Nonparametric Modern Hopfield Models

Anonymous Author
Anonymous Institution

Abstract

We present a nonparametric construction for deep learning compatible modern Hopfield models and utilize this framework to debut an efficient variant. Our core innovation stems from interpreting the memory storage and retrieval processes in modern Hopfield models as soft-margin support vector regression problems. Crucially, the proposed framework not only facilitates the replication of known results from the original dense modern Hopfield model but also fills the void in the literature regarding efficient modern Hopfield models, by introducing *sparse-structured* modern Hopfield models with sub-quadratic complexity. We establish that this sparse model inherits the appealing theoretical properties of its dense analogue — connection with transformer attention, rapid fixed point convergence and exponential memory capacity — even without knowing details of the Hopfield energy function. Additionally, we showcase the versatility of our framework by constructing a family of modern Hopfield models as extensions, including linear, multi-head, top-K and positive random feature modern Hopfield models. Empirically, we validate the efficacy of our framework in both synthetic and real-world settings.

1 Introduction

We tackle the challenges in computational efficiency of modern Hopfield models [Hu et al., 2023, Ramsauer et al., 2020] by presenting a nonparametric framework, and then debuting the first (to our knowledge) efficient modern Hopfield model with sub-quadratic complexity and appealing theoretical properties. Such a construction is of practical importance. As in many Hopfield-centric methods [Schimunek et al., 2023, Fürst et al., 2022, Paischer et al., 2022, Seidl et al., 2022, Widrich et al., 2020], modern Hopfield models (and their derived deep learning layers) serve as powerful alternatives to the attention mechanism with additional functionalities, but lack efficient implementation for gigantic deep models [Hu et al., 2023, Section C.2]. This issue becomes more prominent in this era of Large Foundation Models [Bommasani et al., 2021], where huge attention-based models, pretrained on massive datasets, play a central role not only in machine learning but also in a wide range of scientific domains, such as ChatGPT [Floridi and Chiriatti, 2020] for natural language, BloombergGPT [Wu et al., 2023] for finance, DNABERT [Ji et al., 2021] for genomics, and many others. To push toward Hopfield-based large foundation models, this work provides a timely efficient solution, back-boned by a solid theoretical ground.

Modern Hopfield models [Ramsauer et al., 2020], motivated by the dense associative memory models [Demircigil et al., 2017, Krotov and Hopfield, 2016], are (auto-)associative memory models that (i) have exponential memory capacity, (ii) retrieve stored patterns based on input queries with only one retrieval step, and (iii) are compatible with deep learning architectures. They achieve (i) by adopting highly non-linear energy functions, (ii) by adopting a memory-retrieval dynamics ensuring monotonic minimization of the energy function[1], and (iii) by the connection between their memory retrieval dynamics and attention mechanism. Deepening (ii) and (iii), Hu et al. [2023] propose a theoretical framework for deriving modern Hopfield models using various entropic regularizers. In addition, they introduce a sparse extension of the original modern Hopfield model to handle its computational burden and vulnerability to noise. As a result, their proposal not only connects to sparse attention mechanism [Martins and Astudillo, 2016] but also offers both provably computational advantages and robust empirical performance.

However, there are still some missing pieces toward a unified theoretical framework for modern Hopfield models:

Preliminary work. Under review by AISTATS 2024. Do not distribute.

[1] Recall that, Hopfield models achieve memory storage and retrieval via embedding the memory patterns in the energy landscape of a physical system (e.g., the Ising model in [Hopfield, 1982, Peretto and Niez, 1986]), where each memory pattern corresponds to a local minimum. When a query is given, the model initiates energy-minimizing retrieval dynamics at the query, which then navigate the energy landscape to find the nearest local minimum, effectively retrieving the memory most similar to the query.

- **(P1) Lack of Efficiency.** Computationally, while Hu et al. [2023] indeed introduce sparsity into their model, this sparsity does not implies computational efficiency. In fact, it only increases efficiency at the level of memory retrieval, (i.e. the sparsity in [Hu et al., 2023] only leads to faster memory retrieval but not necessarily shorter running time, as discussed in [Hu et al., 2023, Section C.2]). Namely, the sparse modern Hopfield model still suffers by the $\mathcal{O}(n^2)$ complexity (with the input sequence length n), which hampers its scalability.
- **(P2) Lack of Rigorous Analysis on Sparsity.** Theoretically, because Hu et al. [2023] choose not to make strong assumptions (on the memory and query patterns) in order to maintain their model's generality, they only offer qualitative justifications [Hu et al., 2023, Section 3]. They do not rigorously characterize how sparsity impacts different aspects of the sparse model, e.g., the retrieval error, the well-separation condition, and the memory capacity.
- **(P3) Incomplete Connection between Attention and Hopfield Models.** Methodologically, while numerous variants of the attention module exist [Choromanski et al., 2021, Katharopoulos et al., 2020, Beltagy et al., 2020, Child et al., 2019], [Hu et al., 2023] only bridges a subset of them to modern Hopfield models. A natural question arises: How can we integrate the advancements of state-of-the-art attention into modern Hopfield models? As noted in [Hu et al., 2023], this question is far from trivial. Naively substituting the softmax activation function with other alternatives does not necessarily yield well-defined Hopfield models and might sabotage their desirable properties and functionalities.

To fill these gaps, this work presents a nonparametric framework for deep learning compatible modern Hopfield models. To fill **(P1)**, this framework allows us to not only recover the standard dense modern Hopfield model [Ramsauer et al., 2020], but also introduce an efficient modern Hopfield model, termed sparse-structured modern Hopfield model (Lemma 3.2). To fill **(P2)**, our framework facilitates the derivation of a retrieval error bound of the sparse modern Hopfield with explicit sparsity dependence (Theorem 4.1). This bound offers rigorous characterizations of the sparsity-induced advantages of the sparse model compared with its dense counterpart, including higher precision in memory retrieval (Corollary 4.1.1 and Corollary 4.1.2), enhanced robustness to noise (Remark 4.2) and exponential-in-d capacity (Theorem 4.2 and Lemma 4.2, d refers to pattern size). Interestingly, unlike existing Hopfield models [Hu et al., 2023, Ramsauer et al., 2020] requiring an explicit energy function to guarantee the stability of the model, we show that the sparse modern Hopfield model guarantees the fixed-point convergence even without details of the Hopfield energy function (Lemma 4.1). To fill **(P3)**, beyond introducing the sparse modern Hopfield model, our framework supports a family of modern Hopfield models that connect with various attention variants. This complements the findings in [Hu et al., 2023], pushing us toward a more unified understanding.

Contributions. Our contributions are as follows:

- We propose a nonparametric framework for deep learning compatible modern Hopfield models. Building upon this, we introduce the first efficient sparse modern Hopfield model with sub-quadratic complexity.
- We provide rigorous characterizations of the sparsity-induced advantages of the proposed efficient model: tighter retrieval error bound (Corollary 4.1.1 and Corollary 4.1.2), stronger noise robustness (Remark 4.2) and exponential-d-capacity (Theorem 4.2 and Lemma 4.2).
- Based on the proposed framework, we construct a family of modern Hopfield models connecting to many existing attention variants [Choromanski et al., 2021, Zaheer et al., 2020, Beltagy et al., 2020, Katharopoulos et al., 2020], and verify their efficacy through thorough numerical experiments in both synthetic and realistic settings.

Related Works

Modern Hopfield Models. The classical Hopfield models [Hopfield, 1984, 1982, Krotov and Hopfield, 2016] are canonical models of the human brain's associative memory. Their primary function is the storage and retrieval of specific memory patterns. Recently, a resurgence of interest in Hopfield models within the machine learning field is attributed to developments in understanding memory storage capacities [Krotov and Hopfield, 2016, Demircigil et al., 2017], innovative architecture [Hoover et al., 2023, Seidl et al., 2022, Fürst et al., 2022, Ramsauer et al., 2020], and their biologically-grounded rationale [Kozachkov et al., 2022, Krotov and Hopfield, 2021]. Notably, the modern Hopfield models [Ramsauer et al., 2020, Brandstetter, 2021], demonstrate not only a strong connection to the transformer attention mechanisms in deep learning, but also superior performance, and a theoretically guaranteed exponential memory capacity. In this regard, seeing the modern Hopfield models as an advanced extension of attention mechanisms opens up prospects for crafting Hopfield-centric architectural designs. Therefore, their applicability spans diverse areas like drug discovery [Schimunek et al., 2023], immunology [Widrich et al., 2020], time series forecasting [Auer et al., 2023], reinforcement learning [Paischer et al., 2022], and large language models [Fürst et al., 2022]. This work emphasizes refining this line of research towards efficient models. We posit that this effort is crucial in guiding future research towards Hopfield-driven design paradigms, especially for larger models.

Sparse Modern Hopfield Model. [Ramsauer et al., 2020] establish a connection between Hopfield models and the vanilla softmax attention. Motivated by this connection, [Hu et al., 2023] propose a theoretical framework for modern Hopfield models based on the relationship between entropic regularizers and finite-domain distributions with

varying support sets. Importantly, they not only show that [Ramsauer et al., 2020] is just special case within their framework but also propose a sparse extension with superior properties (e.g., robust representation learning, fast fixed-point convergence, and exponential memory capacity) and connection to certain types of sparse attention. However, this is not end of the story. As highlighted in [Hu et al., 2023, Section E], their framework only bridges a subset of existing attention variants (with dense quadratic attention score matrix) and hence is not complete. This work fills this theoretical gap by providing a principle construction for the many modern Hopfield models with theoretical guarantees. Moreover, our framework supports a family of modern Hopfield models mirroring many popular structured efficient attention mechanisms, including Attention with Pre-defined Patterns (each sequence token attends to a predetermined subset of tokens instead of the entire sequence, e.g, Big Bird [Zaheer et al., 2020], Longformer [Beltagy et al., 2020], Blockwise [Qiu et al., 2019], Sparse [Child et al., 2019]), and Kernelized Attention (e.g., Performer [Choromanski et al., 2021], Linear [Clevert et al., 2015] and Multi-head [Vaswani et al., 2017]).

Notations. We denote 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 $I \in \mathbb{N}_+$. The spectral norm is denoted by $\|\cdot\|$, which is equivalent to the l_2-norm when applied to a vector. 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 $\{\boldsymbol{\xi}_\mu\}_{\mu \in [M]}$. Moreover, we set norm $n := \|\mathbf{x}\|$ be the norm of the query pattern, and $m := \operatorname{Max}_{\mu \in [M]} \|\boldsymbol{\xi}_\mu\|$ be the largest norm of memory patterns.

Organization. Section 2 reviews modern Hopfield models. Section 3 presents a nonparametric construction for modern Hopfield models, and debut the sparse-structured (efficient) modern Hopfield models. Section 4 provides the theoretical analysis on the sparse-structured modern Hopfield models. Appendix C includes a family of modern Hopfield models as possible extensions. We conduct numerical experiments to support our framework in Appendix E.

2 Background: Modern Hopfield Models

Let $\mathbf{x} \in \mathbb{R}^d$ be the input query pattern and $\boldsymbol{\Xi} = (\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_M) \in \mathbb{R}^{d \times M}$ the M memory patterns.

Hopfield Models. The aim of Hopfield models is to store these memory patterns $\boldsymbol{\Xi}$ and retrieve a specific memory $\boldsymbol{\xi}_\mu$ when given a query $\mathbf{x}$. The models achieve these by embedding the memories in the energy landscape $E(\mathbf{x})$ of a physical system (e.g., the Ising model in [Hopfield, 1982] or its higher-order generalizations [Lee et al., 1986, Peretto and Niez, 1986, Newman, 1988]), where each memory $\boldsymbol{\xi}_\mu$ corresponds to a local minimum. When a query $\mathbf{x}$ is presented, the model initiates energy-minimizing retrieval dynamics $\mathcal{T}$ at the query, which then navigate the energy landscape to find the nearest local minimum, effectively retrieving the memory most similar to the query.

These models comprise two primary components: an *energy function* $E(\mathbf{x})$ that encodes memories into its local minima, and a *retrieval dynamics* $\mathcal{T}(\mathbf{x})$ that fetches a memory by iteratively minimizing $E(\mathbf{x})$ starting with a query.

Constructing the energy function, $E(\mathbf{x})$, is straightforward. As outlined in [Krotov and Hopfield, 2016], memories get encoded into $E(\mathbf{x})$ using the *overlap-construction*: $E(\mathbf{x}) = F(\boldsymbol{\Xi}^\mathsf{T}\mathbf{x})$, where $F : \mathbb{R}^M \to \mathbb{R}$ is a smooth function. This ensures that the memories $\{\boldsymbol{\xi}_\mu\}_{\mu \in [M]}$ sit at the stationary points of $E(\mathbf{x})$, given $\nabla_\mathbf{x} F(\boldsymbol{\Xi}^\mathsf{T}\mathbf{x})|_{\boldsymbol{\xi}_\mu} = 0$ for all $\mu \in [M]$. The choice of F results in different Hopfield model types, as demonstrated in [Krotov and Hopfield, 2016, Demircigil et al., 2017, Ramsauer et al., 2020, Krotov and Hopfield, 2021]. However, determining a suitable retrieval dynamics, $\mathcal{T}$, for a given energy $E(\mathbf{x})$ is more challenging. For effective memory retrieval, $\mathcal{T}$ must:

(T1) Monotonically reduce $E(\mathbf{x})$ when applied iteratively.

(T2) Ensure its fixed points coincide with the stationary points of $E(\mathbf{x})$ for precise retrieval.

Modern Hopfield Models. Ramsauer et al. [2020] propose the modern Hopfield model with a specific set of E and $\mathcal{T}$ satisfying above requirements, and integrate it into deep learning architectures via its strong connection with attention mechanism, offering enhanced performance, and theoretically guaranteed exponential memory capacity. Specifically, they introduce the energy function:

$$E(\mathbf{x}) = -\operatorname{lse}(\beta, \boldsymbol{\Xi}^\mathsf{T}\mathbf{x}) + \frac{1}{2}\langle \mathbf{x}, \mathbf{x} \rangle, \tag{2.1}$$

where the retrieval dynamics is given by

$$\mathbf{x}^{\text{new}} = \mathcal{T}_{\text{Dense}}(\mathbf{x}) = \boldsymbol{\Xi} \cdot \operatorname{Softmax}(\beta \boldsymbol{\Xi}^\mathsf{T}\mathbf{x}). \tag{2.2}$$

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 $\beta > 0$. Their analysis reveals that:

(i) The $\mathcal{T}_{\text{Dense}}$ dynamics converge well (T2) and can retrieve patterns accurately in just one step (T1).

(ii) The modern Hopfield model from (2.1) possesses an exponential memory capacity in pattern size d.

(iii) Notably, the one-step approximation of $\mathcal{T}_{\text{Dense}}$ mirrors the attention mechanism in transformers, leading to a novel architecture design: the Hopfield layers.

Attention $\leftrightarrow$ Modern Hopfield Model. To see above (iii), suppose that $\mathbf{X}$ and $\boldsymbol{\Xi}$ are embedded from the *raw* query $\mathbf{R}$ and $\mathbf{Y}$ memory patterns, respectively, via $\mathbf{X}^\mathsf{T} = \mathbf{R}\mathbf{W}_Q := \mathbf{Q}$, and $\boldsymbol{\Xi}^\mathsf{T} = \mathbf{Y}\mathbf{W}_K := \mathbf{K}$, with some projection matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$. Then, taking the transport of $\mathcal{T}$ in (2.2) and multiplying with $\mathbf{W}_V$ such that $\mathbf{V} := \mathbf{K}\mathbf{W}_V$, we obtain

$$\mathbf{Z} := \mathbf{Q}^{\text{new}}\mathbf{W}_V = \operatorname{Softmax}\left(\beta \mathbf{Q}\mathbf{K}^\mathsf{T}\right)\mathbf{V}. \tag{2.3}$$

This result enables that the modern Hopfield models are able to serve as powerful alternatives to the attention mechanism equipped with additional functionalities.

Given this equivalence, one might wonder if the quest for efficient modern Hopfield models is equivalent to seeking efficient attention mechanisms [Tay et al., 2022], specifically in terms of finding efficient implementations of the Softmax matrix computation. We contend that they are not the same. For our rationale, please see below motivation.

Motivation. From above, we know that to build a modern Hopfield model we need it to not only connect to attention via attention-score-like matrix computation, but also be a *well-defined* Hopfield model. To be well-defined, a Hopfield model must satisfy conditions (T1) and (T2) for effective memory retrieval. Therefore, naively switching the softmax function in retrieval dynamics (2.2) with efficient alternatives do not necessarily lead to Hopfield models.

To motivate our framework, we first make the following observation based on [Ramsauer et al., 2020, Hu et al., 2023].

Claim 1. In fact, condition (T2) already implies (T1) for modern Hopfield models.

To see this claim, we follow the analysis of [Hu et al., 2023]. Let $\{\mathbf{x}_t\}_{t=0}^{\infty}$ be a sequence generated by iteratively applying $\mathcal{T}$ to get $\mathbf{x}_{t+1} := \mathcal{T}(\mathbf{x}_t)$. By Zangwill's global convergence theory [Zangwill, 1969], Hu et al. [2023, Lemma 2.2] prove that if $\mathcal{T}$ monotonically minimizes $E(\mathbf{x})$ when applied iteratively, all limit points of $\{\mathbf{x}_t\}_{t=0}^{\infty}$ are fixed points of $\mathcal{T}$, and moreover $\lim_{t\to\infty} E(\mathbf{x}_t) = E(\mathbf{x}^\star)$ with $\mathbf{x}^\star$ denoting stationary points of E. Therefore, Claim 1 is justified: (T2) itself is enough for representing a well-defined modern Hopfield model.

Then, we ask the following question:

> Given Claim 1, is it possible to construct modern Hopfield models solely based on condition (T2) (i.e., the fixed points of $\mathcal{T}$ correspond to stored memory patterns) while still linking to attention mechanisms?

This question motivates us to view the construction of $\mathcal{T}$ as a learning problem: we aim to learn a function $\mathcal{T}$ satisfying (T2) from a dataset consisting of query-memory pairs. Thus, rather than using the traditional Hopfield model's learning rule — where the model memorizes memories by defining an energy function, like the overlap-construction — we interpret the memorization process as learning a function that maps queries to memories. This new perspective allows us to construct novel modern Hopfield models that are equivalent to various attention variants.

3 Nonparametric Modern Hopfield Models

In this section, we first align the definition of $\mathcal{T}$ (the retrieval dynamics (2.2)) with a nonparametric regression problem subject to a set of query-memory pairs. Then, by solving this regression model, we derive a nonparametric formulation of $\mathcal{T}$ satisfying (T2), which is able to describe many new modern Hopfield models.

3.1 Retrieval Dynamics

Formally, the retrieval dynamics (2.2) is a map $\mathcal{T}(\boldsymbol{\Xi}, \mathbf{x}) : \mathbb{R}^{M\times d} \times \mathbb{R}^d \to \mathbb{R}^d$ that maps a input query $\mathbf{x}$ and a set of stored memory patterns $\boldsymbol{\Xi}$ to a target memory pattern $\boldsymbol{\xi}_\mu$ closest to $\mathbf{x}$. To be concrete, we start with the notion of memory storage and retrieval of modern Hopfield models following [Hu et al., 2023, Ramsauer et al., 2020].

Definition 3.1 (Stored and Retrieved)**.** Assuming that every memory pattern $\boldsymbol{\xi}_\mu$ surrounded by a sphere $\mathcal{S}_\mu$ with finite radius $R := \frac{1}{2}\operatorname{Min}_{\mu,\nu\in[M];\mu\neq\nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$, we say $\boldsymbol{\xi}_\mu$ is *stored* if there exists a generalized fixed point of $\mathcal{T}$, $\mathbf{x}_\mu^\star \in \mathcal{S}_\mu$, to which all limit points $\mathbf{x} \in \mathcal{S}_\mu$ converge to, 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 $\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \epsilon$.

Intuitively, Definition 3.1 states that $\mathcal{T}$ satisfying (T2) is a contraction map within attractor basins and its fixed points corresponds to the stored memory patterns in the modern Hopfield model. Namely, as long as the input query $\mathbf{x}$ is initialized inside the attractor basin of a memory $\boldsymbol{\xi}_\mu$, applying $\mathcal{T}$ iteratively draws $\mathbf{x}$ closer to $\boldsymbol{\xi}_\mu$ with each step. In this regard, in this work, we consider the retrieval dynamics $\mathcal{T}$ of a modern Hopfield model as a regression model aiming to map the query $\mathbf{x}$ onto $\boldsymbol{\xi}$ within an error-tolerance margin R. A natural choice is Support Vector Regression (SVR): it fits the best hyperplane to the data points within a predefined error margin, aiming to minimize the error rate while ensuring the model remains insensitive to errors within a certain threshold. We first define the regression model.

Definition 3.2 (Regression Model)**.** Given an input vector $\mathbf{x} \in \mathbb{R}^d$, and output $\mathbf{y} \in \mathbb{R}^d$. We define the regression model defined as:

$$f(\mathbf{x}) := \mathbf{y} = \mathbf{W}\frac{\Phi(\mathbf{x})}{h(\mathbf{x})} \in \mathbb{R}^d, \tag{3.1}$$

where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_d]^\mathsf{T} \in \mathbb{R}^{d\times D_\Phi}$, $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_{D_\Phi}(\mathbf{x})] : \mathbb{R}^d \to \mathbb{R}^{D_\Phi}$ and $h(\mathbf{x}) \in \mathbb{R}$ denote weight matrix, feature mapping and normalization function, respectively.

Remark 3.1. Intuitively, (3.1) is a linear regression model of kernelized features $\Phi(\mathbf{x})$, where $\Phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{D_\Phi}$ is a feature map onto a kernel space $\mathcal{H} \subseteq \mathbb{R}^{D_\Phi}$. Here, the kernel space is D_Φ-dimensional metric space with inner product kernel (norm) $\mathcal{K}(\cdot,\cdot) := \langle \Phi(\cdot), \Phi(\cdot)\rangle : \mathbb{R}^{D_\Phi} \times \mathbb{R}^{D_\Phi} \to \mathbb{R}_+$ for the given feature mapping Φ.

To cast $\mathcal{T}$ as a SVR problem using (3.1), we now specify the data points that $f(\mathbf{x})$ should fit. Since the goal of $\mathcal{T}$ is to retrieve the memory pattern most similar to given query $\mathbf{x}$, we consider the training dataset $\mathcal{D} = \{(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu)\}_{\mu\in[M]}$, where the input query $\mathbf{x} = \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu$ is the contaminated target memory pattern with noise $\delta\boldsymbol{\xi}_\mu$, and the output $\mathbf{y} = \boldsymbol{\xi}_\mu$ is target memory pattern. Further,

we make an assumption to connect the robustness of the memory retrieval model with SVR error margin.

Assumption 3.1. $\|\delta\boldsymbol{\xi}_\mu\| \leq R - \epsilon\sqrt{M}$ for some $\epsilon \geq 0$.

Remark 3.2. Assumption 3.1 ensures that the robustness of memory retrieval in Definition 3.1 is linked to the ϵ error threshold of the SVR problem defined below[2].

Next, we frame the memorization in modern Hopfield models as fitting f to the dataset $\mathcal{D}$, and obtain the following nonparametric (support vector) regression problem.

Definition 3.3 (Optimization Form of Retrieval Dynamics)**.** Given the training dataset $\mathcal{D} = \{(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu)\}_{\mu\in[M]}$. We define the support vector regression problem for $\mathcal{T}$ as

$$\underset{\mathbf{W},\boldsymbol{\eta},\widetilde{\boldsymbol{\eta}}}{\text{Min}} \left[\frac{1}{2}\|\mathbf{W}\|^2 + C\sum_{\mu=1}^{M}\sum_{i=1}^{d}(\boldsymbol{\eta}_\mu[i] + \widetilde{\boldsymbol{\eta}}_\mu[i])\right] \tag{3.2}$$

$$\text{subject to} \begin{cases} \boldsymbol{\xi}_\mu[i] - \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)} \right\rangle \leq \epsilon' + \boldsymbol{\eta}_\mu[i] \leq R' \\ \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)} \right\rangle - \boldsymbol{\xi}_\mu[i] \leq \epsilon' + \widetilde{\boldsymbol{\eta}}_\mu[i] \leq R' \\ \boldsymbol{\eta}_\mu[i], \widetilde{\boldsymbol{\eta}}_\mu[i] \geq 0. \end{cases}$$

for all $\mu \in [M]$, where $\boldsymbol{\xi}_\mu[i]$ represents i-th component of memory pattern, ϵ' denotes the SVR (component-wise) error margin, $\boldsymbol{\eta}$ and $\widetilde{\boldsymbol{\eta}}$ denote the slack variables, $R' \coloneqq \text{Max}_{i\in[d]}\{\epsilon' + \boldsymbol{\eta}_\mu[i]\} \leq \epsilon/\sqrt{M}$ and $C \geq 0$ denote penalized coefficients of the SVR problem.

Remark 3.3. We denote the solution to (3.2) as $\mathcal{T}_{\text{SVR}}(\mathbf{x})$, a SVR surrogate for $\mathcal{T}$ satisfying (T2) without knowing E.

Remark 3.4. ϵ' is the component-wise SVR error, *not* the ϵ in Hopfield retrieval error defined in Definition 3.1.

Remark 3.5. This regression problem is nonparametric. That is, it does not assume a specific functional form for $\mathcal{T}_{\text{SVR}}$ and is flexible in the number of parameters, allowing the number of support vectors to adjust based on the data.

Intuitively, this optimization problem learns a regression model to replace $\mathcal{T}$, denoted by $\mathcal{T}_{\text{SVR}}$, such that, for any given query $\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu$, it retrieves a target memory pattern $\boldsymbol{\xi}_\mu$ with ϵ precision, for all $\mu \in [M]$. Specifically, this ϵ precision comes from the upper bound of the maximum component-wise error $\epsilon' + \boldsymbol{\eta}_\mu[i]$ (and $\epsilon' + \widetilde{\boldsymbol{\eta}}_\mu[i]$): $R' \leq \epsilon/\sqrt{M}$. This choice of SVR error margin replicates the ϵ-retrieval of modern Hopfield models via the flexibility of soft-margin SVR. As a result, the objective of the SVR problem (3.2) coincides with the memorization and retrieval process of modern Hopfield models. While $\mathcal{T}$ retrieves memory patterns $\{\boldsymbol{\xi}\}$ based on $\mathbf{x}$ with an error tolerance ϵ, **(Memorization:)** the SVR problem (3.2) fits a function $\mathcal{T}_{\text{SVR}}$ that **(Retrieval:)** maps queries onto memory patterns within a component-wise error-margin $R' \leq \epsilon/\sqrt{M}$. By Assumption 3.1 (and Definition 3.1), the solution to (3.2) must satisfy (T2), leading to well-defined Hopfield models represented by $\mathcal{T}$.

The standard practice of solving the optimization problem like (3.2) is through its dual problem and the kernel trick [Awad et al., 2015]. Therefore, we construct the Lagrangian corresponding to (3.2) as

$$\begin{aligned}\mathcal{L} \coloneqq \frac{1}{2}\sum_{i=1}^{d}\|\mathbf{w}_i\|^2 + C\sum_{\mu=1}^{M}\sum_{i=1}^{d}(\boldsymbol{\lambda}_\mu[i]\boldsymbol{\eta}_\mu[i] + \widetilde{\boldsymbol{\lambda}}_\mu[i]\widetilde{\boldsymbol{\eta}}_\mu[i]) \\ - \sum_{\mu=1}^{M}\sum_{i=1}^{d}\boldsymbol{\alpha}_\mu[i]\left(\epsilon' + \boldsymbol{\eta}_\mu[i] - \boldsymbol{\xi}_\mu[i] + \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}\right\rangle\right) \\ - \sum_{\mu=1}^{M}\sum_{i=1}^{d}\widetilde{\boldsymbol{\alpha}}_\mu[i]\left(\epsilon' + \widetilde{\boldsymbol{\eta}}_\mu[i] - \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}\right\rangle + \boldsymbol{\xi}_\mu[i]\right),\end{aligned} \tag{3.3}$$

where $\boldsymbol{\lambda}_\mu[i]$, $\widetilde{\boldsymbol{\lambda}}_\mu[i]$, $\boldsymbol{\alpha}_\mu[i]$ and $\widetilde{\boldsymbol{\alpha}}_\mu[i]$ are Lagrange multipliers. By solving for optimality of (3.3), we obtain the following nonparametric framework for constructing many well-defined modern Hopfield models.

Theorem 3.1 (Nonparametric Retrieval Dynamics)**.** Let $\mathcal{T}_{\text{SVR}}$ be the optimal solution to the SVR problem (3.2) (and (3.3)). Given an input query $\mathbf{x}$, the i-th component of the retrieved pattern by applying $\mathcal{T}_{\text{SVR}}(\mathbf{x})$ once is

$$\mathbf{x}^{\text{new}}[i] = \mathcal{T}_{\text{SVR}}(\mathbf{x})[i] \coloneqq \left\langle \mathbf{w}_i^\star, \frac{\Phi(\mathbf{x})}{h(\mathbf{x})}\right\rangle, \tag{3.4}$$

where $\mathbf{w}_i^\star \in \mathbb{R}^{D_\Phi}$ is the i-th row of the learned weight, $\mathbf{W}$,

$$\mathbf{w}^\star[i] \coloneqq \sum_{\mu=1}^{M}\underbrace{\frac{\boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}}_{\in\mathbb{R}}\underbrace{\Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}_{\in\mathbb{R}^{D_\Phi}}. \tag{3.5}$$

Proof. See Appendix B.1 for a detailed proof. □

Here, $\mathbf{w}_i^\star \in \mathbb{R}^{D_\Phi}$ is a learned hyperplane in the D_Φ-dimensional kernel space such that the regression model gives memory patterns $\boldsymbol{\xi}_\mu$ with ϵ precision based on a given input query $\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu$. Importantly, Theorem 3.1 enables us to derive a family of nonparametric modern Hopfield models through constructing their retrieval dynamics with various kernel functions $\Phi(\cdot)$, including Dense [Ramsauer et al., 2020], Linear [Katharopoulos et al., 2020], Multi-Head [Vaswani et al., 2017], Sparse-Structured [Zaheer et al., 2020, Beltagy et al., 2020, Child et al., 2019] and Generalized Kernelizable or PRFs (Positive Random Features) [Choromanski et al., 2021] modern Hopfield models.

In Appendix C, we present constructions of these modern Hopfield models as extensions of our framework.

3.2 Nonparametric Dense and Sparse-Structured Modern Hopfield Models

In this section, we showcase the nonparametric framework Theorem 3.1 with two special cases. First, we recover the

[2] Specifically, we introduce this assumption because the error threshold of the SVR problem is defined via component-wise constraints [Awad et al., 2015], rather than in terms of the $\|\cdot\|$ used in modern Hopfield models' analysis [Hu et al., 2023, Ramsauer et al., 2020].

standard dense modern Hopfield model [Ramsauer et al., 2020]. Then, we introduce the efficient *sparse-structured modern Hopfield models* with sub-quadratic complexity.

Dense Modern Hopfield Model [Ramsauer et al., 2020].

Lemma 3.1 (Nonparametric Dense Modern Hopfield Model)**.** Let $h(\mathbf{x}) \coloneqq \sum_{\mu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle$ and $\Phi(\cdot) = (\phi_0^{(0)}, \phi_1^{(1)}, \ldots, \phi_{D_1}^{(1)}, \ldots, \phi_1^{(n)}, \ldots, \phi_{D_n}^{(n)}, \ldots)$ with, for $1 \leq D' \leq D_n$,

$$\phi_{D'}^{(n)} \coloneqq \frac{(\sqrt{\beta}x_1)^{\ell_1} \cdots (\sqrt{\beta}x_d)^{\ell_d}}{\sqrt{\ell_1! \cdots \ell_d!}}, \tag{3.6}$$

where $\ell_1 + \cdots + \ell_d = n$, and $D_n \coloneqq \binom{d+n-1}{n}$. By Theorem 3.1, fitting $\mathcal{T}_{\text{SVR}}$ on $\mathcal{D}$ following (3.2) gives

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \sum_{\mu=1}^{M} \text{Softmax}\left(\beta \left\langle \mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu \right\rangle\right) \boldsymbol{\xi}_\mu. \tag{3.7}$$

Proof Sketch. We first select Φ to be the Taylor expansion of the exp function via the homogeneous infinite polynomial kernel [Chen et al., 2005]. Then, by solving KKT conditions for Lemma 3.1, we arrive a retrieval dynamics resembling (2.2). See Appendix B.2 for a detailed proof. □

Remark 3.6 (*Hetero-* v.s. *Auto*-Associative Memory.)**.** So far, we derive a nonparametric framework for hetero-associative modern Hopfield models, differentiating $\mathbf{x}$ and $\mathbf{y}$ by incorporating inherent noise $\delta\boldsymbol{\xi}$ into $\mathcal{D}$. If we eliminate noises $\{\delta\boldsymbol{\xi}_\mu\}_{\mu\in[M]}$ from the training memory patterns, (3.7) reduces to that of the standard *auto-associative* dense modern Hopfield model, as shown in (2.2).

With Remark 3.6, Lemma 3.1 facilitates the replication of known results from the standard dense modern Hopfield model [Ramsauer et al., 2020].

Sparse-Structured Modern Hopfield Models. Next, we present a set of efficient modern Hopfield models with sparse-structured patterns via the following mask.

Definition 3.4 (Sparse-Structured Mask)**.** Let $\mathcal{M} \coloneqq \{\mathcal{M}(1), \ldots, \mathcal{M}(k)\} \subseteq \{1, \ldots, M\}$ be the reduced support set for $\mathcal{T}_{\text{SVR}}$ of size $k \leq M$. We obtain $\mathcal{M}$ by masking (3.7) with the following indicator function:

$$\mathbb{1}_{\mathcal{M}(\mu)} \coloneqq \begin{cases} 1 & \text{, if } \mu \in \mathcal{M}, \\ 0 & \text{, otherwise.} \end{cases} \tag{3.8}$$

With Definition 3.4, we obtain the following sparse-structured retrieval dynamics (and thereby its corresponding Hopfield model(s)) by fitting $\mathcal{T}_{\text{SVR}}$ on $\mathcal{D}$ masked by $\mathcal{M}$.

Lemma 3.2 (Sparse-Structured Modern Hopfield Models)**.** Let $h(\mathbf{x}) \coloneqq \sum_{\mu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle$ and $\Phi(\cdot) = (\phi_0^{(0)}, \phi_1^{(1)}, \ldots, \phi_{D_1}^{(1)}, \ldots, \phi_1^{(n)}, \ldots, \phi_{D_n}^{(n)}, \ldots)$ with, for $1 \leq D' \leq D_n$,

$$\phi_{D'}^{(n)} \coloneqq \frac{(\sqrt{\beta}x_1)^{\ell_1} \cdots (\sqrt{\beta}x_d)^{\ell_d}}{\sqrt{\ell_1! \cdots \ell_d!}}, \tag{3.9}$$

where $\ell_1 + \cdots + \ell_d = n$, and $D_n \coloneqq \binom{d+n-1}{n}$. By Theorem 3.1, fitting $\mathcal{T}_{\text{SVR}}$ on $\mathcal{D}$ masked by $\mathcal{M}$ following (3.2) gives

$$\mathcal{T}_{\text{Sparse}}(\mathbf{x}) = \sum_{\mu=1}^{M} \text{Softmax}\left(\beta \left\langle \mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu \right\rangle\right) \boldsymbol{\xi}_\mu \mathbb{1}_{\mathcal{M}(\mu)} \tag{3.10}$$

Proof. See Appendix B.3 for a detailed proof. □

We want to emphasize that (3.10) is in fact generic and is able to describe many sparse-structured modern Hopfield models with various support set. Importantly, it allows us to construct efficient variants with sub-quadratic complexity, and hence fills the void in the literature regarding efficient modern Hopfield models, as discussed in the limitations section of [Hu et al., 2023].

Remark 3.7 (Random Masked Modern Hopfield Model with $\mathcal{O}(kL)$ Complexity)**.** By setting $\mathbb{1}_{\mathcal{M}}$ to randomly mask $M - k$ entries, we obtain an efficient modern Hopfield model with a sub-quadratic $\mathcal{O}(kL)$ complexity. This model is by design connected to the random attention of BigBird [Zaheer et al., 2020].

Remark 3.8 (Efficient Modern Hopfield Model with $\mathcal{O}(L\sqrt{L})$ Complexity)**.** To analyze efficiency for long query sequences, we first generalized (3.10) from a single query $\mathbf{x}$ to a sequence of L query denoted by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_L)$. Let $\mathbb{I}_{\mathcal{M}} = [\mathbb{1}_{\mathcal{M}_1}, \ldots, \mathbb{1}_{\mathcal{M}_L}]$ be the corresponding sparse-sturctured mask matrix. By setting $\mathbb{1}_{\mathcal{M}}$ for each query in a way that $\mathbb{I}_{\mathcal{M}}$ reproduces the sliding window pattern of window size $\sqrt{L}$, we obtain an efficient modern Hopfield model with a sub-quadratic $\mathcal{O}(L\sqrt{L})$ complexity. This model is by design connected to the Longformer attention [Beltagy et al., 2020].

Remark 3.9 (Top-K Modern Hopfield Model)**.** By setting the vector $\mathbf{p}$ as the inner product of memory and query:

$$\mathbf{p} = \left(\langle \mathbf{x}, \boldsymbol{\xi}_1 \rangle, \ldots, \langle \mathbf{x}, \boldsymbol{\xi}_M \rangle\right), \tag{3.11}$$

and denote $p^\star$ as the k-th largest element of the vector $\mathbf{p}_\mu$. Then, we have $\mathcal{M}$ by the following indicator function:

$$\mathbb{1}_{\mathcal{M}(\mu)} \coloneqq \begin{cases} 1 & \text{, if } \mathbf{p}[\mu] \geq \mathbf{p}^\star \\ 0 & \text{, if } \mathbf{p}[\mu] < \mathbf{p}^\star, \end{cases} \tag{3.12}$$

thereby obtaining the top-K modern Hopfield model (with quadratic complexity, i.e., inefficient). This model is by design connected to the top-K attention [Gupta et al., 2021].

Next, we provide analytic characterizations of how sparsity affects the sparse-structured models defined in (3.10).

4 Theoretical Analysis of Sparse-Structured Modern Hopfield Models

In this section, our theoretical analysis on sparse modern Hopfield models[3] consists of the following two aspects:

[3] We use plural "models" as $\mathbb{1}_{\mathcal{M}}$ in (3.10) is a generic expression for many models with different sparse patterns.

1. Derive the sparsity-dependent retrieval error bound of sparse modern Hopfield model and prove its rapid convergence property compared with its dense counterpart.
2. Characterize the fundamental limit of memory capacity of the sparse-structured modern Hopfield models.

As a reminder, we adopt Definition 3.1 for memory storage and retrieval. Additionally, we recall the following definition regarding the separation between memory patterns.

Definition 4.1 (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: $\Delta_\mu := \mathrm{Min}_{\nu,\nu\neq\mu}\left[\langle\boldsymbol{\xi}_\mu,\boldsymbol{\xi}_\mu\rangle - \langle\boldsymbol{\xi}_\mu,\boldsymbol{\xi}_\nu\rangle\right]$. Similarly, the separation of $\boldsymbol{\xi}_\mu$ at a given $\mathbf{x}$ from all memory patterns $\boldsymbol{\Xi}$ is given by $\widetilde{\Delta}_\mu := \mathrm{Min}_{\nu,\nu\neq\mu}\left[\langle\mathbf{x},\boldsymbol{\xi}_\mu\rangle - \langle\mathbf{x},\boldsymbol{\xi}_\nu\rangle\right]$.

4.1 Memory Retrieval: Error Bounds & Convergence

Memory Retrieval Error Bounds. To characterize the accuracy of memory retrieval, we derive the upper bound on retrieval error of the sparse-structured models.

Theorem 4.1 (Sparsity-Dependent Retrieval Error)**.** Let $\mathcal{T}_{\text{Sparse}}$ be the sparse-structured retrieval dynamics (3.10). For query $\mathbf{x}\in\mathcal{S}_\mu$, it holds

$$\|\mathcal{T}_{\text{Sparse}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \tag{4.1}$$
$$m(M+k-2)\exp\left\{-\beta\left(\langle\boldsymbol{\xi}_\mu,\mathbf{x}\rangle - \max_{\nu\in[M],\nu\neq\mu}\langle\boldsymbol{\xi}_\mu,\boldsymbol{\xi}_\nu\rangle\right)\right\},$$

for all $\mu\in\mathcal{M}$, where $k := |\mathcal{M}| \in [M]$ denotes the size of the support set $\mathcal{M}$, and $m = \mathrm{Max}_\mu\|\boldsymbol{\xi}_\mu\|$.

Proof. See Appendix B.4 for a detailed proof. □

Interestingly, the retrieval error bound in Theorem 4.1 is sparsity-dependent, which is governed by the size of the support set $\mathcal{M}$, i.e. sparsity dimension $k := |\mathcal{M}|$.

Remark 4.1 (Comparing with the Sparse Modern Hopfield Model [Hu et al., 2023])**.** Compared to the retrieval error bound in [Hu et al., 2023], which lacks explicit dependence on its input (data)-dependent sparsity, the sparsity (size of $\mathcal{M}$) here is pre-specified. When there are fewer elements in the sparse-structured mask, i.e., when k is small, the retrieval error bound is tighter, and vice versa.

Remark 4.2 (Noise Robustness)**.** By Theorem 4.1, in cases involving contaminated query or memory, i.e. $\widetilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\delta}\mathbf{x}$ (noise in query) or $\widetilde{\boldsymbol{\xi}} = \boldsymbol{\xi} + \boldsymbol{\delta}\boldsymbol{\xi}$ (noise in memory), the impact of noise on the sparse retrieval error (4.1) is less than that its impact on the dense counterpart due to the smaller coefficient $(M+k-2)$.

Corollary 4.1.1. Let $\mathcal{T}_{\text{Dense}}$ and $\mathcal{T}_{\text{Sparse}}$ be the dense (3.10) and sparse-structured (3.10) retrieval dynamics, respectively. For any query pattern $\mathbf{x}\in\mathcal{S}_\mu$ and $\mu\in\mathcal{M}$, it holds

$$\|\mathcal{T}_{\text{Sparse}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \|\mathcal{T}_{\text{Dense}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|. \tag{4.2}$$

Proof. See Appendix B.5 for a detailed proof. □

Computationally, Corollary 4.1.1 suggests that $\mathcal{T}_{\text{Sparse}}$ necessitates fewer iterations to reach fixed points compared to $\mathcal{T}_{\text{Dense}}$, given the same error tolerance level. In other words, $\mathcal{T}_{\text{Sparse}}$ retrieves stored memory patterns faster than $\mathcal{T}_{\text{Dense}}$.

To bridge to deep learning methodologies, we show that $\mathcal{T}_{\text{Sparse}}$ retrieves memory patterns with high accuracy after a single activation in the following corollary, akin to [Hu et al., 2023, Ramsauer et al., 2020].

Corollary 4.1.2 (One-Step Retrieval with High Accuracy)**.** For any query $\mathbf{x}\in S_\mu$ and $\mu\in\mathcal{M}$, $\mathcal{T}_{\text{Sparse}}$ retrieve the memory pattern $\boldsymbol{\xi}_\mu$ with retrieval error ϵ exponentially suppressed by Δ_μ.

Proof. See Appendix B.5 for a detailed proof. □

Corollary 4.1.2 indicates that, with sufficiently large Δ_μ, $\mathcal{T}_{\text{Sparse}}$ retrieves memory patterns in a single *iteration*, allowing the integration of sparse-structured modern Hopfield models into deep learning architectures similarly to [Schimunek et al., 2023, Hoover et al., 2023, Seidl et al., 2022, Fürst et al., 2022, Paischer et al., 2022].

Fixed Point Convergence. By design, the retrieval dynamics constructed via Lemma 3.1 satisfy (T2). We now verify this adherence as a sanity check. Interestingly, while previous studies [Hu et al., 2023, Ramsauer et al., 2020] rely on the detailed energy functions to show the convergence properties of modern Hopfield models, we prove them for sparse-structured modern Hopfield models even without knowing E in the next lemma.

Lemma 4.1 (Fixed Point Convergence)**.** Let $\mathcal{T}_{\text{Sparse}}$ be the sparse-structured retrieval dynamics (3.10). For all $\mu\in\mathcal{M}$, the query $\mathbf{x}\in S_\mu$ converges to fixed point if it is iteratively applied by $\mathcal{T}_{\text{Sparse}}$.

Proof. See Appendix B.6 for a detailed proof. □

Lemma 4.1 affirms that $\mathcal{T}_{\text{Sparse}}$ in (3.10) satisfies (T2).

4.2 Memory Capacity

To characterize the fundamental limit of memory capacity, we ask the following two questions for sparse-structured modern Hopfield models following [Hu et al., 2023]:

(A) What is the necessary condition for a pattern $\boldsymbol{\xi}_\mu$ being considered well-stored, and correctly retrieved?

(B) What is the expected number of memory patterns such that the above condition is satisfied?

Well-Separation Condition. To address (A), we identify the necessary condition for a pattern being well-stored and retrieved by the sparse-structured modern Hopfield models: the well-separation condition.

Theorem 4.2 (Well-Separation Condition)**.** Following Definition 3.1, for $\mu\in\mathcal{M}$, suppose every memory pattern $\{\boldsymbol{\xi}_\mu\}_{\mu\in\mathcal{M}}$ is enclosed by a sphere $\mathcal{S}_\mu := \{\mathbf{x} \mid \|\mathbf{x}-\boldsymbol{\xi}_\mu\| \leq R\}$, with finite radius $R := \frac{1}{2}\mathrm{Min}_{\mu,\nu\in\mathcal{M};\mu\neq\nu}\|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$. Then, the retrieved dynamics $\mathcal{T}_{\text{Sparse}}$ maps $\mathcal{S}_\mu$ to itself if

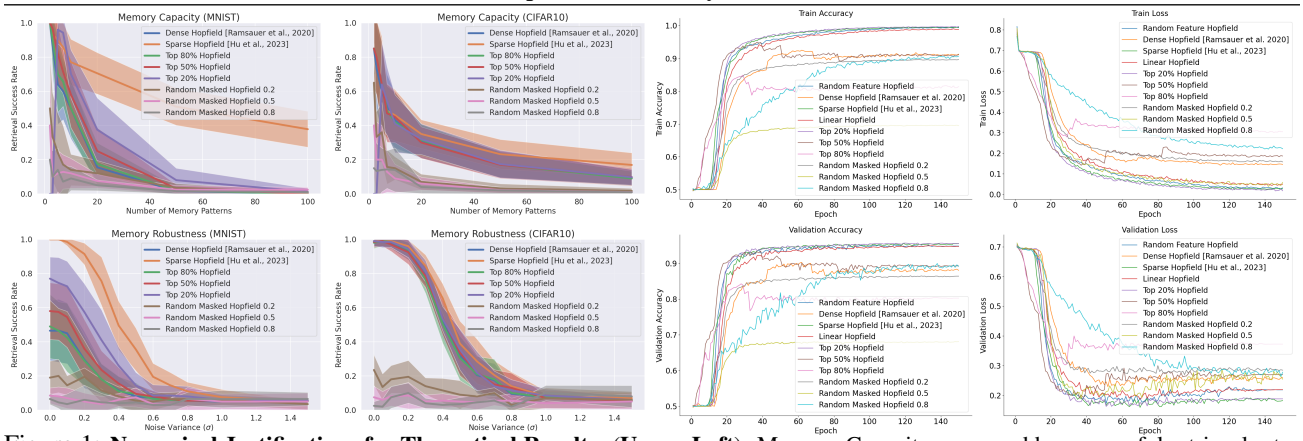

Figure 1: **Numerical Justifications for Theoretical Results.** (**Upper Left**): Memory Capacity measured by successful retrieval rates from half-masked queries (Lemma 4.2). (**Bottom Left**): Memory Robustness measured by success retrieval rates from noisy queries with different scales of Gaussian noise (Remark 4.2). For all Hopfield models, we set $\beta = .01/0.1$ (for MNIST/CIFAR10) for better visualizations. A query pattern is considered correctly retrieved if its sum-of-square similarity error is below a set threshold. For both datasets, we set the error thresholds to be 20%. Plotted are the means and standard deviations of 10 runs. We see that Top-K Hopfield shows similar exponential capacity as Dense [Ramsauer et al., 2020] and Sparse Hopfield [Hu et al., 2023]. Note that the Random Masked Hopfield models perform poorly. This is because they violate the $\mu \in \mathcal{M}$ assumption in Lemma 3.2, as the random mask might inadvertently mask out the correct pattern in the memory set. (**Upper Right**): Training loss and accuracy curve of different Hopfield models on MNIST multiple instance learning task (Theorem 4.1). (**Bottom Right**): Validation loss and accuracy curve of different Hopfield models on MNIST multiple instance learning task (Theorem 4.1). We train all the model with 150 epochs with cosine annealing learning rate decay. Plotted are the means over 10 runs. We observe that with Sparse Hopfield having the highest validation accuracy, Random feature Hopfield also shows competitive performance with faster convergence speed. On the other hand, Top 20% Hopfield also converges fast with almost no performance drop. More experimental details can be found in Appendix E.

1. The starting point $\mathbf{x}$ is inside $\mathcal{S}_\mu$: $\mathbf{x} \in \mathcal{S}_\mu$.
2. The *well-separation* condition:

$$\Delta_\mu \geq \frac{1}{\beta} \ln\left(\frac{(M+k-2)m}{R}\right) + 2mR. \tag{4.3}$$

Proof. See Appendix B.7 for a detailed proof. □

Intuitively, the well-separation condition establishes a threshold that ensures any pattern $\{\boldsymbol{\xi}_\mu\}_{\mu\in\mathcal{M}}$ is distinguishable from all others, enabling patterns to be well-stored at a fixed point of $\mathcal{T}_{\text{Sparse}}$ and retrieved with R precision by $\mathcal{T}_{\text{Sparse}}$. Notably, Theorem 4.2 reveals that the lower bound on Δ_μ diminishes as k decreases. Consequently, as $\mathcal{M}$ becomes sparser, satisfying the well-separation condition becomes easier, facilitating the storage of patterns and leading to a larger memory capacity lower bound for sparse-structured modern Hopfield models.

Memory Capacity. To andress (B), we derive the lower bound for the maximum number of memory patterns that are well-stored and retrievable according to Theorem 4.2:

Lemma 4.2 (Modified from [Hu et al., 2023])**.** Define the probability of storing and retrieving a memory pattern as $1-p$. Memory capacity, the maximum number of patterns randomly sampled from a sphere with radius m that the sparse modern Hopfield models can store and retrieve, has an lower bound: $M_{\text{Sparse}} \geq \sqrt{p}C^{\frac{d-1}{4}}$, where C is the solution for the identity $C = {}^{b}/_{W_0(\exp\{a+\ln b\})}$ with the principal branch of Lambert W function, $a := ({}^{4}/_{d-1})\left(\ln\left[{}^{m(\sqrt{p}+k-1)}/_{R}\right]+1\right)$ and $b := {}^{4m^2\beta}/_{5(d-1)}$.

Proof. See Appendix B.8 for a detailed proof. □

Remark 4.3. Theorem 4.2 gives a memory capacity exponential in the pattern size d (maximum allowed value k). Since $k \leq M$, the scaling behavior of sparse-structured modern Hopfield models is similar to that of [Ramsauer et al., 2020, Hu et al., 2023]. This result mirrors findings in [Ramsauer et al., 2020, Hu et al., 2023].

5 Conclusion and Discussion

We introduce a nonparametric framework for modern Hopfield models. With Theorem 3.1, we replicate the known results of the original modern Hopfield model [Ramsauer et al., 2020]. With Lemma 3.2, we introduce the efficient sparse-structured Hopfield models with robust theoretical properties: tighter retrieval error bound (Corollary 4.1.1 & Corollary 4.1.2), stronger noise robustness (Remark 4.2) and exponential-d-capacity (Theorem 4.2 & Lemma 4.2).

Comparing with Existing Works. Our framework complements [Hu et al., 2023] by filling the efficiency gaps and connecting to various attentions in the following. Notably, when the size of the support set $k = M$, the results of Theorem 4.1, Theorem 4.2 and Lemma 4.2 reduce to those of the dense modern Hopfield model in [Ramsauer et al., 2020]. **Extensions.** In Appendix C, we present a family of modern Hopfield models connecting to many other existing attention mechanisms, including Linear [Katharopoulos et al., 2020], Multi-Head [Vaswani et al., 2017], and Generalized Kernelizable or PRFs (Positive Random Features) [Choromanski et al., 2021] modern Hopfield models. **Hopfield Layers and Numerical Experiments.** In line with [Hu et al., 2023, Ramsauer et al., 2020], we introduce deep learning layers as competitive attention alternatives with memory-enhanced functionalities, corresponding to our nonparametric modern Hopfield models (sparse-structured and above extensions) in Appendix D and verify them numerically in Figure 1 and Appendix E.

References

Andreas Auer, Martin Gauch, Daniel Klotz, and Sepp Hochreiter. Conformal prediction for time series with modern hopfield networks. *arXiv preprint arXiv:2303.12783*, 2023.

Mariette Awad, Rahul Khanna, Mariette Awad, and Rahul Khanna. Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, pages 67–80, 2015.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

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.

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

Johann S Brauchart, Alexander B Reznikov, Edward B Saff, Ian H Sloan, Yu Guang Wang, and Robert S Womersley. Random point sets on the sphere—hole radii, covering, and separation. *Experimental Mathematics*, 27(1): 62–81, 2018. URL https://arxiv.org/abs/1512.07470.

Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

Degang Chen, Qiang He, and Xizhao Wang. The infinite polynomial kernel for support vector machine. In *Advanced Data Mining and Applications: First International Conference, ADMA 2005, Wuhan, China, July 22-24, 2005. Proceedings 1*, pages 267–275. Springer, 2005.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

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.

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*, 35:20450–20468, 2022. URL https://arxiv.org/abs/2110.11316.

Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. *arXiv preprint arXiv:2106.06899*, 2021.

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, 2023. URL https://arxiv.org/abs/2309.12673.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. URL https://arxiv.org/abs/1802.04712.

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.

Melih Kandemir, Chong Zhang, and Fred A Hamprecht. Empowering multiple instance histopathology cancer diagnosis by cell graphs. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part II 17*, pages 228–235. Springer, 2014.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

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*, 29, 2016.

Dmitry Krotov and John J. Hopfield. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2008.06996.

YC Lee, Gary Doolen, HH Chen, GZ Sun, Tom Maxwell, and HY Lee. Machine learning using a higher order correlation network. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States); Univ. of Maryland, College Park, MD (United States), 1986.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016. URL https://arxiv.org/abs/1602.02068.

Charles M Newman. Memory capacity in neural network models: Rigorous lower bounds. *Neural Networks*, 1(3): 223–238, 1988.

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*, pages 17156–17185. PMLR, 2022.

Pierre Peretto and Jean-Jacques Niez. Long term memory storage capacity of multiconnected neural networks. *Biological Cybernetics*, 54(1):53–63, 1986.

Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*, 2019.

Hubert Ramsauer, Bernhard Schafl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlovic, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

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*, 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.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, 33:18832–18845, 2020.

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.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Willard I Zangwill. *Nonlinear programming: a unified approach*, volume 52. Prentice-Hall Englewood Cliffs, NJ, 1969.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**No**]
   Authors: We will make our code publicly available for reproducibility once the paper is accepted.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**]
   (b) Complete proofs of all theoretical results. [**Yes**]
   (c) Clear explanations of any assumptions. [**Yes**]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**No**]
   Authors: We will make our code publicly available for reproducibility once the paper is accepted.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**]
   (b) The license information of the assets, if applicable. [**Yes**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**No**]
   Authors: We will make our code publicly available for reproducibility once the paper is accepted.
   (d) Information about consent from data providers/curators. [**Yes**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Yes**]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Yes**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Yes**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Yes**]

Supplementary Material

A Table of Notations

Table 1: Mathematical Notations and Symbols

Symbol	Description
$\mathbf{a}[i]$	The i-th component of vector $\mathbf{a}$
$\langle \mathbf{a}, \mathbf{b} \rangle$	Inner product for vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$
$[I]$	Index set $\{1, \cdots, I\}$, where $I \in \mathbb{N}^+$
$\|\cdot\|$	Spectral norm, equivalent to the l_2-norm when applied to a vector
d	Dimension of patterns
M	Number of stored memory patterns
β	Scaling factor of the energy function controlling the learning dynamics. We set $\beta = 1/\sqrt{d}$ in practice
$\mathbf{x}$	State/configuration/query pattern in $\mathbb{R}^d$
$\mathbf{x}^\star$	Stationary points of the Hopfield energy function
$\boldsymbol{\xi}$	Memory patterns (keys) in $\mathbb{R}^d$
$\delta\boldsymbol{\xi}$	Noises in memory patterns in $\mathbb{R}^d$
$\mathcal{D}$	Training data set $\{(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu)\}_{\mu\in[M]}$
$\boldsymbol{\Xi}$	Shorthand for M stored memory (key) patterns $\{\boldsymbol{\xi}_\mu\}_{\mu\in[M]}$ in $\mathbb{R}^{d\times M}$
$\boldsymbol{\Xi}^\mathsf{T}\mathbf{x}$	M-dimensional overlap vector $(\langle\boldsymbol{\xi}_1, \mathbf{x}\rangle, \cdots, \langle\boldsymbol{\xi}_\mu, \mathbf{x}\rangle, \cdots, \langle\boldsymbol{\xi}_M, \mathbf{x}\rangle)$ in $\mathbb{R}^M$
$\Phi(\cdot)$	Kernelized feature mapping $\Phi(\cdot) : \mathbb{R}^d \to D_\phi$
ϕ	Element in the $\Phi(\cdot) = (\phi_0^{(0)}, \phi_1^{(1)}, \ldots, \phi_{D_1}^{(1)}, \ldots, \phi_1^{(n)}, \ldots, \phi_{D_n}^{(n)}, \ldots)$
D_Φ	Dimension of the kernel space, i.e., dimension of output of $\Phi(\cdot)$
$h(\cdot)$	Normalization mapping in the regression model defined by Definition 3.2
$\mathbf{W}$	Weighted matrix in the regression model defined by Definition 3.2 in $\mathbb{R}^{d\times D_\Phi}$
$\mathbf{w}_i$	i-th row of the weighted matrix $\mathbf{W}$ in $\mathbb{R}^{D_\Phi}$
$\mathcal{K}(\cdot,\cdot)$	Kernel function takes the inner product form $\mathcal{K}(\cdot,\cdot) = \langle\Phi(\cdot), \Phi(\cdot)\rangle$ in $\mathcal{K} : \mathbb{R}^{D_\Phi} \times \mathbb{R}^{D_\Phi} \to \mathbb{R}_+$
ϵ'	Component-wise term error margin in the support vector regression problem
$\boldsymbol{\eta}, \widetilde{\boldsymbol{\eta}}$	Slack variables in the support vector regression
R'	Largest sum of the error margin and slack variable, i.e., $R' \coloneqq \mathrm{Max}_{i\in[d]}\{\epsilon' + \boldsymbol{\eta}_\mu[i]\} \le \epsilon/\sqrt{M}$
C	Penalized coefficient of the support vector regression
$\mathcal{L}$	Lagrangian corresponding to (3.2)
$\boldsymbol{\alpha}, \widetilde{\boldsymbol{\alpha}}, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\lambda}}$	Dual variables in the Lagrangian defined by (3.3)
$\mathcal{M}$	Reduced support set for $\mathcal{T}_{\text{SVR}}$ $\mathcal{M} \coloneqq \{\mathcal{M}(1), \ldots, \mathcal{M}(k)\} \subseteq \{1, \ldots, M\}$
$\mathbb{1}_{\mathcal{M}(\mu)}$	Indicator function corresponding to $\mathcal{M}$, where $\mathbb{1}_{\mathcal{M}(\mu)} = 1$ for $\mu \in \mathcal{M}$ and $\mathbb{1}_{\mathcal{M}(\mu)} = 0$ for $\mu \notin \mathcal{M}$
k	Size of the support set $\mathcal{M}$, defined as $k \coloneqq \lvert\mathcal{M}\rvert$
n	Norm of $\mathbf{x}$, denoted as $n \coloneqq \|\mathbf{x}\|$
m	Largest norm of memory patterns, denoted as $m \coloneqq \mathrm{Max}_{\mu\in[M]}\|\boldsymbol{\xi}_\mu\|$
R	Minimal Euclidean distance across all possible pairs of memory patterns, denoted as $R \coloneqq \frac{1}{2}\mathrm{Min}_{\mu,\nu\in[M]}\|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$
S_μ	Sphere centered at memory pattern $\boldsymbol{\xi}_\mu$ with finite radius R
$\mathbf{x}_\mu^\star$	Fixed point of $\mathcal{T}$ covered by S_μ, i.e., $\mathbf{x}_\mu^\star \in S_\mu$
Δ_μ	Separation of a memory pattern $\boldsymbol{\xi}_\mu$ from all other memory patterns $\boldsymbol{\Xi}$, defined in (4.1)
$\widetilde{\Delta}_\mu$	Separation of $\boldsymbol{\xi}_\mu$ at a given $\mathbf{x}$ from all memory patterns $\boldsymbol{\Xi}$, defined in (4.1)

B Proofs of Main Text

B.1 Theorem 3.1

Proof of Theorem 3.1. To obtain the solution of $\mathbf{W}$, the partial derivatives of $\mathcal{L}$ with respect to $\mathbf{w}_i, \boldsymbol{\eta}_\mu[i]$ and $\widetilde{\boldsymbol{\eta}}_\mu[i]$ must satisfy the stationarity condition, and thus we have:

$$\begin{cases} \mathbf{w}_i - \sum_{\mu=1}^M \frac{\boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}\Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu) = 0, \\ C - \boldsymbol{\lambda}_\mu[i] - \boldsymbol{\alpha}_\mu[i] = 0, \\ C - \widetilde{\boldsymbol{\lambda}}_\mu[i] - \widetilde{\boldsymbol{\lambda}}_\mu[i] = 0, \end{cases} \tag{B.1}$$

as the necessary conditions for the optimum of (3.2).

Inserting this result into Definition 3.2, we obtain

$$\mathbf{x}^{\text{new}}[i] = \mathcal{T}_{\text{SVR}}(\mathbf{x})[i] \coloneqq \left\langle \mathbf{w}_i^\star, \frac{\Phi(\mathbf{x})}{h(\mathbf{x})} \right\rangle, \tag{B.2}$$

where $\mathbf{w}_i^\star \in \mathbb{R}^{D_\Phi}$ is the i-th row of the learned weight, $\mathbf{W}$,

$$\mathbf{w}^\star[i] \coloneqq \sum_{\mu=1}^{M} \underbrace{\frac{\boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}}_{\in\mathbb{R}} \underbrace{\Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}_{\in\mathbb{R}^{D_\Phi}}. \tag{B.3}$$

In addition, by complementary slackness condition and dual feasibility, we have

$$\begin{cases} \boldsymbol{\alpha}_\mu[i] \left(\epsilon' + \boldsymbol{\eta}_\mu[i] - \boldsymbol{\xi}_\mu[i] + \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)} \right\rangle \right) = 0 \\ \widetilde{\boldsymbol{\alpha}}_\mu[i] \left(\epsilon' + \widetilde{\boldsymbol{\eta}}_\mu[i] - \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)} \right\rangle + \boldsymbol{\xi}_\mu[i] \right) = 0 \\ \boldsymbol{\alpha}_\mu[i], \widetilde{\boldsymbol{\alpha}}_\mu[i], \boldsymbol{\lambda}_\mu[i], \widetilde{\boldsymbol{\lambda}}_\mu[i] \geq 0, \end{cases} \tag{B.4}$$

for all $\mu \in [M]$ and $i \in [d]$. □

B.2 Lemma 3.1

In order to prove Lemma 3.1, first we introduce the following three auxiliary lemmas.

Lemma B.1. Following the setting in Theorem 3.1, if the optimaltiy of (3.2) occurs, the inequality

$$-C \leq \boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i] \leq C \tag{B.5}$$

holds for all $\mu \in [M]$ and $i \in [d]$.

Proof. We prove this by contradiction. From (B.4), we know the dual variables $\boldsymbol{\alpha}_\mu[i], \widetilde{\boldsymbol{\alpha}}_\mu[i] \geq 0$. Firstly, we assume $\boldsymbol{\alpha}_\mu[i]$ and $\widetilde{\boldsymbol{\alpha}}_\mu[i]$ are *positive*, for all $\mu \in [M]$ and $i \in [d]$. Subsequently, by complementary slackness condition (B.4), we have

$$\begin{cases} \epsilon' + \boldsymbol{\eta}_\mu[i] - \boldsymbol{\xi}_\mu[i] + \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)} \right\rangle = 0 \\ \epsilon' + \widetilde{\boldsymbol{\eta}}_\mu[i] - \left\langle \mathbf{w}_i, \frac{\Phi(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)}{h(\boldsymbol{\xi}_\mu+\delta\boldsymbol{\xi}_\mu)} \right\rangle + \boldsymbol{\xi}_\mu[i] = 0. \end{cases} \tag{B.6}$$

Combining the above, since the error $\epsilon' \geq 0$, we have

$$\boldsymbol{\eta}_\mu[i] + \widetilde{\boldsymbol{\eta}}_\mu[i] = -2\epsilon' \leq 0, \tag{B.7}$$

which implies that the sum of dual variables $\boldsymbol{\eta}_\mu[i]$ and $\widetilde{\boldsymbol{\eta}}_\mu[i]$ is negative. This contradicts the assumption of the non-negative condition on slack variables $\boldsymbol{\eta}_\mu[i], \widetilde{\boldsymbol{\eta}}_\mu[i] \geq 0$. Therefore, together with (B.4), at least one of $\boldsymbol{\alpha}_\mu[i]$,$\widetilde{\boldsymbol{\alpha}}_\mu[i]$ must be 0, for all μ and all i. Subsequently, we have

$$0 \leq \boldsymbol{\alpha}_\mu[i] \leq C \quad \text{and} \quad 0 \leq \widetilde{\boldsymbol{\alpha}}_\mu[i] \leq C, \tag{B.8}$$

which leads to

$$-C \leq \boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i] \leq C. \tag{B.9}$$

□

Lemma B.2 (Multinomial Expansion)**.** Given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. The identity

$$\frac{(\mathbf{x}^\mathsf{T}\mathbf{y})^n}{n!} = \sum_{\ell_1+\cdots+\ell_d=n} \left(\frac{x_1^{\ell_1}\cdots x_d^{\ell_d}}{\sqrt{\ell_1!\cdots\ell_d!}}\right)\left(\frac{y_1^{\ell_1}\cdots y_d^{\ell_d}}{\sqrt{\ell_1!\cdots\ell_d!}}\right)$$

holds for all $n \in \mathbb{N}$.

Proof.

$$\begin{aligned}
\frac{(\mathbf{x}^\mathsf{T}\mathbf{y})^n}{n\,!} &= \frac{1}{n\,!}(x_1y_1+\cdots+x_dy_d)^n \\
&= \frac{1}{n\,!}\left[(x_1y_1)^n+\cdots+\frac{n!}{\ell_1!\cdots\ell_d!}\prod_{i=1}^d(x_iy_i)^{\ell_i}+\cdots+(x_dy_d)^n\right] && \left(\textstyle\sum_{i=1}^d \ell_i = n\right) \\
&= \sum_{\ell_1+\cdots+\ell_d=n} \frac{1}{\ell_1!\cdots\ell_d!}\prod_{i=1}^d(x_i)^{\ell_i}\prod_{i=1}^d(y_i)^{\ell_i} \\
&= \sum_{\ell_1+\cdots+\ell_d=n} \frac{\left(x_1^{\ell_1}\cdots x_d^{\ell_d}\right)\left(y_1^{\ell_1}\cdots y_d^{\ell_d}\right)}{\ell_1!\cdots\ell_d!} \\
&= \sum_{\ell_1+\cdots+\ell_d=n} \left(\frac{x_1^{\ell_1}\cdots x_d^{\ell_d}}{\sqrt{\ell_1!\cdots\ell_d!}}\right)\left(\frac{y_1^{\ell_1}\cdots y_d^{\ell_d}}{\sqrt{\ell_1!\cdots\ell_d!}}\right).
\end{aligned}$$

□

Lemma B.3. Let $\mathcal{K}(\cdot,\cdot)$ be the homogeneous infinite polynomial kernel [Chen et al., 2005]:

$$\mathcal{K}(\mathbf{x},\mathbf{y}) = \langle\Phi(\mathbf{x}),\Phi(\mathbf{y})\rangle \coloneqq \sum_{n=0}^\infty \frac{(\mathbf{x}^\mathsf{T}\mathbf{y})^n}{n!}, \tag{B.10}$$

where $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$ and Φ maps the feature vectors $\mathbf{x}$ and $\mathbf{y}$ into infinite dimensional space. Then, $\Phi(\cdot) = (\phi_0^{(0)}, \phi_1^{(1)}, \ldots, \phi_{D_1}^{(1)}, \ldots, \phi_1^{(n)}, \ldots, \phi_{D_n}^{(n)}, \ldots)$ has a closed form solution

$$\phi_{D'}^{(n)} = \frac{x_1^{\ell_1}\cdots x_d^{\ell_d}}{\sqrt{\ell_1\,!\cdots\ell_d\,!}}, \tag{B.11}$$

where $\ell_1+\cdots+\ell_d = n$, $1 \le D' \le D_n$ and $D_n \coloneqq \binom{d+n-1}{n}$.

Proof. Applying multinomial expansion on homogeneous infinite polynomial kernel, we have

$$\begin{aligned}
\langle\Phi(\mathbf{x}),\Phi(\mathbf{y})\rangle &= \sum_{n=0}^\infty \frac{(\mathbf{x}^\mathsf{T}\mathbf{y})^n}{n!} \\
&= \sum_{n=0}^\infty \sum_{\ell_1+\cdots+\ell_d=n} \frac{1}{\ell_1!\cdots\ell_d!}\prod_{i=1}^d(x_i)^{\ell_i}\prod_{i=1}^d(y_i)^{\ell_i} \\
&= \sum_{n=0}^\infty \sum_{\ell_1+\cdots+\ell_d=n} \frac{\left(x_1^{\ell_1}\cdots x_d^{\ell_d}\right)\left(y_1^{\ell_1}\cdots y_d^{\ell_d}\right)}{n_1!\cdots n_d!} \\
&= \sum_{n=0}^\infty \sum_{\ell_1+\cdots+\ell_d=n} \left(\frac{x_1^{\ell_1}\cdots x_d^{\ell_d}}{\sqrt{\ell_1!\cdots\ell_d!}}\right)\left(\frac{y_1^{\ell_1}\cdots y_d^{\ell_d}}{\sqrt{\ell_1!\cdots\ell_d!}}\right).
\end{aligned}$$

From above, we observe that, for each fixed n, there are $\binom{d+n-1}{n}$ terms in the summation. Consequently, $\Phi(\mathbf{x})$ has a solution

$$\Phi(\mathbf{x}) = (\phi_0^{(0)}, \underbrace{\phi_1^{(1)}, \ldots, \phi_{D_1}^{(1)}}_{\binom{d+1-1}{1} \text{ elements}}, \ldots, \underbrace{\phi_1^{(n)}, \ldots, \phi_{D_n}^{(n)}}_{\binom{d+n-1}{n} \text{ elements}}, \ldots), \tag{B.12}$$

where $D_n = \binom{d+n-1}{n}$ and

$$\phi_{D'}^{(n)} = \frac{x_1^{\ell_1} \cdots x_d^{\ell_d}}{\sqrt{\ell_1 ! \cdots \ell_d !}} \tag{B.13}$$

for $1 \leq D' \leq D_n$ and $\ell_1 + \cdots + \ell_d = n$. □

Proof of Lemma 3.1. By Theorem 3.1, the component of the learned weight matrix $\mathbf{W}$ is

$$\mathbf{w}_i^\star = \sum_{\mu=1}^{M} \frac{\boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)} \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \tag{B.14}$$

where $\mathbf{w}_i^\star$ denote i-th component of learned matrix.

Inserting $\mathbf{w}^\star$ into the regression model (3.1), we have

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \left[\sum_{\mu=1}^{M} \frac{\boldsymbol{\alpha}_\mu[1] - \widetilde{\boldsymbol{\alpha}}_\mu[1]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)} \frac{\langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle}{h(\mathbf{x})}, \cdots, \sum_{\mu=1}^{M} \frac{\boldsymbol{\alpha}_\mu[d] - \widetilde{\boldsymbol{\alpha}}_\mu[d]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)} \frac{\langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x} + \delta\boldsymbol{\xi}_\mu) \rangle}{h(\mathbf{x})} \right]. \tag{B.15}$$

Suppose that $\mathbf{v}_\mu = \left[\frac{\boldsymbol{\alpha}_\mu[1] - \widetilde{\boldsymbol{\alpha}}_\mu[1]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}, \ldots, \frac{\boldsymbol{\alpha}_\mu[d] - \widetilde{\boldsymbol{\alpha}}_\mu[d]}{h(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)} \right]$ and $h(\mathbf{x}) \coloneqq \sum_{\mu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle$. Then $\mathcal{T}_{\text{Dense}}$ becomes

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \sum_{\mu=1}^{M} \frac{\langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle}{h(\mathbf{x})} \mathbf{v}_\mu = \sum_{\mu=1}^{M} \frac{\langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle}{\sum_{\nu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\nu + \delta\boldsymbol{\xi}_\nu), \Phi(\mathbf{x}) \rangle} \boldsymbol{\xi}_\mu. \tag{B.16}$$

Following Lemma B.3, here we define the inner product of Φ as a kernel $\mathcal{K} : \mathbb{R}^{D_\phi} \times \mathbb{R}^{D_\phi} \to \mathbb{R}_+$

$$\langle \Phi(\mathbf{x}), \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu) \rangle \coloneqq \mathcal{K}(\mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \tag{B.17}$$

then we rewrite $\mathcal{T}_{\text{Dense}}$ as:

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \sum_{\mu=1}^{M} \frac{\mathcal{K}(\mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)}{\sum_{\nu=1}^{M} \mathcal{K}(\mathbf{x}, \boldsymbol{\xi}_\nu + \delta\boldsymbol{\xi}_\nu)} \boldsymbol{\xi}_\mu. \tag{B.18}$$

To derive the softmax function in (2.2), we need a kernel function $\mathcal{K}$ such that $\mathcal{T}_{\text{Dense}}$ becomes Boltzmann form: $\exp\{\cdot\} / \sum_{\nu=1}^{M} \exp\{\cdot\}$. By Lemma B.3, we identify

$$\phi_{D'}^{(n)} = \frac{(\sqrt{\beta} x_1)^{\ell_1} \cdots (\sqrt{\beta} x_d)^{\ell_d}}{\sqrt{\ell_1 ! \cdots \ell_d !}}, \tag{B.19}$$

as the component of Taylor expansion of the $\exp$ function. Therefore, we choose

$$\mathcal{K}(\mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu) = \sum_{n=0}^{\infty} \frac{(\langle \sqrt{\beta}\mathbf{x}, \sqrt{\beta}\boldsymbol{\xi}_\mu + \sqrt{\beta}\delta\boldsymbol{\xi}_\mu \rangle)^n}{n!}. \tag{B.20}$$

Inserting (B.20) into (B.18), we obtain

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \sum_{\mu=1}^{M} \frac{\sum_{n=0}^{\infty} \left(\left\langle \sqrt{\beta}\mathbf{x}, \sqrt{\beta}\boldsymbol{\xi}_\mu + \sqrt{\beta}\delta\boldsymbol{\xi}_\mu \right\rangle\right)^n / n!}{\sum_{\nu=1}^{M} \sum_{\bar{n}=0}^{\infty} \left(\left\langle \sqrt{\beta}\mathbf{x}, \sqrt{\beta}\boldsymbol{\xi}_\nu + \sqrt{\beta}\delta\boldsymbol{\xi}_\nu \right\rangle\right)^{\bar{n}} / \bar{n}!} \boldsymbol{\xi}_\mu. \tag{B.21}$$

By Taylor's theorem, $\mathcal{T}_{\text{Dense}}$ takes the form

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \sum_{\mu=1}^{M} \frac{\exp\{\beta \langle \mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu \rangle\}}{\sum_{\nu=1}^{M} \exp\{\beta \langle \mathbf{x}, \boldsymbol{\xi}_\nu + \delta\boldsymbol{\xi}_\nu \rangle\}} \boldsymbol{\xi}_\mu, \tag{B.22}$$

and, therefore, we obtain

$$\mathcal{T}_{\text{Dense}}(\mathbf{x}) = \sum_{\mu=1}^{M} \text{Softmax}(\beta \langle \mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu \rangle) \boldsymbol{\xi}_\mu. \tag{B.23}$$

□

B.3 Lemma 3.2

Proof of Lemma 3.2. To obtain $\mathbf{w}^\star$ for the sparse-structured model, the partial derivatives of $\mathcal{L}$ (the Lagrangian (3.3)) with respect to $\mathbf{w}_i, \boldsymbol{\eta}_\mu[i]$ and $\widetilde{\boldsymbol{\eta}}_\mu[i]$ must satisfy the stationarity condition, and thus we have:

$$\begin{cases} \mathbf{w}_i - \sum_{\mu=1}^{M} \frac{\boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i]}{h(\boldsymbol{\xi}_\mu)} \Phi(\boldsymbol{\xi}_\mu) \mathbb{1}_{\mathcal{M}(\mu)} = 0, \\ C - \boldsymbol{\lambda}_\mu[i] - \boldsymbol{\alpha}_\mu[i] = 0, \\ C - \widetilde{\boldsymbol{\lambda}}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i] = 0, \end{cases} \tag{B.24}$$

where $\mathbb{1}_{\mathcal{M}(\mu)} = 1$ for $\mu \in \mathcal{M}$ and $\mathbb{1}_{\mathcal{M}(\mu)} = 0$ for $\mu \notin \mathcal{M}$. Then, we arrive

$$\mathbf{w}_i^\star = \sum_{\mu=1}^{M} \frac{\boldsymbol{\alpha}_\mu[i] - \widetilde{\boldsymbol{\alpha}}_\mu[i]}{h(\boldsymbol{\xi}_\mu)} \Phi(\boldsymbol{\xi}_\mu) \mathbb{1}_{\mathcal{M}(\mu)} \tag{B.25}$$

By Appendix B.2, we obtain the retrieval dynamics for sparse-structured modern Hopfield model:

$$\mathcal{T}_{\text{Sparse}}(\mathbf{x}) = \sum_{\mu=1}^{M} \text{Softmax}\left(\beta \langle \mathbf{x}, \boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu \rangle\right) \boldsymbol{\xi}_\mu \mathbb{1}_{\mathcal{M}(\mu)}. \tag{B.26}$$

□

B.4 Theorem 4.1

Proof of Theorem 4.1. To connect $\mathcal{T}_{\text{Sparse}}$ with Δ_μ, first we derive the bound on $\|\mathcal{T}_{\text{Sparse}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|$ via [Ramsauer et al., 2020]

$$\begin{aligned} \|\mathcal{T}_{\text{Sparse}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| &\leq \left\| \boldsymbol{\xi}_\mu - \sum_{\nu \in \mathcal{M}} \text{Softmax}\left(\beta \langle \mathbf{x}, \boldsymbol{\xi} \rangle\right) \boldsymbol{\xi}_\nu \right\| \\ &\leq \|(1 - [\text{Softmax}\left(\beta \langle \mathbf{x}, \boldsymbol{\xi}_\nu \rangle\right)]_\mu) \boldsymbol{\xi}_\mu\| + \left\| \sum_{\nu \in \mathcal{M}} \text{Softmax}\left(\beta \langle \mathbf{x}, \boldsymbol{\xi}_\nu \rangle\right) \boldsymbol{\xi}_\nu \right\| - \|[\text{Softmax}\left(\beta \langle \mathbf{x}, \boldsymbol{\xi}_\nu \rangle\right)]_\mu \boldsymbol{\xi}_\mu\| \\ &\leq m \frac{M + k - 2}{M - 1} \widetilde{\epsilon} \qquad \text{(B.27)} \\ &= m(M + k - 2) \exp\left\{ -\beta \left(\langle \boldsymbol{\xi}_\mu, \mathbf{x} \rangle - \max_{\nu \in [M]} \langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle \right) \right\}, \qquad \text{(B.28)} \end{aligned}$$

where $k \coloneqq |\mathcal{M}|$, $m \coloneqq \operatorname{Max}_\mu \|\boldsymbol{\xi}_\mu\|$, $\widetilde{\epsilon} \coloneqq (M-1)\exp\{-\beta\left(\langle \boldsymbol{\xi}_\mu, \mathbf{x}\rangle - \operatorname{Max}_{\nu\in[M]}\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu\rangle\right)\}$ and the inequality

$$\left[\operatorname{Softmax}(\beta \boldsymbol{\Xi}^\mathsf{T}\mathbf{x})\right]_\nu = \frac{\exp\{\beta\left(\langle \mathbf{x}, \boldsymbol{\xi}_\nu\rangle - \langle \mathbf{x}, \boldsymbol{\xi}_\mu\rangle\right)\}}{1+\sum_{\nu'\neq\mu}\exp\{\beta\left(\langle \mathbf{x}, \boldsymbol{\xi}_{\nu'}\rangle - \langle \mathbf{x}, \boldsymbol{\xi}_\mu\rangle\right)\}} \leq \exp\left\{-\beta\left(\langle \boldsymbol{\xi}_\mu, \mathbf{x}\rangle - \operatorname*{Max}_{\nu\in[M]}\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu\rangle\right)\right\}, \tag{B.29}$$

is used in (B.28). □

B.5 Corollary 4.1.1 and Corollary 4.1.2

Proof of Corollary 4.1.1 and Corollary 4.1.2. Since the support set of *dense* modern Hopfield model is full, i.e. $k = M$, (B.28) reduces to

$$\|\mathcal{T}_{\text{Dense}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq 2m(M-1)\exp\left\{-\beta\left(\langle \boldsymbol{\xi}_\mu, \mathbf{x}\rangle - \operatorname*{Max}_{\nu\in[M]}\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu\rangle\right)\right\}. \tag{B.30}$$

Comparing (B.28) with (B.30), we obtain

$$\|\mathcal{T}_{\text{Sparse}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \|\mathcal{T}_{\text{Dense}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|. \tag{B.31}$$

From [Ramsauer et al., 2020, Theorem 4], for any query $\mathbf{x}$, $\mathcal{T}_{\text{Dense}}$ approximately retrieves a memory pattern $\boldsymbol{\xi}_\mu$ with retrieval error ϵ exponentially suppressed by Δ_μ:

$$\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq 2m(M-1)\exp\{-\beta\left(\Delta_\mu - 2m\operatorname{Max}\left[\|\mathbf{x}-\boldsymbol{\xi}_\mu\|, \left\|\mathbf{x}-\mathbf{x}_\mu^\star\right\|\right]\right)\}. \tag{B.32}$$

By (B.31), $\mathcal{T}_{\text{Sparse}}$ also enjoys above retrieval error bound. Therefore, $\mathcal{T}_{\text{Sparse}}(\mathbf{x})$ retrieves a memory pattern $\boldsymbol{\xi}_\mu$ with high accuracy after a single activation with a sufficiently large Δ_μ. □

B.6 Lemma 4.1

Proof of Lemma 4.1. By [Hu et al., 2023], we know that

$$\lim_{t\to\infty}\|\mathbf{x}_t - \boldsymbol{\xi}_\mu\| = 0, \text{ for initial query } \mathbf{x}_0 \in S_\mu, \tag{B.33}$$

where $\{\mathbf{x}_t\}_{t=0}^\infty$ is a sequence generated by $\mathcal{T}_{\text{Dense}}$ from $\mathbf{x}_0$, i.e. $\mathcal{T}_{\text{Dense}}(\mathbf{x}_t) = \mathbf{x}_{t+1}$.

Since the inequality

$$0 \leq \|\mathcal{T}_{\text{Sparse}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \|\mathcal{T}_{\text{Dense}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\|, \tag{B.34}$$

holds for any query pattern $\mathbf{x} \in S_\mu$, we obtain

$$\lim_{t\to\infty}\|\widetilde{\mathbf{x}}_t - \boldsymbol{\xi}_\mu\| = 0, \tag{B.35}$$

where $\{\widetilde{\mathbf{x}}_t\}_{t=0}^\infty$ is a sequence generated by $\mathcal{T}_{\text{Sparse}}$, i.e. $\mathcal{T}_{\text{Sparse}}(\widetilde{\mathbf{x}}_t) = \widetilde{\mathbf{x}}_{t+1}$, by applying squeeze theorem on (B.34) and (B.33).

□

B.7 Theorem 4.2

Proof of 4.2. By the Cauchy-Schwartz inequality, for all $\mu \in \mathcal{M}$,

$$|\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu\rangle - \langle \mathbf{x}, \boldsymbol{\xi}_\mu\rangle| \leq \|\boldsymbol{\xi}_\mu - \mathbf{x}\| \cdot \|\boldsymbol{\xi}_\mu\| \leq \|\boldsymbol{\xi}_\mu - \mathbf{x}\| m, \tag{B.36}$$

we find that $\widetilde{\Delta}_\mu$ can be formulated in terms of Δ_μ:

$$\widetilde{\Delta}_\mu = \Delta_\mu - 2\|\boldsymbol{\xi}_\mu - \mathbf{x}\|m = \Delta_\mu - 2mR, \qquad (\text{By } \mathbf{x} \in S_\mu)$$

where R is radius of the sphere S_μ. Therefore, for $\mathcal{T}$ to be a mapping $\mathcal{T}: S_\mu \to S_\mu$, it is sufficient to obtain

$$R \geq (M + k - 2) \exp\{-\beta(\Delta_\mu - 2mR)\}m, \tag{B.37}$$

which implies that

$$\Delta_\mu \geq \frac{1}{\beta} \ln\left(\frac{(M + k - 2)m}{R}\right) + 2mR. \tag{B.38}$$

□

B.8 Lemma 4.2

We built our proof on top of [Hu et al., 2023, Lemma 2.1], which consists 3 steps:

- **(Step 1.)** We establish a more refined well-separation condition, ensuring that patterns $\{\boldsymbol{\xi}_\mu\}_{\mu\in[M]}$ are well-stored in $\mathcal{H}$ and can be retrieved by $\mathcal{T}$ with an error ϵ at most R.
- **(Step 2.)** This condition is then related to the cosine similarity of memory patterns, from which we deduce an inequality governing the probability of successful pattern storage and retrieval.
- **(Step 3.)** We pinpoint the conditions for exponential memory capacity and confirm their satisfaction.

We start with an auxiliary lemma.

Lemma B.4 ([Hu et al., 2023, Ramsauer et al., 2020])**.** With the given equation $ac + c\ln c - b = 0$, we have

$$ac + c\ln c - b = 0, \tag{B.39}$$

holds, then the solution is

$$c = \frac{b}{W_0(\exp(a + \ln b))}. \tag{B.40}$$

Proof. We restate the proof of [Hu et al., 2023, Lemam 3.1] here for completeness. We rearrange (B.39) and solve for c as follows:

$$\begin{aligned}
ac + c\ln c - b &= 0,\\
a + \ln c &= \frac{b}{c},\\
\frac{b}{c} + \ln\left(\frac{b}{c}\right) &= a + \ln b,\\
\frac{b}{c}\exp\left(\frac{b}{c}\right) &= \exp(a + \ln b),\\
\frac{b}{c} &= W_0(\exp(a + \ln b)),\\
c &= \frac{b}{W_0(\exp(a + \ln b))}.
\end{aligned}$$

This completes the proof. □

Proof of Lemma 4.2. Our proof is built on top of [Hu et al., 2023, Corollary 3.1.1] with a different well-separation condition.

Let $\Delta_{\min} \coloneqq \mathrm{Min}_{\mu\in[M]}\, \Delta_\mu$ and $\theta_{\mu\nu}$ Here we define $\Delta_{\min}$ and $\theta_{\mu\nu}$ be the angle between two patterns $\boldsymbol{\xi}^\mu$ and $\boldsymbol{\xi}^\nu$.

In order for a pattern $\boldsymbol{\xi}_\mu$ to be well-stored, we need

$$\Delta_{\min} \geq \frac{1}{\beta} \ln\left(\frac{(M + k - 2)m}{R}\right) + 2mR, \tag{B.41}$$

by Theorem 4.2. On the other hand, we observe

$$\Delta_{\min} = \underset{1\le\mu\le\nu\le M}{\mathrm{Min}} \left[m^2\left(1-\cos(\theta_{\mu\nu})\right)\right] = m^2\left[1-\cos(\theta_{\min})\right], \tag{B.42}$$

where $\theta_{\min} \coloneqq \mathrm{Min}_{1\le\mu\le\nu\le M}\,\theta_{\mu\nu} \in [0,\pi]$. Then, we have

$$m^2\left[1-\cos(\theta_{\min})\right] \ge \frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR. \tag{B.43}$$

As a result, the probability of successful storage and retrieval, i.e., the minimal separation $\Delta_{\min}$ that satisfies Theorem 4.2, is given by

$$P\left(\Delta_\mu \ge \frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR\right) = 1-p. \tag{B.44}$$

Inserting (B.43) into above, we obtain

$$P\left(m^2\left[1-\cos(\theta_{\min})\right] \ge \frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR\right) = 1-p. \tag{B.45}$$

From [Olver et al., 2010, Equation (4.22.2)], for $0 \le \cos(\theta_{\min}) \le 1$, $\cos(\theta_{\min})$ has an upper bound

$$\cos(\theta_{\min}) \le 1 - \frac{\theta_{\min}^2}{5}. \tag{B.46}$$

It holds

$$P\left(\frac{m^2\theta_{\min}^2}{5} \ge \frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR\right) = 1-p, \tag{B.47}$$

which leads to

$$P\left(M^{\frac{2}{d-1}}\theta_{\min} \ge \frac{\sqrt{5}M^{\frac{2}{d-1}}}{m}\left[\frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR\right]^{\frac{1}{2}}\right) = 1-p. \tag{B.48}$$

For later convenience, here we introduce an extra $M^{2/d-1}$ on both sides.

Let $\omega_d \coloneqq \frac{2\pi^{d+1/2}}{\Gamma\left(\frac{d+1}{2}\right)}$ be the area of a d-dimensional unit sphere manifold, with $\Gamma(\cdot)$ denoting the gamma function.

Following [Brauchart et al., 2018, Lemma 3.5], we have

$$\begin{aligned} P\left(M^{\frac{2}{d-1}}\theta_{\min} \ge \frac{\sqrt{5}M^{\frac{2}{d-1}}}{m}\left[\frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR\right]^{\frac{1}{2}}\right) &= 1-p \\ \ge 1 - \frac{1}{2}\gamma_{d-1}5^{\frac{d-1}{2}}M^2m^{-(d-1)}&\left[\frac{1}{\beta}\ln\left(\frac{(M+k-2)m}{R}\right) + 2mR\right]^{\frac{d-1}{2}}, \end{aligned} \tag{B.49}$$

where γ_d is the ratio between the surface areas of the unit spheres in $(d-1)$ and d dimensions:

$$\gamma_d \coloneqq \frac{1}{d}\frac{\omega_{d-1}}{\omega_d} = \frac{1}{d\sqrt{\pi}}\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}. \tag{B.50}$$

Recall $d, M \in \mathbb{N}_+$, $p \in [0,1]$. Hence, it holds $M = \sqrt{p}C^{\frac{d-1}{4}}$ for some real values $C \in \mathbb{R}$.

Then, by (B.49), we have

$$5^{\frac{d-1}{2}}\left(\sqrt{p}C^{\frac{d-1}{4}}\right)^2 m^{-(d-1)}\left\{\frac{1}{\beta}\ln\left[\frac{\left(\sqrt{p}C^{\frac{d-1}{4}}+k-1\right)m}{R}\right]+\frac{1}{\beta}\right\}^{\frac{d-1}{2}}-p\leq 0, \tag{B.51}$$

and thus

$$5^{\frac{d-1}{2}}C^{\frac{d-1}{2}}m^{-(d-1)}\left\{\frac{1}{\beta}\ln\left[\frac{\left(\sqrt{p}C^{\frac{d-1}{4}}+k-1\right)m}{R}\right]+\frac{1}{\beta}\right\}^{\frac{d-1}{2}}\leq 1. \tag{B.52}$$

Further, we rewrite (B.52) as

$$\frac{5C}{m^2\beta}\left\{\ln\left[\frac{\left(\sqrt{p}C^{\frac{d-1}{4}}+k-1\right)m}{R}\right]+1\right\}-1\leq 0, \tag{B.53}$$

and identify

$$a\coloneqq\frac{4}{d-1}\left\{\ln\left[\frac{m(\sqrt{p}+k-1)}{R}\right]+1\right\},\quad b\coloneqq\frac{4m^2\beta}{5(d-1)}. \tag{B.54}$$

By Lemma B.4, C takes the form

$$C=\frac{b}{W_0(\exp\{a+\ln b\})}, \tag{B.55}$$

where $W_0(\cdot)$ is the upper branch of the Lambert W function. Since the domain of the Lambert W function is $x>(-1/e,\infty)$ and the fact $\exp\{a+\ln b\}>0$, the solution for (B.55) exists.

When the inequality (B.52) holds, the lower bound on the exponential storage capacity M can be written as:

$$M\geq\sqrt{p}C^{\frac{d-1}{4}}. \tag{B.56}$$

In particular, the above lower bound takes a form similar to [Ramsauer et al., 2020, Theorem 3]. □

C Nonparametric Modern Hopfield Family

In this section, we derive a family of modern Hopfield models as possible extensions based on the proposed framework (Theorem 3.1).

C.1 Linear Modern Hopfield Model

Proposition C.1 (Linear Modern Hopfield Model)**.** Let $h(\mathbf{x}) \coloneqq \sum_{\mu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle$ and $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_d(\mathbf{x}))$ with the component ϕ:

$$\phi_i(\mathbf{x}) \coloneqq \mathrm{elu}(\mathbf{x}[i]) + 1, \quad \forall i \in [d], \tag{C.1}$$

where $\mathrm{elu}(\cdot)$ denotes the exponential linear unit activation function proposed by [Clevert et al., 2015]. By Theorem 3.1, fitting $\mathcal{T}_{\text{SVR}}$ on $\mathcal{D}$ following (3.2) gives

$$\mathcal{T}_{\text{Linear}}(\mathbf{x}) = \frac{\sum_{\mu=1}^{M} \langle \Phi(\mathbf{x}), \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu) \rangle \boldsymbol{\xi}_\mu}{\sum_{\mu=1}^{M} \langle \Phi(\mathbf{x}), \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu) \rangle}. \tag{C.2}$$

By setting the kernel mapping Φ to linear feature map (C.1), we obtain a **linear modern Hopfield model** with linear complexity $\mathcal{O}(n)$. Compared with dense modern Hopfield model, our proposed linear modern Hopfield model has time and memory complexity $\mathcal{O}(n)$ instead of $\mathcal{O}(n^2)$ since we only need to compute $\sum_{\mu=1}^{M} \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)\boldsymbol{\xi}_\mu$ and $\sum_{\mu=1}^{M} \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu)$ once and reuse them for the computation of every query pattern. This model is by design connected to the random attention of linear attention [Katharopoulos et al., 2020].

C.2 Multi-Head Modern Hopfield Models

To derive the multi-head Hopfield model, we cast $\mathcal{T}_{\text{Multi}}$ as multiple SVR problems such that the memorization of memory patterns $\boldsymbol{\Xi}$ corresponds to training a regression model $\mathcal{T}_{\text{Multi}}$ on datasets $\{\boldsymbol{\Xi}_s\}_{s\in[H]}$. These S training data sets are given as $\{(\boldsymbol{\xi}_\mu^1 + \delta\boldsymbol{\xi}_\mu^1, \boldsymbol{\xi}_\mu^1)\}_{\mu\in[M]}, \cdots, \{(\boldsymbol{\xi}_\mu^H + \delta\boldsymbol{\xi}_\mu^H, \boldsymbol{\xi}_\mu^H)\}_{\mu\in[M]}$. To handle multiple regression problems, we extend the regression model Definition 3.2 into the following.

Definition C.1 (Multi-Head Regression Model)**.** Given an input vector $\mathbf{x} \in \mathbb{R}^d$. The output $\widehat{\mathbf{y}} \in \mathbb{R}^d$ of the regression model $\mathcal{T}_{\text{multi}}$ is defined as:

$$\widehat{\mathbf{y}} = \mathcal{T}_{\text{Multi}}(\mathbf{x}) \coloneqq \sum_{s=1}^{H} \mathbf{W}_O^s \left(\mathbf{W}^s \frac{\Phi^s(\mathbf{x})}{h^s(\mathbf{x})} \right) \in \mathbb{R}^d, \tag{C.3}$$

where $\mathbf{W}_O^s \in \mathbb{R}^{d\times d}$, $\mathbf{W}^s = [\mathbf{w}_1^s, \cdots, \mathbf{w}_d^s]^\mathsf{T} \in \mathbb{R}^{d\times D_\Phi}$ for all $s \in [H]$, $\Phi^s(\mathbf{x}) = [\phi_1^s(\mathbf{x}), \cdots, \phi_{D_\Phi}^s(\mathbf{x})] : \mathbb{R}^d \to \mathbb{R}^{D_\Phi}$ and $h^s(\mathbf{x}) \in \mathbb{R}$ denote a series of output projection matrices, weighted matrix, kernel mapping and normalization function, respectively.

Adopting this multi-head regression model, we introduce the following multi-head modern Hopfield model.

Proposition C.2 (Multi-Head Modern Hopfield Models)**.** Let $h(\mathbf{x}) \coloneqq \sum_{\mu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle$ and $\Phi(\cdot) = (\phi_0^{(0)}, \phi_1^{(1)}, \ldots, \phi_{D_1}^{(1)}, \ldots, \phi_1^{(n)}, \ldots, \phi_{D_n}^{(n)}, \ldots)$ with, for $1 \le D' \le D_n$,

$$\phi_{D'}^{(n)} \coloneqq \frac{(\sqrt{\beta}x_1)^{\ell_1} \cdots (\sqrt{\beta}x_d)^{\ell_d}}{\sqrt{\ell_1! \cdots \ell_d!}}, \tag{C.4}$$

where $\ell_1 + \cdots + \ell_d = n$, and $D_n \coloneqq \binom{d+n-1}{n}$. By Theorem 3.1, fitting $\mathcal{T}_{\text{SVR}}$ on H training data sets $\{(\boldsymbol{\xi}_\mu^1 + \delta\boldsymbol{\xi}_\mu^1, \boldsymbol{\xi}_\mu^1)\}_{\mu\in[M]}, \cdots, \{(\boldsymbol{\xi}_\mu^H + \delta\boldsymbol{\xi}_\mu^H, \boldsymbol{\xi}_\mu^H)\}_{\mu\in[M]}$ following (3.2) gives

$$\mathcal{T}_{\text{Multi}}(\mathbf{x}) = \sum_{s=1}^{H} \mathbf{W}_O^s \left(\sum_{\mu=1}^{M} \mathrm{Softmax}(\beta \langle \mathbf{x}, \boldsymbol{\xi}_\mu^s + \delta\boldsymbol{\xi}_\mu^s \rangle) \boldsymbol{\xi}_\mu^s \right). \tag{C.5}$$

This model is by design connected to the standard multi-head attention.

C.3 PRFs (Positive Random Features) Kernel Modern Hopfield Model

Proposition C.3 (Positive Random Features Modern Hopfield Model)**.** Let $h(\mathbf{x}) \coloneqq \sum_{\mu=1}^{M} \langle \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu), \Phi(\mathbf{x}) \rangle$ and $\Phi(\cdot) = (\phi_1, \ldots, \phi_{D_\Phi})$ with

$$\Phi(\mathbf{x}) \coloneqq \frac{\Psi(\mathbf{x})}{\sqrt{D_\Phi}} (\psi_1(\langle \mathbf{p}_1, \mathbf{x} \rangle), \ldots, \psi_1(\langle \mathbf{p}_m, \mathbf{x} \rangle), \ldots, \psi_l(\langle \mathbf{p}_1, \mathbf{x} \rangle), \ldots, \psi_l(\langle \mathbf{p}_m, \mathbf{x} \rangle)), \tag{C.6}$$

where $D_\Phi = l \cdot m$, $\Psi : \mathbb{R}^d \to \mathbb{R}$, $\psi_1, \ldots, \psi_m$ are functions that map from $\mathbb{R} \to \mathbb{R}$, and $\mathbf{p}_1, \ldots, \mathbf{p}_m \overset{iid}{\sim} \mathcal{P}$ are vectors from some distribution $\mathcal{P} \in \Delta^d$ ($\Delta^d \coloneqq \{\mathbf{p} \in \mathbb{R}_+^d \mid \sum_{i=1}^d p_i = 1\}$ is the $(d-1)$-dimensional unit simplex.). By Theorem 3.1, fitting $\mathcal{T}_{\text{SVR}}$ on $\mathcal{D}$ following (3.2) gives

$$\mathcal{T}_{\text{PRF}}(\mathbf{x}) = \sum_{\mu=1}^{M} \mathbb{E}_{\mathcal{D}}[\widehat{D}^{-1} \langle \Phi(\mathbf{x}), \Phi(\boldsymbol{\xi}_\mu + \delta\boldsymbol{\xi}_\mu) \rangle] \boldsymbol{\xi}_\mu, \tag{C.7}$$

where we adopt the normalization map $\widehat{D}^{-1} \coloneqq \langle \boldsymbol{\xi}_1, \mathbf{x} \rangle$ given by [Choromanski et al., 2021].

Comparing with regular modern Hopfield model, PRF Hopfield model only has the linear space and time complexity, without any additional treatment such as introducing sparsity or low-rankness. The significance of this representational capability lies in its ability to facilitate a precise comparison between softmax and alternative kernels in the context of extensive tasks, surpassing the capabilities of regular modern Hopfield models and enabling a comprehensive exploration of optimal kernels. This model is by design connected to the Performer-type attention [Choromanski et al., 2021]. In practice, the default option for $\mathcal{P}$ is standard Gaussian [Choromanski et al., 2021].

D Nonparametric Modern Hopfield Layers for Deep Learning

Building on the link between the nonparametric modern Hopfield models and the attention mechanisms, we introduce the Nonparametric Hopfield (NPH) layers for deep learning.

Following [Hu et al., 2023, Ramsauer et al., 2020], we say $\mathbf{X}$ and $\mathbf{\Xi}$ are in the associative space (embedded space), as they are embedded from the *raw* query $\mathbf{R}$ and $\mathbf{Y}$ memory patterns, respectively, via $\mathbf{X}^\mathsf{T} = \mathbf{R}\mathbf{W}_Q \coloneqq \mathbf{Q}$, and $\mathbf{\Xi}^\mathsf{T} = \mathbf{Y}\mathbf{W}_K \coloneqq \mathbf{K}$, with some $\mathbf{W}_Q$ and $\mathbf{W}_K$. Taking the transport of $\mathcal{T}$ in (3.4) (with a given feature map Φ) and multiplying with $\mathbf{W}_V$ such that $\mathbf{V} \coloneqq \mathbf{K}\mathbf{W}_V$, we have

$$\mathbf{Z} \coloneqq \mathbf{Q}^{\text{new}}\mathbf{W}_V = \mathcal{T}_{\text{SVR}}\left(\beta\mathbf{Q}\mathbf{K}^\mathsf{T}\right)\mathbf{V}, \tag{D.1}$$

which leads to an attention mechanisms with various $\mathcal{T}_{\text{SVR}}$ as activation functions. Plugging back the raw patterns $\mathbf{R}$ and $\mathbf{Y}$, we arrive the Nonparametric Modern Hopfield (`NPH`) layer(s),

$$\texttt{NPH}\left(\mathbf{R}, \mathbf{Y}\right) = \mathcal{T}_{\text{SVR}}\left(\beta\mathbf{R}\mathbf{W}_Q\mathbf{W}_K^\mathsf{T}\mathbf{Y}^\mathsf{T}\right)\mathbf{Y}\mathbf{W}_K\mathbf{W}_V, \tag{D.2}$$

which can be seamlessly integrated into deep learning architectures. Concretely, the `NPH` layers take matrices $\mathbf{R}$, $\mathbf{Y}$ as inputs, with the weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$. Depending on its configuration, it offers several functionalities:

1. **Memory Retrieval:** In this learning-free setting, weight matrices $\mathbf{W}_K$, $\mathbf{W}_Q$, and $\mathbf{W}_V$ are set as identity matrices. Here, $\mathbf{R}$ represents the query input, and $\mathbf{Y}$ denotes the stored memory patterns for retrieval.
2. `NPH`**:** This configuration takes $\mathbf{R}$ and $\mathbf{Y}$ as inputs. Intending to substitute the attention mechanism, the weight matrices $\mathbf{W}_K$, $\mathbf{W}_Q$, and $\mathbf{W}_V$ are rendered learnable. Furthermore, $\mathbf{R}$, $\mathbf{Y}$, and $\mathbf{Y}$ serve as the sources for query, key, and value respectively. Achieving a self-attention-like mechanism requires setting $\mathbf{R}$ equal to $\mathbf{Y}$.
3. `NPHPooling`**:** With inputs $\mathbf{Q}$ and $\mathbf{Y}$, this layer uses $\mathbf{Q}$ as a static **prototype pattern**, while $\mathbf{Y}$ contains patterns over which pooling is desired. Given that the query pattern is replaced by the static prototype pattern $\mathbf{Q}$, the only learnable weight matrices are $\mathbf{W}_K$ and $\mathbf{W}_V$.
4. `NPHLayer`**:** The `NPHLayer` layer takes the query $\mathbf{R}$ as its single input. The layer equips with learnable weight matrices $\mathbf{W}_K$ and $\mathbf{W}_V$, which function as our stored patterns and their corresponding projections. This design ensures that our key and value are decoupled from the input. In practice, we set $\mathbf{W}_Q$ and $\mathbf{Y}$ as identity matrices.

E Experimental Studies

We verify the method proposed in the main content with the following experimental sections.

1. Memory Retrieval Task (LHS of Figure 1).
2. Multiple Instance Learning on MNIST (RHS of Figure 1).
3. Multiple Instance Learning on Real World Datasets.
4. Time Series Prediction.
5. Computational Efficiency.

We consider the following variations of Modern Hopfield Models in this paper:

- Dense Modern Hopfield [Ramsauer et al., 2020]
- Sparse Modern Hopfield [Hu et al., 2023]
- Sparse-Structured Modern Hopfield:
  - Random Masked Modern Hopfield
  - Window Modern Hopfield
  - Top-K Modern Hopfield
- Linear Modern Hopfield
- Random Feature Modern Hopfield

E.1 Memory Retrieval Task (LHS of Figure 1)

In the memory retrieval task, we examine two datasets: MNIST (sparse) and CIFAR10 (dense). We employ the sum-of-squares distance between the retrieved image and the ground truth image to measure retrieval error. This experiment encompasses two settings: (1) Half-masked image recovery, and (2) Noisy image recovery. In the half-masked image recovery scenario, we obscure half of the pixels in the image. The memory set size (M) is varied from 10 to 100, and we report the average retrieval success rate over 50 runs. In the noisy image recovery scenario, we fix the memory set size at 20, and introduce varying scales of Gaussian noise to the image, with variance ranging from 0.1 to 1.4.

E.1.1 Implementation Details

For each memory set size (M), we randomly select 20 images from the memory set for retrieval. The memory set itself is chosen randomly from the dataset in each iteration. We adhere to the implementation outlined in [Hu et al., 2023].

E.2 Multiple Instance Learning on MNIST (LHS of Figure 1)

Quoted from [Hu et al., 2023, Section 4.2]:

Multiple Instance Learning (MIL) [Ilse et al., 2018, Carbonneau et al., 2018] is a variation of supervised learning where the training set consists of labeled bags, each containing multiple instances. The goal of MIL is to predict the bag labels based on the instances they contain, which makes it particularly useful in scenarios where labeling individual instances is difficult or impractical, but bag-level labels are available. Examples of such scenarios include medical imaging (where a bag could be an image, instances could be patches of the image, and the label could indicate the presence or absence of disease) and document classification (where a bag could be a document, instances could be the words or sentences in the document, and the label could indicate the topic or sentiment of the document).

In this experiment, we designate one digit from MNIST as a negative signal, and the remaining digits as positive signals. The objective is to predict whether a given bag of instances (digits) contains the negative signal. We vary the memory set size (M) from 5 to 100 and report the mean accuracy over 10 runs. We compare the performance of Dense Hopfield, Sparse Hopfield, Top-K Hopfield (with 20%, 50%, and 80%), Random Feature Hopfield, Random Masked Hopfield and Linear Hopfield. We omit Window Hopfield for reasons mentioned earlier. The result can be found in Figure 2.

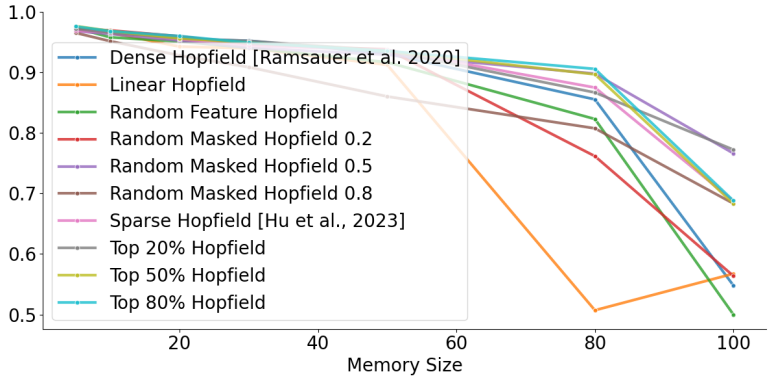

Figure 2: **Comparison of MIL performance varying bag sizes.** We employ various variants of the `HopfieldPooling` layers to observe to performance change with respect to different bag size. We see that for those generate actual sparse matrices (Top-K, Random Masked and Sparse Hopfield), their performances are more robust against bag size increase.

E.2.1 Implementation Details

We employ an embedding layer to project the flattened MNIST images into the hidden space, followed by a layer of layer normalization. Subsequently, we utilize the Hopfield Pooling layer to pool over all the instances in the bag, followed by a second layer normalization layer. Finally, a fully connected layer is used to project the hidden representation of the bag into the label space. All models are trained using the AdamW optimizer for 150 epochs, with a cosine annealing learning rate decay applied to all models. Note that we exclude Window Hopfield in this and the subsequent MIL experiment since Window Hopfield requires both the query and memory pattern numbers to be large to perform the sliding window operation. However, in our model structure, the number of query patterns in the pooling layer is set to 2. The details of the hyperparameters can be found in Table 2.

Table 2: Hyperparameter used in the MIL MNIST experiment.

parameter	**values**
batch size	256
learning rate	1e-3
embedding dimension	256
number of heads	4
head dimension	64
test set size	500
train set size	2000
scaling	0.1
num of pattern	2
epochs	150

E.3 Multiple Instance Learning on Real World Datasets

For this experiment, we follow [Ramsauer et al., 2020, Hu et al., 2023] to conduct MIL experiments on real world datasets. However, we employ a simpler model structure and a smaller hyperparameter search space, rendering our results incomparable. We utilize four datasets: Elephant, Fox, and Tiger for image annotation [Ilse et al., 2018], and UCSB breast cancer classification [Kandemir et al., 2014]. We compare Dense Hopfield, Sparse Hopfield, TopK Hopfield at 20%, 50%, and 80%, Random Feature Hopfield, and Linear Hopfield. Random Masked Hopfield is excluded due to its non-deterministic inference, and Window Hopfield is omitted as previously mentioned. The results are presented in Table 5.

E.3.1 Dataset Details

The experiment is conducted on four MIL datasets. **Elephant**, **Fox**, and **Tiger** are designed for image annotation and consist of preprocessed and segmented colored images. Each image is characterized by descriptors for color, texture, and shape. These datasets each contain 100 positive images featuring the specified animal and 100 negative images drawn from a set of images depicting other animals. Additionally, we evaluate our model on the **UCSB** breast cancer classification task. In the UCSB dataset, each instance comprises a patch of a histopathological image depicting either cancerous or normal tissue. The detailed statistics of the datasets are reported in Table 3.

Table 3: Statistics of MIL benchmark datasets

Name	Instances	Features	Bags	+bags	−bags
Elephant	1391	230	200	100	100
Fox	1302	230	200	100	100
Tiger	1220	230	200	100	100
UCSB	2002	708	58	26	32

E.3.2 Implementation Details

We follow the experimental setting in [Ramsauer et al., 2020] and employ stratified 10-fold cross-validation to evaluate the performance of each baseline Hopfield model. In each fold, we utilize a stratified sampling process to partition the data into a training set and a validation set, with a split rate of 0.1. Hyperparameters are optimized via random search by maximizing the ROC-AUC score on the validation set. All reported ROC-AUC scores represent the average results over 5 runs with different random seeds. The random search space is delineated in Table 4, with the number of trials set to 50 for each fold. The embedding layer, a pre-HopfieldPooling linear network, has its layer width determined by the number of hidden units. A dropout operation, also referred to as bag dropout, is applied post the embedding layer and the Hopfield Pooling layer. Notably, to better showcase the performance of Top-k Hopfield, dropout is not applied to the attention weight. All models are trained using the Adam optimizer over 50 epochs. To mitigate overfitting, an early-stopping mechanism is employed, selecting the best checkpoint based on the validation set.

Table 4: Hyperparameter random search space on the respective validation sets of the Elephant, Fox, Tiger and UCSB breast cancer datasets.

parameter	**values**
batch size	$\{4, 8, 16\}$
learning rates	$\{10^{-3}, 10^{-4}, 10^{-5}\}$
weight decay	$\{0, 10^{-3}, 10^{-4},\}$
layer width	$\{128, 256, 512\}$
number of heads	$\{4, 8\}$
scaling factors	$\{0.1, 1\}$
dropout	$\{0.0, 0.3\ 0.5\}$

Table 5: Results for MIL benchmark datasets in terms of AUC score. The results suggest that the proposed model achieves performance comparable to the existing Dense and Sparse Modern Hopfield models [Hu et al., 2023, Ramsauer et al., 2020]. Note that, since our aim here is to conduct an *atomic* setting for fair comparison, we employ a simpler network structure (with smaller hyperparameter search space) compared to the ones used in [Hu et al., 2023, Ramsauer et al., 2020]. Consequently, our results do not align with those in [Hu et al., 2023] for Dense and Sparse Modern Hopfield Models.

Method	Tiger	Fox	Elephant	UCSB
Dense Hopfield [Ramsauer et al., 2020]	0.813	0.563	0.877	0.524
Sparse Hopfield [Hu et al., 2023]	0.830	0.573	0.893	0.585
Top-20% Hopfield	0.824	0.562	0.848	0.586
Top-50% Hopfield	0.812	0.566	0.852	0.572
Top-80% Hopfield	0.812	0.560	0.872	0.551
Random Feature Hopfield	0.802	0.508	0.875	0.566
Linear Hopfield	0.797	0.571	0.869	0.561

E.4 Time Series Prediction

We further showcase the performance (in Table 6) and efficiency (in Figure 3) of the proposed nonparametric modern Hopfield models with multivariate time series prediction tasks.

E.4.1 Implementation Details

For ease of comparison, we employ the simplest possible architecture: an embedding layer to project each signal into a hidden space, followed by a single Hopfield layer. By doing so, we treat every signal as a query pattern. Next, we employ a Hopfield Pooling layer to pool over all the signals into a single hidden vector. Finally, we utilize a fully connected layer to generate the prediction. For all experiments, we maintain the same input and prediction horizon for simplicity. The results can be found in Table 6 and Figure 3.

Datasets. We conduct the experiments on four multivariate time series real-world datasets: ETTh1 (Electricity Transformer Temperature-hourly), ETTm1 (Electricity Transformer Temperature-minutely), WTH (Weather), ECL (Electricity Consuming Load), Traffic.

Setup. For each dataset, we use their univariate setting for our time series prediction experiment. We choose Dense, Sparse, Random Feature, Linear, TopK and Window Hopfield as baselines. We select 4 different prediction horizons for demonstration, which are $96, 196, 336, 720$. We report the average error of 5 runs, evaluated using Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics. For window Hopfield, we set the window size as $8, 12, 14, 16$, w.r.t. $96, 196, 336, 720$.

Table 6: Time series prediction using different Hopfield layers (Appendix D) across five datasets. We evaluate each dataset with different prediction horizons (showed in the second column). We report the average Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics of 5 runs. **RF** denotes the **R**andom **F**eature Hopfield layer. One notable observation is that the noise level of the dataset significantly influences time series prediction. Therefore, employing Hopfield layers with strong noise-robustness offers performance improvements. Moreover, based on our results, the proposed efficient Hopfield models not only offer significant computational efficiency but also maintain comparable performance. Especially, the Random Feature Hopfield and Linear Hopfield layers (models) not only match but even outperform Dense Hopfield model in several settings. As a side note, Window Hopfield exhibits significant performance degradation in most settings. This degradation arises because it solely focuses on local information. Being the only Hopfield model that does not span the entire associative range (i.e., sequence length), it overlooks a substantial portion of the autoregressive correlation present in time series data. We also record the time used for one epoch on ETTh1 dataset with different prediction horizon (input length as well). The duration time per epoch was showed in Figure 3.

Models		**Dense**		**Sparse**		**Top**20%		**Top**50%		**Top**80%		**Window**		**RF**		**Linear**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.137	0.307	0.144	0.314	0.148	0.319	0.153	0.321	0.147	0.318	1.043	0.881	0.147	0.312	0.149	0.320
	192	0.153	0.326	0.152	0.325	0.146	0.318	0.161	0.333	0.150	0.320	1.003	0.870	0.158	0.332	0.141	0.313
	336	0.148	0.319	0.146	0.319	0.156	0.327	0.122	0.286	0.160	0.333	0.889	0.767	0.151	0.322	0.138	0.307
	720	0.169	0.331	0.148	0.314	0.184	0.345	0.161	0.327	0.123	0.287	0.756	0.761	0.141	0.271	0.171	0.333
ETTm1	96	0.148	0.301	0.147	0.301	0.144	0.311	0.151	0.310	0.142	0.31o	0.943	0.854	0.151	0.314	0.155	0.319
	192	0.189	0.350	0.187	0.340	0.191	0.347	0.185	0.338	0.188	0.341	1.054	0.893	0.190	0.347	0.192	0.348
	336	0.163	0.320	0.165	0.322	0.168	0.331	0.161	0.312	0.169	0.330	0.873	0.334	0.175	0.333	0.176	0.337
	720	0.159	0.300	0.161	0.303	0.165	0.313	0.167	0.313	0.169	0.320	0.764	0.731	0.162	0.309	0.165	0.310
ECL	96	0.378	0.371	0.373	0.370	0.384	0.382	0.386	0.386	0.383	0.376	0.989	0.854	0.390	0.403	0.365	0.378
	192	0.486	0.426	0.535	0.507	0.502	0.427	0.501	0.464	0.519	0.481	1.000	0.843	0.543	0.438	0.549	0.464
	336	0.748	0.693	0.760	0.688	0.650	0.549	0.674	0.571	0.638	0.545	1.012	0.849	0.767	0.588	0.672	0.578
	720	0.961	0.711	0.993	0.758	1.145	0.843	1.166	0.847	1.211	0.872	1.061	0.865	1.362	0.896	1.052	0.770
WTH	96	0.347	0.474	0.347	0.477	0.348	0.474	0.348	0.474	0.356	0.479	0.952	0.819	0.345	0.470	0.355	0.476
	192	0.399	0.505	0.386	0.497	0.360	0.482	0.370	0.490	0.361	0.482	0.977	0.828	0.368	0.487	0.354	0.478
	336	0.407	0.512	0.387	0.501	0.376	0.489	0.397	0.503	0.403	0.505	0.931	0.808	0.392	0.504	0.407	0.514
	720	0.669	0.631	0.632	0.623	0.590	0.604	0.569	0.593	0.618	0.618	0.564	0.595	0.564	0.595	0.747	0.676
Traffic	96	1.466	0.654	1.489	0.638	1.483	0.645	1.517	0.630	1.477	0.638	1.520	0.625	1.515	0.635	1.489	0.644
	192	1.551	0.654	1.550	0.657	1.557	0.649	1.548	0.657	1.551	0.652	1.570	0.637	1.551	0.654	1.551	0.653
	336	1.595	0.663	1.595	0.662	1.599	0.663	1.592	0.665	1.604	0.657	1.612	0.646	1.613	0.646	1.614	0.646
	720	1.660	0.681	1.671	0.671	1.664	0.674	1.676	0.663	1.682	0.661	1.683	0.661	1.682	0.661	1.681	0.660

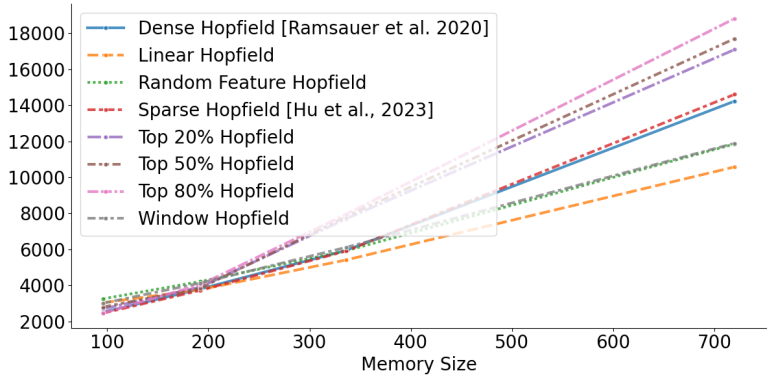

Figure 3: The processing time comparison among different Hopfield models utilized in the time series prediction task described in Table 6. We evaluate the efficiency of multivariate time series prediction on ETTh1 dataset. The findings are consistent with the efficiency discussion in Section 3.2, where the Sparse/Dense/Top-K models (all with $\mathcal{O}(d^2)$ complexity) necessitate more time to complete an epoch. In conjunction with the results in Figure 1, it is evident that the efficient modern Hopfield models (Window, Linear, Random Feature) not only converge in fewer or comparable epochs but also require less time per epoch compared to the less efficient (Sparse/Dense/Top-K) Hopfield models.

E.5 Computational Efficiency

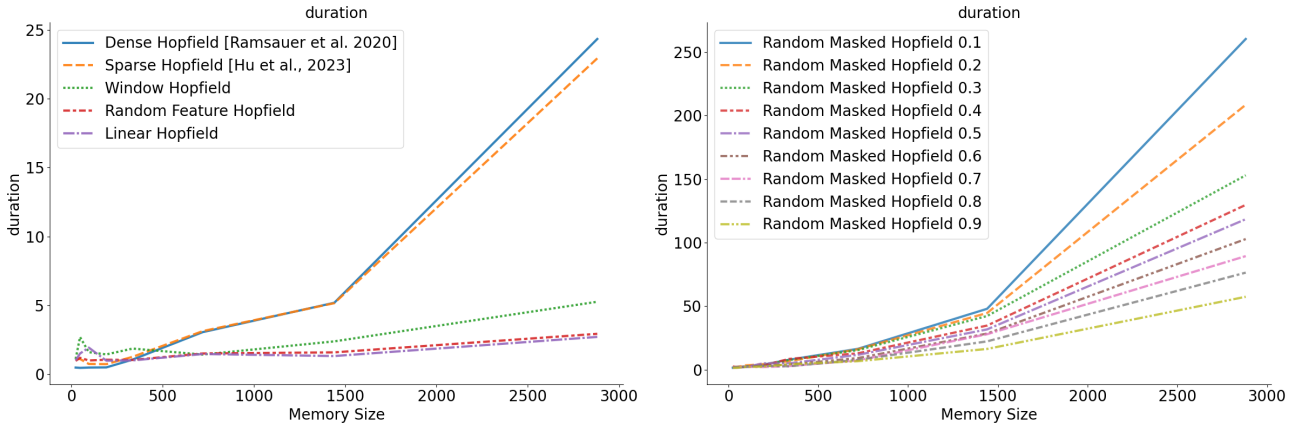

Figure 4: **(LHS:)** Comparison of duration (ms) per batch for different Hopfield Models. **(RHS:)** The scaling behavior of Random Masked Hopfield with different masking ratios. The probability denotes the ratio being masked out. We employ various variants of the `Hopfield` layers to process a batch of tensors, with a batch size of 4 and a hidden dimension of 16. We vary the input memory size (input length). Note that we separate the Random Masked Hopfield from other baselines since the sparse matrix operation in PyTorch, still in the beta stage, may not be as fully optimized as dense tensor operations.

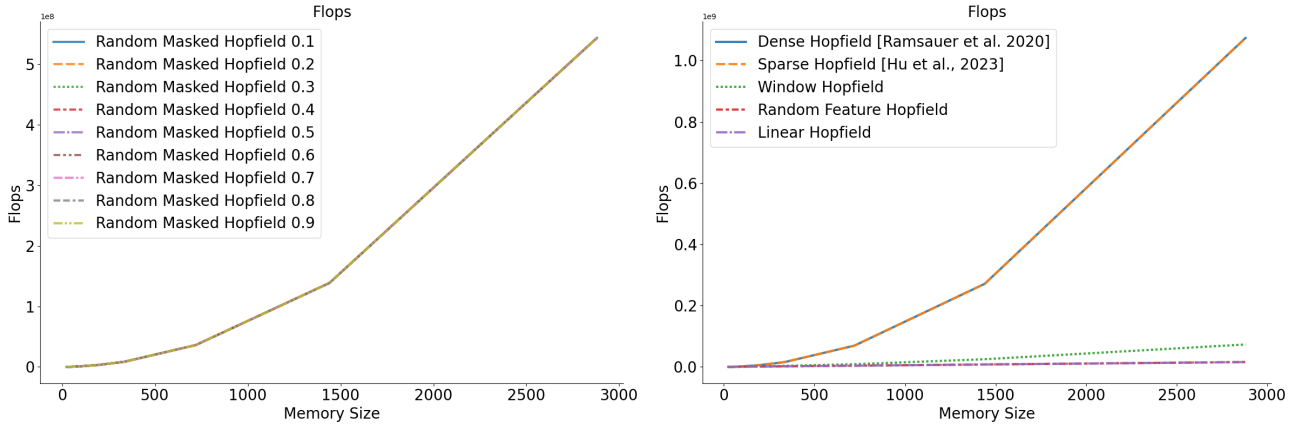

Figure 5: **(LHS:)** The FLOPs comparison for Random Masked Hopfield with different probabilities is depicted. The lines for Dense and Sparse Hopfield are overlapped, as are the lines for Random Feature Hopfield and Linear Hopfield. **(RHS:)** The FLOPs comparison across different Hopfield Models is shown. We employ the same settings as in the duration figure. Note that the **fvcore** package may count sparse matrix operations as normal floating point operations, which is why we might not see a difference.

E.5.1 Implementation Details

In this section, we exclusively evaluate the computational efficiency of different Hopfield models with respect to varying input lengths using the *Hopfield* layer. We report the average duration time per batch, as shown in Figure 4, and the FLOPs concerning different input lengths (memory sizes), as depicted in Figure 5. It's notable that different code implementation methods could potentially affect computational efficiency. We use a randomized batched tensor as input x, where $x \in \mathbb{R}^{\text{memory size} \times 16}$, and the batch size is 4 [4]. For Random Feature Hopfield and Linear Hopfield, we adhere to the Performer

[4] approximately $(4 \times 4 \times 16 \times \text{memory size})$ bytes

implementation[5], while for Window Hopfield, we follow the Longformer implementation[6]. For Random Masked Hopfield, we utilize the `torch.sparse.sampled_addmm`[7] feature, and for other baselines, we employ standard PyTorch built-in functions for implementation. We report the average forward pass time over 10 runs, alongside the FLOPs, with both metrics evaluated on different input lengths. FLOPs are calculated using the **fvcore** package[8]. Note that most publicly available packages for FLOPs profiling are either under development or in beta, hence calculation errors are anticipated. Additionally, the `torch.sparse` package is also in beta, implying its performance may not be fully optimized, especially regarding FLOPs calculation and operation overhead.

E.5.2 Discussion

Note that, by nature, both Dense and Sparse Hopfield exhibit the same FLOPs. Moreover, it is observed that Random Feature Hopfield and Linear Hopfield also share the same FLOPs, as the only distinction between them lies in the kernel function. Regarding Window Hopfield, its FLOPs fall in between, demonstrating notable efficiency compared to both Dense and Sparse Hopfield. In terms of duration time per batch, Sparse Hopfield appears slightly faster than its dense counterpart, likely due to the additional zeros generated by sparsemax. Window Hopfield, on the other hand, showcases a significant reduction in duration compared to Sparse Hopfield. Lastly, it is noted that the processing time for both Random Feature Hopfield and Linear Hopfield converges as the memory size increases.

[5] https://github.com/lucidrains/performer-pytorch
[6] https://github.com/allenai/longformer
[7] https://pytorch.org/docs/stable/generated/torch.sparse.sampled_addmm.html#torch.sparse.sampled_addmm
[8] https://github.com/facebookresearch/fvcore