction. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b. URL https://arxiv.org/abs/2312.17346.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023. URL https://arxiv.org/abs/2303.17564.

Chenwei Xu, Yu-Chao Huang, Jerry Yao-Chieh Hu, Weijian Li, Ammar Gilani, Hsi-Sheng Goan, and Han Liu. Bishop: Bi-directional cellular learning for tabular data with generalized sparse modern hopfield model. In *Forty-first International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2404.03830.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023. URL https://arxiv.org/abs/2306.15006.

Zhihan Zhou, Weimin Wu, Harrison Ho, Jiayi Wang, Lizhen Shi, Ramana V Davuluri, Zhong Wang, and Han Liu. Dnabert-s: Learning species-aware dna embedding with genome foundation models. *ArXiv*, 2024. URL https://arxiv.org/abs/2402.08777.

Supplementary Material

A. Supplementary Theoretical Backgrounds

A.1. Low-Degree Approximation of $\exp$ Function

Here we present some useful known results for later convenience.

Lemma A.1 (Approximation Degree of e^x, Theorem 1.3 of (Aggarwal and Alman, 2022))**.** For any real number $B \geq 1$ and $\delta \in (0,1)$, and function $f : [0,B] \to \mathbb{R}$, there is a polynomial function $P : \mathbb{R} \to \mathbb{R}$ of degree tightly bounded by

$$d_{B,\delta}(f = e^x) = \Theta\left(\max\left\{B, \frac{\log(1/\delta)}{\log\left[1/B \cdot \log(1/\delta)\right]}\right\}\right),$$

such that $\sup_{x\in[0,B]} |P(x) - \exp\{x\}| < \delta$.

The polynomial $P(x)$ with degree $d_{B,\delta}(e^x)$ can be computed in $\text{poly}(d_{B,\delta}(e^x))$ time.

Lemma A.2 (Corollary 2.2 of (Alman and Song, 2023))**.** For any real number $B \geq 1$ and $\delta \in (0,1)$, and function $f : [-B,B] \to \mathbb{R}$, there is a polynomial function $P : \mathbb{R} \to \mathbb{R}$ of degree tightly bounded by

$$d_{B,\delta}(f = e^x) = \Theta\left(\max\left\{B, \frac{\log(1/\delta)}{\log\left[1/B \cdot \log(1/\delta)\right]}\right\}\right),$$

such that $\sup_{x\in[0,B]} |P(x) - \exp\{x\}| < \delta \cdot \exp\{x\}$.

For more related topics and techniques, please see (Gao et al., 2023a;b; Song et al., 2023; Reddy et al., 2022) for fast approximation algorithms of attention and tensor regression via tensor trick, (Gu et al., 2024d) for low-rank matrix completion, (Song et al., 2024a; Deng et al., 2023; Brand et al., 2023; Song et al., 2021) for attention kernel regression, and (Gu et al., 2024a;b;c; Alman and Song, 2024a; Song et al., 2024b) for low-rank gradient computation in machine learning and large foundation models.

A.2. Additional Theoretical Results: Matrix Multiplication Polynomial Approximation

Here, we introduce a helper lemma for approximating an exponential function where the exponent involves matrix multiplication in the context of cross-attention. This lemma is instrumental in proving Lemma 3.1.

Lemma A.3 (Generalized from Lemma 3.2 of (Alman and Song, 2023))**.** Consider a polynomial function $P(x)$ representing a degree-g polynomial. Given matrices $\mathbf{X} \in \mathbb{R}^{M\times d}$ and $\mathbf{Y} \in \mathbb{R}^{L\times d}$, there exists an algorithm with a running complexity $\mathcal{O}(\max\{M,L\} \cdot rg)$, where $r = \binom{2(g+d)}{2g}$. This algorithm, upon receiving matrices $\mathbf{X}, \mathbf{Y}$ as input, constructs matrices $\mathbf{U}_1, \mathbf{U}_2$ that satisfy the equality $P(\mathbf{X}\mathbf{Y}^\mathsf{T}) = \mathbf{U}_1\mathbf{U}_2^\mathsf{T}$, where $\mathbf{U}_1 \in \mathbb{R}^{M\times r}$ and $\mathbf{U}_2 \in \mathbb{R}^{L\times r}$.

Proof. See Appendix B.1 for a detailed proof. $\square$

B. Proofs of Main Text

B.1. Proof of Lemma A.3

Proof. For vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, define the union of the components $\mathcal{V} := \{u_1, \cdots, u_d, v_1, \cdots, v_d\}$. Define $\mathcal{F}$ as the set of functions f such that:

$$\mathcal{F} := \left\{ f : \mathcal{V} \to \{0, 1, 2, \cdots, 2g\} \Big| \sum_{v \in \mathcal{V}} f(v) \leq 2g \right\}.$$

The cardinality of $\mathcal{F}$ is derived by solving combination-with-repetition problems, leading to the expression:

$$|\mathcal{F}| = \binom{2d + 2g}{2g}.$$

$P(x)$ can be written as:

$$P(x) = \sum_{i=0}^{g} c_i \cdot x^i.$$

Let $\mathbf{u} := [u_1, \cdots, u_d] \in \mathbb{R}^d$ and $\mathbf{v} := [v_1, \cdots, v_d] \in \mathbb{R}^d$. Consider the polynomial $P(\langle \mathbf{u}, \mathbf{v} \rangle)$:

$$P(\langle \mathbf{u}, \mathbf{v} \rangle) = \sum_{i=0}^{g} c_i \cdot (\langle \mathbf{u}, \mathbf{v} \rangle)^i.$$

There exists a set of constant c_f associated with each function $f \in \mathcal{F}$, such that:

$$\sum_{i=0}^{g} c_i \cdot (\langle \mathbf{u}, \mathbf{v} \rangle)^i = \sum_{f \in \mathcal{F}} c_f \cdot \prod_{v \in \mathcal{V}} v^{f(v)}.$$

Define two vector-valued functions $\boldsymbol{\phi}_u, \boldsymbol{\phi}_v : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{F}|}$. For each $f \in \mathcal{F}$, we define the elements of $\boldsymbol{\phi}_u, \boldsymbol{\phi}_v$ as follows:

$$\boldsymbol{\phi}_{u,f}(\mathbf{u}) = c_f \cdot \prod_{l=1}^{d} u_l^{f(u_l)}, \quad \boldsymbol{\phi}_{v,f}(\mathbf{v}) = \prod_{l=1}^{d} v_l^{f(v_l)}.$$

Thus, $P(\langle \mathbf{u}, \mathbf{v} \rangle)$ becomes:

$$P(\langle \mathbf{u}, \mathbf{v} \rangle) = \langle \boldsymbol{\phi}_u(\mathbf{u}), \boldsymbol{\phi}_v(\mathbf{v}) \rangle.$$

Since $f \leq 2g$, both $\phi_{u,f}$ and $\phi_{v,f}$ require $\mathcal{O}(g)$ time. Furthermore, the inner product $\langle \boldsymbol{\phi}_u(\mathbf{u}), \boldsymbol{\phi}_v(\mathbf{v}) \rangle$ requires $\mathcal{O}(rg)$ time, where $r = |\mathcal{F}|$.

Consider the input matrices $\mathbf{X}, \mathbf{Y}$. Let $\{\mathbf{x}_i\}_{i \in [L]}$, $\{\mathbf{y}_i\}_{i \in [L]}$ be the i-th row vector of matrix $\mathbf{X}$, $\mathbf{Y}$. The polynomial can be generalized to:

$$\begin{aligned} P(\mathbf{X}\mathbf{Y}^\mathsf{T}) &= \begin{bmatrix} P(\langle \mathbf{x}_1, \mathbf{y}_1 \rangle) & P(\langle \mathbf{x}_1, \mathbf{y}_2 \rangle) & \dots & P(\langle \mathbf{x}_1, \mathbf{y}_L \rangle) \\ P(\langle \mathbf{x}_2, \mathbf{y}_1 \rangle) & P(\langle \mathbf{x}_2, \mathbf{y}_2 \rangle) & \dots & P(\langle \mathbf{x}_2, \mathbf{y}_L \rangle) \\ \vdots & \vdots & \ddots & \vdots \\ P(\langle \mathbf{x}_M, \mathbf{y}_1 \rangle) & P(\langle \mathbf{x}_M, \mathbf{y}_2 \rangle) & \dots & P(\langle \mathbf{x}_M, \mathbf{y}_L \rangle) \end{bmatrix} \\ &= \begin{bmatrix} \langle \boldsymbol{\phi}_u(\mathbf{x}_1), \boldsymbol{\phi}_v(\mathbf{y}_1) \rangle & \langle \boldsymbol{\phi}_u(\mathbf{x}_1), \boldsymbol{\phi}_v(\mathbf{y}_2) \rangle & \dots & \langle \boldsymbol{\phi}_u(\mathbf{x}_1), \boldsymbol{\phi}_v(\mathbf{y}_L) \rangle \\ \langle \boldsymbol{\phi}_u(\mathbf{x}_2), \boldsymbol{\phi}_v(\mathbf{y}_1) \rangle & \langle \boldsymbol{\phi}_u(\mathbf{x}_2), \boldsymbol{\phi}_v(\mathbf{y}_2) \rangle & \dots & \langle \boldsymbol{\phi}_u(\mathbf{x}_2), \boldsymbol{\phi}_v(\mathbf{y}_L) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \boldsymbol{\phi}_u(\mathbf{x}_M), \boldsymbol{\phi}_v(\mathbf{y}_1) \rangle & \langle \boldsymbol{\phi}_u(\mathbf{x}_M), \boldsymbol{\phi}_v(\mathbf{y}_2) \rangle & \dots & \langle \boldsymbol{\phi}_u(\mathbf{x}_M), \boldsymbol{\phi}_v(\mathbf{y}_L) \rangle \end{bmatrix}. \end{aligned}$$

Therefore, we can constuct matrices $\mathbf{U}_1 \in \mathbb{R}^{M\times|\mathcal{F}|}$ and $\mathbf{U}_2 \in \mathbb{R}^{L\times|\mathcal{F}|}$ as follows:

$$\mathbf{U}_1 = \left[\boldsymbol{\phi}_u(\mathbf{X}_1)\ \boldsymbol{\phi}_u(\mathbf{X}_2)\ \cdots\ \boldsymbol{\phi}_u(\mathbf{X}_M)\right]^\top,$$
$$\mathbf{U}_2 = \left[\boldsymbol{\phi}_v(\mathbf{Y}_1)\ \boldsymbol{\phi}_v(\mathbf{Y}_2)\ \cdots\ \boldsymbol{\phi}_v(\mathbf{Y}_L)\right]^\top.$$

It's trivial to observe $P(\mathbf{X}\mathbf{Y}^\top) = \mathbf{U}_1\mathbf{U}_2^\top$. Moreover, constructing $\mathbf{U}_1, \mathbf{U}_2$ require time $\mathcal{O}(\max\{M, L\}\cdot rg)$. □

B.2. A Matrix in Proof of Theorem 2.1

$$\mathbf{A} = \begin{bmatrix}
\exp\left\{\frac{B^2}{\widetilde{d}}\langle a_1, b_1\rangle\right\} & \exp\left\{\frac{B^2}{\widetilde{d}}\langle a_1, b_2\rangle\right\} & \cdots & \exp\left\{\frac{B^2}{\widetilde{d}}\langle a_1, b_n\rangle\right\} & \exp\{B^2\} & \exp\{B^2\} & \cdots & \exp\{B^2\} \\
\exp\left\{\frac{B^2}{\widetilde{d}}\langle a_2, b_1\rangle\right\} & \exp\left\{\frac{B^2}{\widetilde{d}}\langle a_2, b_2\rangle\right\} & \cdots & \exp\left\{\frac{B^2}{\widetilde{d}}\langle a_2, b_n\rangle\right\} & \exp\{B^2\} & \exp\{B^2\} & \cdots & \exp\{B^2\} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\exp\left\{\frac{B^2}{\widetilde{d}}\langle a_n, b_1\rangle\right\} & \exp\left\{\frac{B^2}{\widetilde{d}}\langle a_n, b_2\rangle\right\} & \cdots & \exp\left\{\frac{B^2}{\widetilde{d}}\langle a_n, b_n\rangle\right\} & \exp\{B^2\} & \exp\{B^2\} & \cdots & \exp\{B^2\} \\
0 & 0 & \cdots & 0 & \exp\{B^2\} & \exp\{B^2\} & \cdots & \exp\{B^2\} \\
0 & 0 & \cdots & 0 & \exp\{B^2\} & \exp\{B^2\} & \cdots & \exp\{B^2\} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & \exp\{B^2\} & \exp\{B^2\} & \cdots & \exp\{B^2\}
\end{bmatrix}. \tag{B.1}$$

B.3. Formal Statement and Proof of Corollary 3.2.1

Let $\delta := -2MB\delta_A \leq 0$.

Theorem B.1 (Memory Capacity Lower Bound, Formal)**.** Suppose the probability of successfully storing and retrieving memory pattern is given by $1-p$. The number of memory patterns sampled from a sphere of radius m that the textitefficient modern Hopfield model (approximate (1.1) with Algorithm 1) can store and retrieve has a lower bound: $M \geq \sqrt{p}C^{\frac{d-1}{4}}$, where C is the solution for $C = {}^{b}/_{W_0(\exp\{a+\ln b\})}$ with $W_0(\cdot)$ being the principal branch of Lambert W function, $a := {}^{4}/_{d-1}\left\{\ln\left[{}^{2m(\sqrt{p}-1)}/_{(R-2MB\delta_A)}\right]+1\right\}$ and $b := {}^{4m^2\beta}/_{5(d-1)}$.

Remark B.1. For details and background of Lambert W function, we refer the readers to (Olver et al., 2010).

Before the main proof, we introduce the following helper lemma. Let $m := \mathrm{Max}_{\mu\in[M]}\|\boldsymbol{\xi}_\mu\|$.

Lemma B.1. Then, the well-separation condition of memory patterns is:

$$\Delta_\mu \geq \frac{1}{\beta}\ln\left(\frac{2(M-1)m}{R-2MB\delta_A}\right) + 2mR. \tag{B.2}$$

If $2MB\delta_A = 0$, (B.2) reduces to well-separation condition of Softmax-based Hopfield model (Ramsauer et al., 2021).

Proof of Lemma B.1. Let $\mathcal{T}_{\text{Dense}}$ be the retrieval dynamics given by the dense modern Hopfield model (Ramsauer et al., 2021), and $\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|$ and $\|\mathcal{T}_{\text{Dense}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|$ be the approximated efficient and dense modern Hopfield model, respectively.

By (Ramsauer et al., 2021, Lemma A.4), we have

$$\begin{aligned}
&\|\mathcal{T}_{\text{Dense}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \\
&\leq 2m(M-1)\exp\left\{-\beta\left(\langle\boldsymbol{\xi}_\mu, \mathbf{x}\rangle - \underset{\nu\in[M]}{\mathrm{Max}}\langle\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu\rangle\right)\right\}, \\
&\leq 2m(M-1)\exp\{-\beta(\Delta_\mu - 2mR)\},
\end{aligned}$$

where R is radius of the sphere S_μ.

By Theorem 3.2, the retrieval error $\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|$ has an upper bound:

$$\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq 2(M-1)\exp\{-\beta\left(\Delta_\mu - 2mR + \delta\right)\}m - \delta$$

Therefore, for $\mathcal{T}$ to be a mapping $\mathcal{T} : S_\mu \to S_\mu$, we need

$$2(M-1)\exp\{-\beta\left(\Delta_\mu - 2mR + \delta\right)\}m - \delta \leq R$$

This deduces the well-separation condition for this almost linear time model

$$\Delta_\mu \geq \frac{1}{\beta}\ln\left(\frac{2(M-1)m}{R - 2MB\delta_A}\right) + 2mR.$$

This completes the proof. □

Now we start the main proof of Theorem B.1.

Proof of Theorem B.1. We first observe that (B.1) has a slightly tighter lower bound compared to its original counterpart (Ramsauer et al., 2021), we note that under the condition identified in Remark 3.1, the new well-separation condition Lemma B.1 features a smaller denominator inside the logarithmic term. Following a similar approach to that in (Wu et al., 2024b, Lemma 3.4), we complete the proof and obtain a slightly smaller, yet still exponential-in-d, memory capacity lower bound. This is an expected consequence of an efficient-accuracy tradeoff. □