Uniform Memory Retrieval with Larger Capacity for Modern Hopfield Models

Dennis Wu [*1] **Jerry Yao-Chieh Hu** [*1] **Teng-Yun Hsiao** [2] **Han Liu** [1 3]

Abstract

We propose a two-stage memory retrieval dynamics for modern Hopfield models, termed `U-Hop`, with enhanced memory capacity. Our key contribution is a learnable feature map Φ which transforms the Hopfield energy function into kernel space. This transformation ensures convergence between the local minima of energy and the fixed points of retrieval dynamics within the kernel space. Consequently, the kernel norm induced by Φ serves as a novel similarity measure. It utilizes the stored memory patterns as learning data to enhance memory capacity across all modern Hopfield models. Specifically, we accomplish this by constructing a separation loss $\mathcal{L}_\Phi$ that separates the local minima of kernelized energy by separating stored memory patterns in kernel space. Methodologically, `U-Hop` memory retrieval process consists of: **(Stage I)** minimizing separation loss for a more uniformed memory (local minimum) distribution, followed by **(Stage II)** standard Hopfield energy minimization for memory retrieval. This results in a significant reduction of possible metastable states in the Hopfield energy function, thus enhancing memory capacity by preventing memory confusion. Empirically, with real-world datasets, we demonstrate that `U-Hop` outperforms all existing modern Hopfield models and SOTA similarity measures, achieving substantial improvements in both associative memory retrieval and deep learning tasks. Code is available at GitHub; future updates are on arXiv.

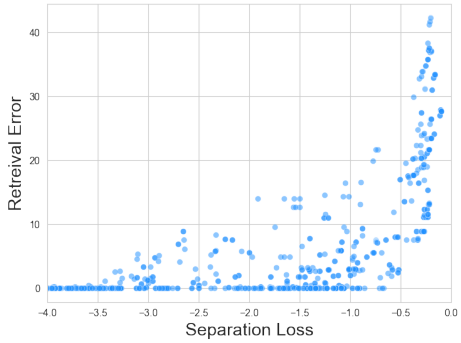

Figure 1. **Separation Loss over Memory Set v.s. Retrieval Error.** We perform 200 runs of memory retrieval with `U-Hop` on MNIST. The result shows a strong correlation between low separation loss and low retrieval error.

1. Introduction

We address the memory confusion problem in the modern Hopfield models by proposing a two-stage optimization formulation, termed `U-Hop`, for the memory retrieval dynamics of modern Hopfield models. We construct the similarity measure of modern Hopfield models with a learnable kernel. The feature map of the kernel is trained by maximizing the separation among the entire stored memory set (Figure 2). This allows Hopfield models under `U-Hop` to distinguish different memory patterns with larger separation and hence achieve larger memory capacity.

Let $\mathbf{x} \in \mathbb{R}^d$ be the input query pattern, $\mathbf{\Xi} := [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_M] \in \mathbb{R}^{d\times M}$ be the memory patterns, and $\langle \mathbf{a}, \mathbf{b}\rangle := \mathbf{a}^\mathsf{T}\mathbf{b}$ be the inner product for vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. Hopfield models are energy-based associative memory models. They store memory patterns on the local minima of their energy landscapes. For any input query $\mathbf{x}$, they retrieve its closest memory pattern through some energy minimization algorithms initialized at $\mathbf{x}$. These algorithms are also known as memory retrieval dynamics. See Figure 2 for a visualization. Ramsauer et al. (2020) proposed a large foundation model compatible variant, the Modern Hopfield Model (MHM). This model has a specific set of energy function and retrieval dynamics, such that it subsumes transformer attention as its special case (see Appendix C) and enjoys superior theoretical properties (see (Hu et al., 2023; Wu et al., 2024; Ramsauer et al., 2020)). Specifically, they

[*]Equal contribution [1]Department of Computer Science, University of Northwestern, Evanston, USA [2]Department of Physics, National Taiwan University, Taipei, Taiwan [3]Department of Statistics and Data Science, University of Northwestern, Evanston, USA. Correspondence to: Dennis Wu <hibb@northwestern.edu>, Jerry Yao-Chieh Hu <jhu@northwestern.edu>, Teng-Yun Hsiao <b10502058@ntu.edu.tw>, Han Liu <hanliu@northwestern.edu>.

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

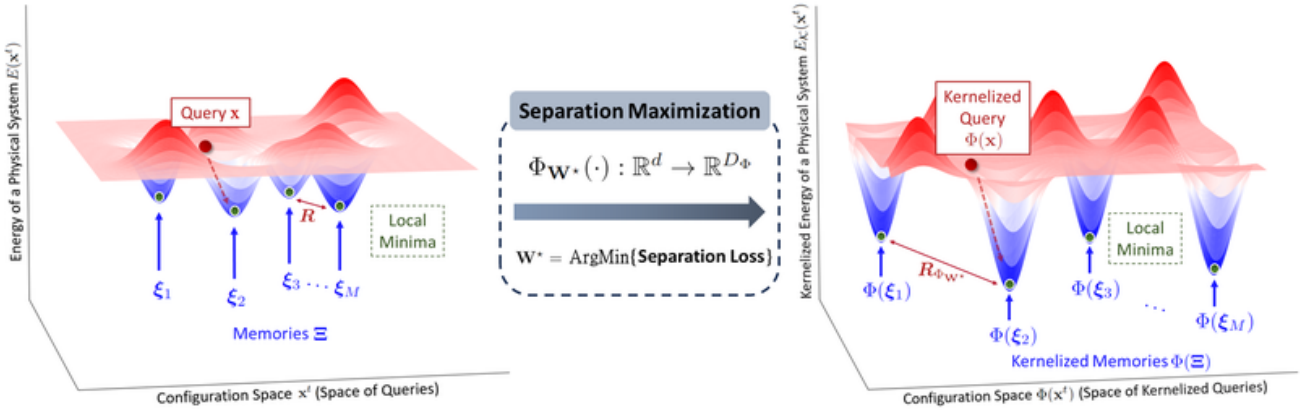

Figure 2. **Visualization of U-Hop: Separation Maximization First, then Memory Retrieval Dynamics.** The LHS represents the energy landscape in original state space, where the memories stay close to each other. With separation loss minimization, we obtain a Φ parameterized by $\mathbf{W}^\star$, that is able to relocate memory patterns in the kernel space to more uniform locations, and thus results in the separation between local minima of $E_{\mathcal{K}}$.

introduce the energy function:

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

and the retrieval dynamics

$$\mathbf{x}^{\text{new}} = \mathcal{T}_{\text{MHM}}(\mathbf{x}) = \boldsymbol{\Xi} \cdot \operatorname{Softmax}(\beta \boldsymbol{\Xi}^\mathsf{T}\mathbf{x}), \qquad (1.2)$$

where $\operatorname{lse}(\beta, \mathbf{z}) := \log\left(\sum_{\mu=1}^M e^{\beta z_\mu}\right)/\beta$ is the log-sum-exponential function for any given vector $\mathbf{z} \in \mathbb{R}^M$ and $\beta > 0$. The dot-product $\boldsymbol{\Xi}^\mathsf{T}\mathbf{x}$ in the lse function is known as the *overlap construction* and serves as the similarity measure between the input query $\mathbf{x}$ and memory set $\boldsymbol{\Xi}$.

One highlighted property of modern Hopfield models is their memory capacity, which is exponential in pattern dimension (Ramsauer et al., 2020; Hu et al., 2023; Wu et al., 2024). However, their memory capacity and retrieval error are dependent on the quality of memory distribution. To be concrete, we define the memory storage and retrieval as[1]

Definition 1.1 (Stored and Retrieved)**.** For all $\mu \in [M]$, let $R := \frac{1}{2}\min_{\mu,\nu\in[M];\mu\neq\nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$ be the finite radius of each sphere $\mathcal{S}_\mu$ centered at memory pattern $\boldsymbol{\xi}_\mu$. We say $\boldsymbol{\xi}_\mu$ is *stored* if all $\mathbf{x} \in \mathcal{S}_\mu$ are generalized fixed points of $\mathcal{T}$, $\mathbf{x}_\mu^\star \in \mathcal{S}_\mu$, and $\mathcal{S}_\mu \cap \mathcal{S}_\nu = \emptyset$ for $\mu \neq \nu$. We say $\boldsymbol{\xi}_\mu$ is ϵ-*retrieved* by $\mathcal{T}$ with $\mathbf{x}$ for an error ϵ, if $\|\mathcal{T}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \epsilon$.

Let $\Delta_\mu := \langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu\rangle - \max_{\nu\in[M],\nu\neq\mu}\langle \boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu\rangle$ be the separation between a memory pattern $\boldsymbol{\xi}_\mu$ from all other memories in $\boldsymbol{\Xi}$, and m be the largest norm among memory patterns. Ramsauer et al. (2020) gives the retrieval error bound:

$$\|\mathcal{T}_{\text{MHM}}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq 2m(M-1)e^{-\beta(\Delta_\mu - 2mR)}, \qquad (1.3)$$

for any $\mathbf{x} \in \mathcal{S}_\mu$. This bound is crucial not only for characterizing retrieval quality but also, in capacity analysis, as a necessary condition for pattern $\boldsymbol{\xi}_\mu$ to be stored in the model (Hu et al., 2023, Theorem 3.1). Yet, it depends on $\boldsymbol{\Xi}$.

Δ_μ measures the distance from a given $\boldsymbol{\xi}_\mu$ to the nearest memory pattern in $\boldsymbol{\Xi}$. R measures the minimal separation among all stored memories $\boldsymbol{\Xi}$. Hence, they are both $\boldsymbol{\Xi}$-dependent. This $\boldsymbol{\Xi}$-dependence in (1.3) results in potential *fuzzy retrievals*, namely metastable states caused by multiple nearby local minima in the energy landscape, especially when $\Delta_\mu - 2mR$ is small. When this occurs, these fuzzy retrievals deviate the retrieval process from the ground truth, thereby hampering performance.

This fuzzy memory (memory confusion) issue is well-known in literature. The dense associative memory model (Krotov and Hopfield, 2016) tries to solve this issue by using polynomial energy fucntion. The modern Hopfield models (Demircigil et al., 2017; Ramsauer et al., 2020; Hu et al., 2023; Wu et al., 2024; Hu et al., 2024b) try to solve this issue by using exponential energy functions. However, all these attempts still rely on the quality of $\boldsymbol{\Xi}$. In this work, we rethink the use of inner-product similarity measure (i.e. $\boldsymbol{\Xi}^\mathsf{T}\mathbf{x}$ in (1.2)), and consider it as primary source of the fuzzy memory problem. Specifically, due to its Euclidean nature, inner-product assigns equal importance to all dimensions of patterns and yields small $\Delta_\mu - 2mR$ if they ($\boldsymbol{\xi}_\mu$ and some $\boldsymbol{\xi}_\nu$) share similar direction. This motivate us to replace the overlap (inner product) construction of the energy function with a similarity measure utilizing this $\boldsymbol{\Xi}$-dependence.

To this end, we propose a kernalized similarity measure for all modern Hopfield models, named U-Hop. This measure is learnable. We propose to learn it by minimizing the average separation among all possible stored memory pairs in set $\boldsymbol{\Xi}$. Namely, it is $\boldsymbol{\Xi}$-sensitive. Physically, it converts the original energy landscape into a *kernalized* landscape with (on average) equally separated minima. While it does not provably guarantee enlarging R (the minimal separation

[1]Recall that, Given a function $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$. A generalized fixed point of $\mathcal{T}$ is a point $\mathbf{x} \in \mathbb{R}^d$ for which $\mathbf{x} \in \mathcal{T}(\mathbf{x})$.

among $\{\boldsymbol{\xi}_\mu\}_{\mu\in[M]}$) in the kernel space, U-Hop addresses the root cause of the fuzzy memory problem with strong empirical evidence. It delivers a larger memory capacity and a tighter retrieval error bound for modern Hopfield models, surpassing all existing modern Hopfield models (Hu et al., 2023; Wu et al., 2024; Ramsauer et al., 2020; Krotov and Hopfield, 2016) and SOTA similarity measures, i.e. ℓ_2-distance and Manhattan distance proposed by Millidge et al. (2022).

Contributions. Our contributions are as follows:

- We introduce a learnable feature map Φ that maps energy E to a kernel space with kernel $\mathcal{K}(\cdot,\cdot) := \langle\Phi(\cdot),\Phi(\cdot)\rangle$. The resulting kernelized energy $E_\mathcal{K}$, and its corresponding retrieval dynamics $\mathcal{T}_\mathcal{K}$ satisfy the defining properties of modern Hopfield models: convergence between local minima of E and fixed points of retrieval dynamics $\mathcal{T}$. This allows us to construct a separation loss $\mathcal{L}_\Phi$ that distinguishes the local minima of $E_\mathcal{K}$ by separating stored memory patterns in kernel space.
- Methodologically, we introduce Uniform Hopfield Memory Retrieval (U-Hop). It is a two-stage optimization formulation. The first stage is separation loss $\mathcal{L}_\Phi$ minimization, distancing stored memory patterns in kernel space. The second stage performs energy minimization with the kernel-induced $\mathcal{T}_\mathcal{K}$. The first stage enhanced $E_\mathcal{K}$, making it able to relocate its local minima to a more separated coordinate. As a result, modern Hopfield models under U-Hop is able to obtain improved memory capacity.
- Empirically, U-Hop improves memory retrieval outcomes by a large margin comparing to other baselines. When applied to deep learning scenarios, U-Hop significantly improves model's memorization capacity, generalization and convergence speed. We show that U-Hop improves memory retrieval tasks by an average 30% margin even under a single iteration of separation minimization, and learning tasks by an average 3% margin.

Organization. Section 2 presents U-Hop. Section 3 connects U-Hop to deep learning. Section 4 conducts extensive numerical experiments to support U-Hop. Appendix includes proofs, experimental details, and additional experimental studies.

Notations. Bold lower case letters denote vectors and bold upper case letters denote matrices. We write $\langle\mathbf{a},\mathbf{b}\rangle := \mathbf{a}^\mathsf{T}\mathbf{b}$ as the inner product for vectors $\mathbf{a},\mathbf{b}\in\mathbb{R}^d$. The index set $\{1,...,I\}$ is denoted by $[I]$, where $I\in\mathbb{N}^+$. The spectral norm is denoted by $\|\cdot\|_2$ which is equivalent to the ℓ_2-norm when applied to a vector. Throughout this paper, we denote the memory patterns (keys) by $\boldsymbol{\xi}\in\mathbb{R}^d$ and the state/configuration/query pattern by $\mathbf{x}\in\mathbb{R}^d$, and $\boldsymbol{\Xi} := (\boldsymbol{\xi}_1,...,\boldsymbol{\xi}_M)\in\mathbb{R}^{d\times M}$ as shorthand for stored memory (key) patterns $\{\boldsymbol{\xi}_\mu\}_{\mu\in[M]}$. We set norm $n := \|\mathbf{x}\|$ to be the norm of the query pattern, and $m := \max_{\mu\in[M]}\|\boldsymbol{\xi}_\mu\|$ be the largest norm of memory patterns. We also provide a nomenclature table (Appendix A) in the appendix.

2. U-Hop: Retrieval as Two-Stage Optimization

In this section, Section 2.1 introduces a learnable feature map that maps patterns and the energy function into a kernel space, and demonstrate the fixed-point convergence property of kernelized modern Hopfield models. Section 2.2 presents U-Hop (Algorithm 1), a two-stage algorithm for the kernel learning with optimal theoretical guarantees. It maximizes pattern separation by minimizing a novel Separation Loss.

2.1. Kernelized Memory Hopfield Energy

In this section, we first parameterize the similarity measure(s) in modern Hopfield model(s) with a learnable kernel (via feature map (2.1)), and then show the induced models (with energy (2.2)) satisfying the defining properties of modern Hopfield models (Theorem 2.1, Lemma 2.1).

Let $\mathcal{K}(\cdot,\cdot) := \langle\Phi(\cdot),\Phi(\cdot)\rangle : \mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}_+$ be the kernel for the given feature mapping $\Phi:\mathbb{R}^d\to\mathbb{R}^{D_\Phi}$ with $D_\Phi\gg d$. In this work, we consider the linear affine feature map: for any $\mathbf{u},\mathbf{v}\in\mathbb{R}^d$,

$$\Phi(\mathbf{u}) := \mathbf{W}\mathbf{u},\quad \text{with}\quad \mathbf{W}\in\mathbb{R}^{D_\Phi\times d}, \tag{2.1}$$

such that $\mathcal{K}(\mathbf{u},\mathbf{v}) = \mathbf{u}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{v}$. Moreover, we shorthand $\{\mathcal{K}(\boldsymbol{\xi}_\mu,\mathbf{x})\}_{\mu=1}^M\in\mathbb{R}^M$ with $\mathcal{K}(\boldsymbol{\Xi},\mathbf{x})\in\mathbb{R}^M$. With (2.1), we introduce the Kernelized Memory Hopfield Energy

$$E_\mathcal{K}(\mathbf{x}) = \mathcal{K}(\mathbf{x},\mathbf{x})/2 - \Psi_\alpha^\star\left(\beta,\mathcal{K}(\boldsymbol{\Xi}^\mathsf{T}\mathbf{x})\right), \tag{2.2}$$

where $\Psi_\alpha^\star$ is the convex conjugate of the Tsallis entropic regularizer introduced in (Wu et al., 2024; Hu et al., 2023): for any $\mathbf{z}=\mathcal{K}(\boldsymbol{\Xi}^\mathsf{T}\mathbf{x})\in\mathbb{R}^M$,

$$\begin{cases}\Psi^\star_{\alpha=1}(\beta,\mathbf{z}) = \mathrm{lse}(\beta,\mathbf{z}),\\ \Psi^\star_{\alpha=2}(\beta,\mathbf{z}) = \frac{1}{2}\|\beta\mathbf{z}\|^2 - \frac{1}{2}\|\mathbf{z}^\star-\beta\mathbf{z}\|^2+\frac{1}{2},\\ \Psi^\star_{\alpha\in[1,2]}(\beta,\mathbf{z}) = \int \mathrm{d}\mathbf{z}\ \alpha\text{-EntMax}(\beta\mathbf{z}),\end{cases} \tag{2.3}$$

with $\mathbf{z}^\star = \mathrm{Sparsemax}(\beta,\mathbf{z})$. Appendix D.1 includes the definitions of α-EntMax and Sparsemax.

Assumption 2.1. $\mathbf{W}\in\mathbb{R}^{D_\Phi\times d}$ with $D_\Phi\gg d$ is full-rank.

Remark 2.1. This assumption is practical and implies that $\mathbf{A}=\mathbf{W}^\mathsf{T}\mathbf{W}\in\mathbb{R}^{d\times d}$ is non-singular. In practice, initializing the weights randomly and independently with a continuous distribution (e.g., Gaussian) makes it almost impossible for $\mathbf{W}$ to be non-full rank, especially if $D_\Phi\gg d$.

Theorem 2.1 (Retrieval Dynamics)**.** With Assumption 2.1, the energy function $E(\mathbf{x})$ was monotonically decreased by

the following retrieval dynamics:

$$\mathcal{T}_{\mathcal{K}}(\mathbf{x}) = \boldsymbol{\Xi} \cdot \mathrm{Sep}_{\alpha}(\beta, \mathcal{K}(\boldsymbol{\Xi}, \mathbf{x})), \tag{2.4}$$

where $\mathrm{Sep}_{\alpha=1}(\cdot) = \mathrm{Softmax}(\cdot)$, $\mathrm{Sep}_{\alpha=2}(\cdot) = \mathrm{Sparsemax}(\cdot)$ and $\mathrm{Sep}_{\alpha\in[1,2]}(\cdot) = \alpha\text{-EntMax}(\cdot)$.

Proof Sketch. By Assumption 2.1 and the convexity of $\mathcal{K}$, there exists an inverse map that transforms the CCCP results in kernel space back to the state space, where $\mathbf{x}$ and $\boldsymbol{\xi}_\mu$ are located. We then complete the proof using the Concave-Convex Procedure (CCCP) and the convex conjugate construction following (Hu et al., 2023; Wu et al., 2024). See Appendix E.2 for a detailed proof. □

The introduction of $\mathcal{K}$ releases similarity measure from Euclidean inner-product to a learnable form via the weight $\mathbf{W}$ of the features map Φ. Moreover, the new Hopfield model ((2.2) and (2.4)) includes all deep learning compatible existing modern Hopfield models (Hu et al., 2023; Wu et al., 2024; Ramsauer et al., 2020). If we replace the kernel $\mathcal{K}(\cdot,\cdot)$ with inner-product $\langle\cdot,\cdot\rangle$, then (2.2) reduces back to the general sparse model Hopfield model (Wu et al., 2024)[2].

While Theorem 2.1 guarantees the monotonic minimization of energy using $\mathcal{T}$, the fixed point of $\mathcal{T}$ might not be the local minima of $E(\mathbf{x})$ according to Sriperumbudur and Lanckriet (2009). Therefore, we provide the next lemma to ensure their alignment, following (Hu et al., 2023; Wu et al., 2024; Ramsauer et al., 2020; Sriperumbudur and Lanckriet, 2009).

Lemma 2.1 (Convergence on retrieval dynamics $\mathcal{T}_{\mathcal{K}}$)**.** Given the energy function $E(\mathbf{x})$ Equation (2.2) and retrieval dynamics $\mathcal{T}_{\mathcal{K}}(\mathbf{x})$ Equation (2.4), respectively. For any sequence $\{\mathbf{x}_t\}_{t=0}^{\infty}$ generated by the iteration $\mathbf{x}_{t'+1} = \mathcal{T}_{\mathcal{K}}(\mathbf{x}_{t'})$, all limit points of this sequence are stationary points of E.

Proof Sketch. By the monotonic energy minimization property of $\mathcal{T}$ (Theorem 2.1) along with (Hu et al., 2023, Lemma 2.2), we prove this through Zangwill's global convergence theory (Zangwill, 1969; Sriperumbudur and Lanckriet, 2009). See Appendix E.1 for a detailed proof. □

In summary, with Φ, the parameterized similarity measure $\mathcal{K}$ introduces an additional degree of freedom for us to relocate the minima of energy landscape $E_{\mathcal{K}}$. We show that the Uniform Memory Hopfield Energy (2.2) and its induced retrieval dynamics (2.4) satisfies the defining properties of modern Hopfield models (Theorem 2.1 and Lemma 2.1). Importantly, Lemma 2.1 states that minimizing the energy E with $\mathcal{T}$ also leads to convergence to the fixed point of $\mathcal{T}$.

[2]Recall that the general sparse Hopfield model encompasses both dense (Ramsauer et al., 2020) and sparse (Hu et al., 2023) models as its special cases.

Algorithm 1 `U-Hop`: Two-Stage Memory Retrieval

Input: Separation (Stage I) iterations N, Energy (Stage II) iteration T, feature map $\Phi(\mathbf{x}) \coloneqq \mathbf{W}\mathbf{x}$, memory set $\boldsymbol{\Xi}$, query $\mathbf{x}$, retrieval dynamics $\mathcal{T}$, learning rate $\gamma \leq 1/G$ where G is the Lipschitz constant of $\mathcal{L}_\Phi(\boldsymbol{\Xi})$
Output: $\mathbf{x}$

1: **for** $i = 1, \ldots N$ **do**
2: $\quad \mathbf{W} \leftarrow \mathbf{W} - \gamma \cdot \nabla_{\mathbf{W}} \mathcal{L}_\Phi(\boldsymbol{\Xi})$. // Stage I
3: **end for**
4: Normalize the rows of $\mathbf{W}$
5: $\mathbf{x}^0 \leftarrow \mathbf{x}$
6: **for** $t = 1, \ldots T$ **do**
7: $\quad \mathbf{x} \leftarrow \mathcal{T}_{\mathcal{K}}(\mathbf{x})$ using Theorem 2.1 // Stage II
8: **end for**
9: return $\mathbf{x}$

This is pivotal in motivating our next step: constructing a separation loss $\mathcal{L}_\Phi$. This loss distinguishes the local minima of $E_{\mathcal{K}}$ by separating stored memory patterns in the kernel space. With $\mathcal{L}_\Phi$, we then formulate the memory retrieval dynamics of the modern Hopfield associative memory model as a two-stage optimization, termed `U-Hop`. This includes an additional stage of separation maximization (by learning the kernel), significantly enhancing memory capacity.

We first extend the standard notion of storage and retrieval (Definition 1.1) literature using kernelized features ($\Phi(\mathbf{x}) \in \mathbb{R}^{D_\Phi}$) to replace states of the model ($\mathbf{x} \in \mathbb{R}^d$.)

Definition 2.1 (Pattern Stored and Retrieved)**.** For all $\mu \in [M]$, let $R_\Phi \coloneqq \frac{1}{2}\min_{\nu\neq\mu;\nu,\mu\in[M]} \|\Phi(\boldsymbol{\xi}_\mu) - \Phi(\boldsymbol{\xi}_\nu)\|$ be the finite radius of each (kernelized) sphere $\mathcal{S}_{\Phi,\mu}$ centered at (kernelized) memory pattern $\Phi(\boldsymbol{\xi}_\mu)$. We say $\boldsymbol{\xi}_\mu$ is *stored* if there exists a generalized fixed point of $\mathcal{T}_{\mathcal{K}}$, such that $\Phi(\mathbf{x}^\star_\mu) \in \mathcal{S}_{\Phi,\mu}$, to which all limit points $\Phi(\mathbf{x}) \in \mathcal{S}_{\Phi,\mu}$ converge to, and $\mathcal{S}_{\Phi,\mu} \cap \mathcal{S}_{\Phi,\nu} = \emptyset$ for $\nu \neq \mu$. We say $\boldsymbol{\xi}_\mu$ is *ϵ-retrieved* by $\mathcal{T}_{\mathcal{K}}$ with $\mathbf{x}$ for an error ϵ.

2.2. Separation Loss and `U-Hop`

In this section, we first introduce a separation loss $\mathcal{L}_\Phi$ (Definition 2.2) over the stored memory set $\boldsymbol{\Xi}$. Minimizing $\mathcal{L}_\Phi$ results in the separation of stored patterns within any given $\boldsymbol{\Xi}$. Consequently, we incorporate this separation-maximization step into the standard memory retrieval process ((2.4)), leading to a novel two-stage formulation/algorithm, `U-Hop`, for memory retrieval (Algorithm 1).

For any $\Phi(\mathbf{u}), \Phi(\mathbf{v}) \in \mathbb{R}^{D_\Phi}$ and some $t > 0$, let

$$\mathcal{G}_t(\Phi(\mathbf{u}), \Phi(\mathbf{v})) \coloneqq \exp\Big\{-t\|\Phi(\mathbf{u}) - \Phi(\mathbf{v})\|_2^2\Big\},$$

be the Radial Basis Function (Φ-RBF) kernel $\mathcal{G}_t : \mathbb{R}^{D_\Phi} \times \mathbb{R}^{D_\Phi} \to \mathbb{R}_+$. We introduce the objective for learning the feature map $\Phi : \mathbb{R}^d \to \mathbb{R}^{D_\Phi}$ (defined in (2.1)) over the

memory set $\Xi = \{\boldsymbol{\xi}_\mu\}_{\mu\in[M]}$.

Definition 2.2 (Average Separation Loss)**.** Given a stored memory set Ξ, and a feature map $\Phi : \mathbb{R}^d \to \mathbb{R}^{D_\Phi}$, the separation loss of the function Φ is

$$\mathcal{L}_\Phi(\Xi; t) \coloneqq \log \mathop{\mathbb{E}}_{\mathbf{u},\mathbf{v}\sim\Xi} \left[\mathcal{G}_t\left(\Phi(\mathbf{u}), \Phi(\mathbf{v})\right)\right],\ t > 0.$$

$\mathcal{L}_\Phi$ indicates the logarithm of average Gaussian separation of Φ vector pairs over Ξ. Naturally, minimization of $\mathcal{L}_\Phi$ leads to an on-average dissimilarity among kernelized memory patterns, i.e., $\{\Phi(\boldsymbol{\xi}_\mu)\}_{\mu\in[M]}$. Notably, $\mathcal{L}_\Phi$ is convex by design and hence exists an optimizer $\mathbf{W}^\star$ the maximizes the average distance between all possible memory pattern pairs.

For later convenience, we also denote the logarithm of Φ-RBF distance of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ as

$$\ell_\Phi(\mathbf{u}, \mathbf{v}) = \log \mathcal{G}_t\left(\Phi(\mathbf{u}), \Phi(\mathbf{v})\right),\ t > 0. \tag{2.5}$$

It has a naive upper bound: $\ell_\Phi(\mathbf{u}, \mathbf{v}) \leq 0$. The upper bound 0 happens only when $\mathbf{u} = \mathbf{v}$ or Φ outputs a fixed feature vector.

Now we introduce Algorithm 1, the Uniform Memory Retrieval `U-Hop`, for learning a suitable kernel and then retrieving stored memory from the learned kernel space.

Algorithm 1 is a 2-stage optimization process. For the first stage, we run N iterations of kernel learning to minimize the separation loss, thus resulting in larger Δ_μ for $\mu \in [M]$. Next, we rescale each row of the affine matrix $\mathbf{W}$ to ensure the magnitude remains the same for memory patterns. For the second stage, we run T update steps for the retrieval dynamics, thus resulting in Hopfield energy minimization. Note that the learned kernel results in a new energy landscape of E, and is expected to encode memory patterns into local minima that separates from all other memory patterns.

2.3. Exact Memory Retrieval

Let $\mathbf{x}^\star$ be fixed points of $\mathcal{T}$. By Definition 2.1 and Hu et al. (2023, Definition 2.2), the retrieval error exhibits a naive bound

$$\|\mathcal{T}_\mathcal{K}(\mathbf{x}) - \boldsymbol{\xi}_\mu\| \leq \max\{\|\mathbf{x} - \boldsymbol{\xi}_\mu\|, \|\mathbf{x}^\star - \boldsymbol{\xi}_\mu\|\}.$$

The $\|\mathbf{x}^\star - \boldsymbol{\xi}_\mu\|$ term forbids the exact memory retrieval. Explicitly, exact memory retrieval requires the memory pattern to be the fixed point of $\mathcal{T}$, namely $\|\mathbf{x}^\star - \boldsymbol{\xi}_\mu\| = 0$. With this observation, we deduce the condition of exact retrieval

$$\mathrm{Sep}\left(\beta, \mathcal{K}\left(\Xi, \boldsymbol{\xi}_\mu\right)\right) = \mathbf{e}_\mu, \tag{2.6}$$

where $\mathbf{e}_\mu$ is the one-hot vector with the μ-th element as 1. By plugging $\boldsymbol{\xi}_\mu$ into $\mathcal{T}(\cdot)$, we see it is a fixed point $\mathcal{T}(\boldsymbol{\xi}_\mu) = \boldsymbol{\xi}_\mu$ and retrieves the target memory $\boldsymbol{\xi}_\mu$ only when (2.6) holds. In the standard modern Hopfield model (utilizing the $\mathrm{Softmax}$ Sep function), the inability of $\mathrm{Softmax}$ to satisfy (2.6) results in a lack of exact retrieval (Martins et al., 2023), thereby preventing the modern Hopfield network from converging to a single memory pattern.

To combat this, we show `U-Hop` achieves exact memory retrieval when $\alpha > 1$, based on the sparse extensions of modern Hopfield model (Wu et al., 2024; Hu et al., 2023; Martins et al., 2023). Specifically, we study the application of `U-Hop` with α-EntMax as separation when $\alpha > 1$.

Theorem 2.2. Let $\mathcal{T}_{\text{sparse}}$ be $\mathcal{T}_\mathcal{K}$ from Theorem 2.1 with $\alpha > 1$. Let $\mathcal{T}_{\text{sparse}}$ a real-valued kernel $\mathcal{K}$ with feature map Φ. Let $t > 0, \beta > 0$. Supposed the query $\mathbf{x} \in \mathcal{S}_{\Phi,\mu}$, $\Phi(\boldsymbol{\xi}_\mu)$ is the fixed point of $\mathcal{T}_{\text{sparse}}$ if the following condition is satisfied:

$$\ell_\Phi(\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu) - \max_{\nu,\nu\neq\mu} \ell_\Phi(\boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu) \leq -\frac{2t}{\beta(\alpha-1)}. \tag{2.7}$$

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

From (2.7), minimizing the separation loss gives the benefit of having the memory pattern to be the fixed point of $\mathcal{T}_{\text{sparse}}$. As a result, Sparse and Generalized Sparse Hopfield models (Hu et al., 2023; Martins et al., 2023; Wu et al., 2024) under `U-Hop` further improves the retrieval accuracy. The next corollary is an extension of the above theorem where we observe the condition with respect to the lipschitzness of Φ.

Corollary 2.2.1. Let $L > 0$ be the Lipschitz constant of Φ. Following Theorem 2.2, $\mathcal{T}_\mathcal{K}$ achieves exact memory retrieval if

$$\min_{\nu\in[M],\nu\neq\mu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\| \geq \sqrt{\frac{2}{L^2\beta(\alpha-1)}}.$$

Proof. See Appendix E.3 for a detailed proof. Note that with Φ defined in (2.1), Φ is always L-Lipschitz. □

3. Connecting to Modern Deep Learning

To incorporate `U-Hop` into deep learning, we first introduce a kernelized Hopfield layer. Here we propose a deep learning compatible layer based on `U-Hop` as

$$\texttt{U-Hop}(\Xi, \boldsymbol{X}) = \mathrm{Sep}\left(\beta\mathbf{W}_K\Phi(\Xi)\mathbf{W}_Q\Phi(\mathbf{x})\right)\mathbf{W}_V\mathbf{W}_K\Xi.$$

Note that this is a kernelized version of `Hopfield` (Ramsauer et al., 2020), `SparseHopfield` (Hu et al., 2023) and `GSH` (Wu et al., 2024) layers, which serve as an alternative to attention mechanism variants.

Next, we introduce the average separation loss for deep learning compatible `U-Hop`.

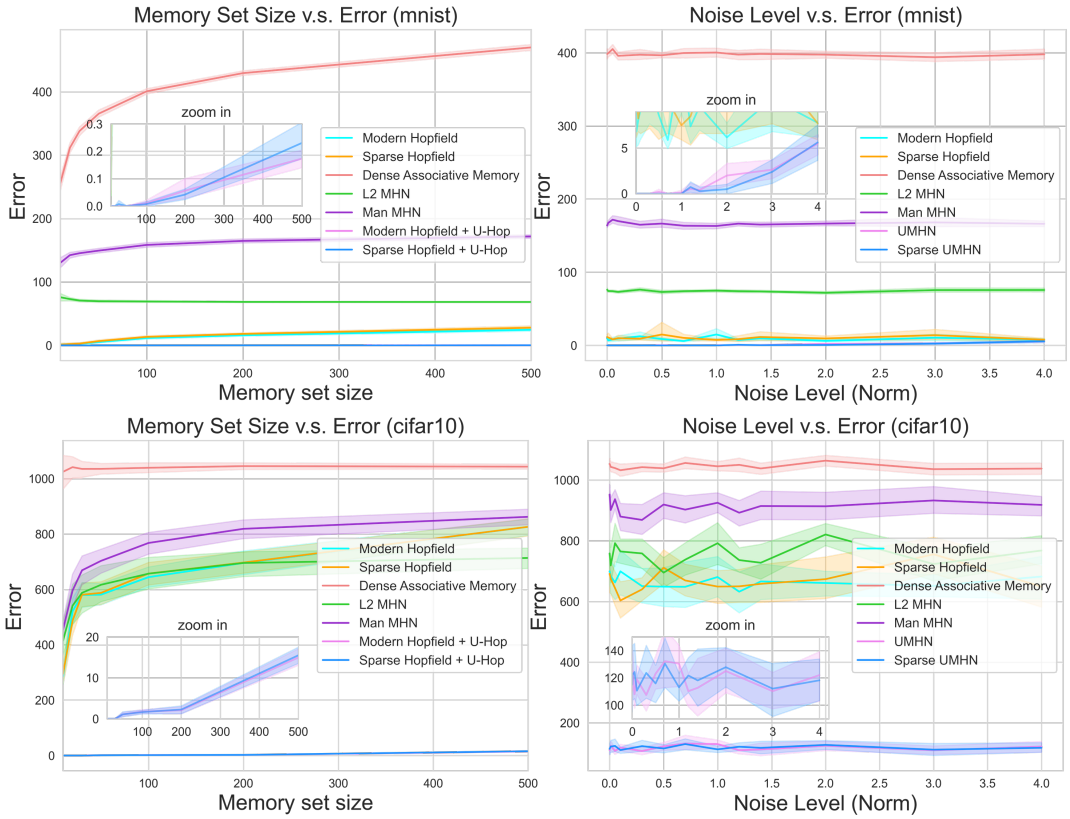

Figure 3. **Memory Retrieval Error Comparison (Section 4.1: Memory Capacity & Noise Robustness).** We conduct memory retrieval experiments on the MNIST and CIFAR10 datasets. For the "Memory Set Size v.s. Error" plots, we vary the memory set size for retrieval. For the "Noise Level v.s. Error" plots, we randomly sample Gaussian noise and rescale the norm of the noise w.r.t. different noise levels. All four plots show U-Hop retrieved patterns with significantly less error compared to all existing Hopfield models across all sizes of memory and noise levels.

Definition 3.1 (Separation Loss for DL)**.** Given a stored memory set Ξ, and a feature map $\Phi : \mathbb{R}^d \to \mathbb{R}^{D_\Phi}$, the deep learning compatible separation loss of the function Φ is defined as

$$\bar{\mathcal{L}}_\Phi(\Xi; t) := \log \mathop{\mathbb{E}}_{\mathbf{u},\mathbf{v}\sim\Xi} \left[\exp\left\{2t\left[\mathcal{K}(\mathbf{u},\mathbf{v})^2 - 1\right]\right\}\right] \quad t > 0.$$

Let the pairwise distance of $\bar{\mathcal{L}}_\Phi$ for any given $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ be

$$\bar{\ell}_{\bar{\Phi}}(\mathbf{u},\mathbf{v}) = 2t\left[\bar{\mathcal{K}}_{\bar{\Phi}}(\mathbf{u},\mathbf{v}) - 1\right], t > 0,$$

with $\bar{\mathcal{K}}(\cdot,\cdot) := \left\langle \bar{\Phi}(\cdot), \bar{\Phi}(\cdot) \right\rangle$ for some $\bar{\Phi} : \mathbb{R}^d \to \mathbb{R}^{D_{\bar{\Phi}}}$. We present next the theorem for the expressiveness of `U-Hop`.

Theorem 3.1 (Kernelized Representation Theorem)**.** Let $\bar{\Phi}$ be a feature map such that $\bar{\Phi} := \mathbb{R}^d \to \mathbb{R}^{D_{\bar{\Phi}}}$, and $\bar{\mathcal{K}}$ be $\bar{\Phi}$-induced kernel. Assuming $\bar{\mathcal{K}}$ satisfies: $\bar{\ell}_{\bar{\Phi}}(\mathbf{u},\mathbf{v}) = -2t$ for any given $\mathbf{u}, \mathbf{v} \in \Xi$. With $\beta > 0$, input $\mathbf{X} \in \mathbb{R}^{d\times M}$, $M \leq d$, an arbitrary positive column stochastic matrix $\mathbf{P} \in \mathbb{R}^{M\times M}$, there always exists matrices $\mathbf{W}_Q, \mathbf{W}_K$ such that

$$\text{Softmax}\left(\beta\left(\mathbf{W}_K\bar{\Phi}(\mathbf{X})\right)^\top \mathbf{W}_Q\bar{\Phi}(\mathbf{X})\right) = \mathbf{P}.$$

Specifically, $\bar{\mathcal{K}}$ is the loss minimizer of $\bar{\mathcal{L}}_\Phi(\Xi; t)$.

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

The empirical validation is in Appendix G. This theorem shows that with a suitable kernel, the expressiveness of `Hopfield` layers under `U-Hop` reaches its full potential. The main difference between this new loss function and the separation loss is the square on $\mathcal{K}(\mathbf{u},\mathbf{v})$. Note that this theorem requires $\bar{\mathcal{K}}(\mathbf{u},\mathbf{v}) = 0$ for any given $\mathbf{u}, \mathbf{v} \in \Xi$, $\mathbf{u} \neq \mathbf{v}$, which implies it is only possible when $d \geq M$. In the context of deep learning, the patch size must not be larger than the hidden dimension to realize this result. This theorem extends the representation theorem in (Bhojanapalli et al., 2020) to a practical setting, showing that $\bar{\mathcal{K}}$ overcomes the low-rank bottleneck of the attention mechanism and Hopfield layer as well.

The next algorithm is the realization of searching for $\bar{\mathcal{K}}$ under supervised learning schema. Consider a supervised learning problem with input data $\mathcal{X} = \{\mathbf{X}_1, ..., \mathbf{X}_n\}$, label $\mathcal{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_n\}$, model $F := \mathcal{X} \to \mathcal{Y}$, where F consists

of one layer of "U-Hop + Hopfield layer". The stage-I of U-Hop is parameterized by θ, and F is parameterized be θ_F.

Algorithm 2 U-Hop for Learning

Input: Data $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, Iteration number N_i, N_o, model: $F : \mathcal{X} \to \mathcal{Y}$, step sizes $(\overline{\gamma}, \overline{\alpha})$, training objective $\mathcal{L}$, Stage I SGD batch size B_1, Stage II SGD batch size B_2

for $i = 1$ to N_o **do**
  for $j = 1$ to N_i **do**
    Sample mini-batch $\mathbf{X} \sim \mathcal{X}$
    $\theta^i = \theta^{i-1} - \overline{\gamma}\widehat{\nabla}\overline{\mathcal{L}}_\Phi(x)$. // SGD with B_1
  end for
  Sample mini-batch $\mathbf{X}, \mathbf{Y} \sim \mathcal{X}, \mathcal{Y}$
  $\theta_F^j = \theta_F^{j-1} - \overline{\alpha}\widehat{\nabla}\mathcal{L}(x, y)$ // SGD with B_2
end for
return F

4. Experimental Studies

To validate the efficacy of U-Hop, we test it on both associative memory retrieval task and deep learning task (image classification) with multiple real world datasets.

4.1. Memory Retrieval

Memory Capacity. The memory retrieval task involves retrieving a memory pattern from a stored memory set. In particular, this experiment aim to reconstruct memories based on a query. The query is generated by randomly masking 50% of pixels in the target image. We compare our method against several modern Hopfield models (Hu et al., 2023; Ramsauer et al., 2020; Krotov and Hopfield, 2016). We also vary the iteration number N for the first stage in Algorithm 1. We use MNIST, CIFAR10 datasets for this task. Please see Appendix F for experimental details.

Noise Robustness. This experiment follows the same procedure as the memory capacity tasks, but with multiple levels of injected noise on the target image instead of masking out pixels to generate queries. We use Gaussian noise to contaminate the queries and vary the noise level by altering the mean of the Gaussian vectors. As the noise level increases, it becomes more difficult to retrieve the memory with low error. A higher noise level results in greater difficulty in achieving low-error retrieval. We use MNIST, CIFAR10 for this task. Please see Appendix G for experimental details.

Baselines. We compare our method with Modern Hopfield Model (Ramsauer et al., 2020), Sparse Modern Hopfield network (Hu et al., 2023), Dense Associative Memory (Polynomial Hopfield) (Krotov and Hopfield, 2016) (using 10-th order polynomial energy function). We also compare U-Hop with existing similarity measures: L2 distance (ℓ_2 MHM) and Manhattan distance (Man. MHM) (Millidge et al., 2022).

Setting and Metrics. We set $\beta = 1, t = 2$ across all the memory retrieval experiments. For the evaluation metric, we follow (Hu et al., 2023; Millidge et al., 2022) to use the Sum-of-Square pixel differences between the ground truth image and the retrieved image.

Results. See Figure 3 for results of memory capacity and noise robustness, Figure 5 for results of "Stage I iteration improve retrieval error" and Appendix G.1 for the relationship between separation loss and retrieval error.

- For **memory capacity**, U-Hop outperforms all other baselines by a large margin. This result shows the retrieval dynamics under U-Hop is near optimal across all memory set sizes. Next, we vary the iteration N to observe how fast the retrieval error decreases as the N goes up. In Figure 5, we show a strong correlation between N and retrieval error.
- For **noise robustness**, U-Hop shows strong performance against all baselines as well as showed in Figure 3.

4.2. Supervised Learning Tasks

Image Classification. For classification tasks, we compare our method against Hopfield (Ramsauer et al., 2020) and SparseHopfield (Hu et al., 2023). We test two settings:

- U-Hop + Dense Modern Hopfield Model (Ramsauer et al., 2020), and
- U-Hop + Sparse Modern Hopfield Model (Hu et al., 2023).

We vary the training sample size and observe model performance. We focus on (i) convergence speed (speed of loss decay), (ii) generalization power (test accuracy). We use CIFAR10, CIFAR100 and TinyImageNet for this task. Please see Appendix F for more experimental details.

We use the following Hopfield layer (Ramsauer et al., 2020) to replace the self-attention mechanism in Vision Transformer:

$$\begin{aligned}&\texttt{Hopfield}\left(\boldsymbol{X}\right)\\ &= \mathbf{W}_V\mathbf{W}_K \cdot \mathrm{Softmax}\left(\beta\mathbf{W}_K\Phi(\mathbf{X})\mathbf{W}_Q\Phi(\mathbf{X})\right),\end{aligned}$$

where $\mathbf{W}_K, \mathbf{W}_Q$ are the same as in self-attention, and $\mathbf{W}_V \in \mathbb{R}^{h\times h}$, where h is the hidden dimension.

Expressiveness. To verify Theorem 3.1, we evaluate how many samples a model can memorize in supervised learning task. We follow the image classification settings, and see how Hopfield models with and without U-Hop react to sample size growth.

Table 1. **Model maximal training accuracy and test accuracy with and without `U-Hop` on CIFAR10, CIFAR100 and Tiny ImageNet (Section 4.2: Supervised Learning Tasks).** MHM denotes Modern Hopfield Model (Ramsauer et al., 2020). We omit variance as all variance are $\leq 0.03\%$. The result demonstrates with `U-Hop`, models are able to consistently memorize more samples in the training data, and further obtain generalization improvement. Note that the improvement on Max. Training accuracy is a validation of Theorem 3.1. In Appendix G, we also show `U-Hop` allows modern Hopfield models to converge faster.

Models	CIFAR10		CIFAR100		Tiny ImageNet	
	Max Train Acc.	Test Acc.	Max Train Acc.	Test Acc.	Max Train Acc.	Test Acc.
MHM	56.0%	52.2%	32.3%	26.3%	48.9%	12.2%
MHM + `U-Hop`	**64.6%**	**55.2%**	**44.1%**	**28.7%**	**61.4%**	**12.7%**
Sparse MHM	55.9%	52.0%	49.6%	26.0%	17.2%	12.3%
Sparse MHM + `U-Hop`	**66.4%**	**55.4%**	**45.4%**	**29.0%**	**60.6%**	**12.5%**

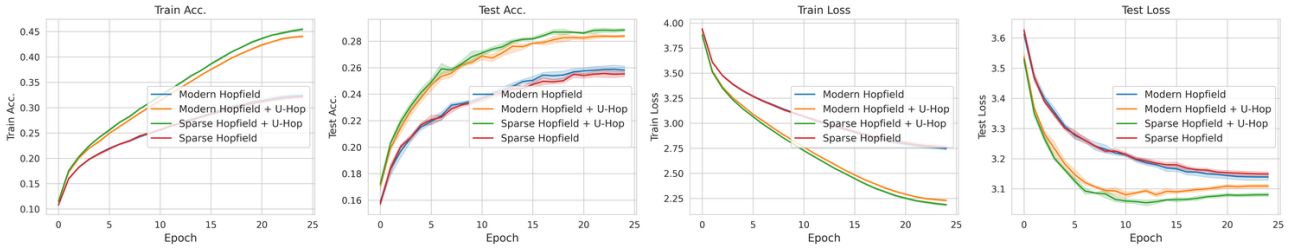

Figure 4. **Model Convergence Comparison with and without `U-Hop` on CIFAR100 (Section 4.2: Image Classification Task).** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss. Yellow and green curves represent modern Hopfield + `U-Hop` and Sparse modern Hopfield + `U-Hop`. Blue and red curves represent modern Hopfield and Sparse modern Hopfield. The result demonstrates without `U-Hop`, Hopfield layers fall into the low-rank bottleneck (Bhojanapalli et al., 2020) despite of high embedding dimension. On the other hand, `U-Hop` successfully avoid such issue and thus have better training accuracy. For generalization power and convergence speed, `U-Hop` also outperforms other baselines by a large margin. For other datasets and sample size, we leave the results in Appendix G.

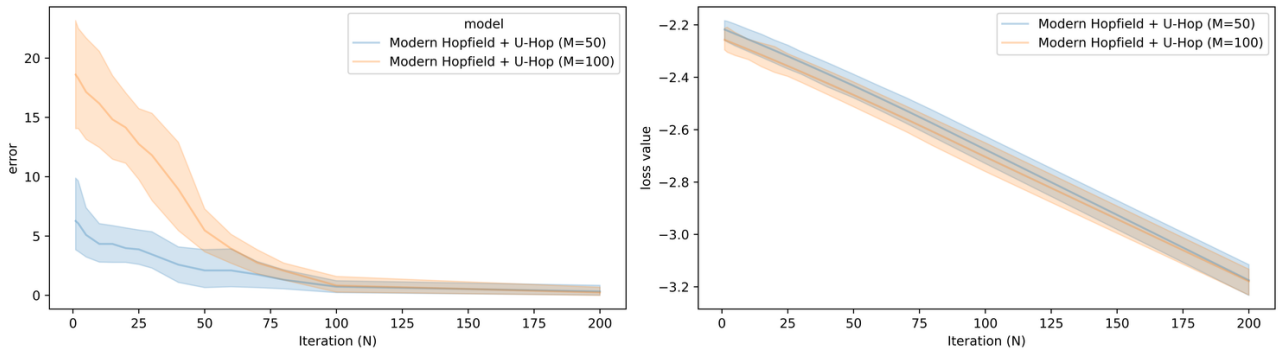

Figure 5. **Retrieval Error v.s. Separation-Maximization (Stage I of Algorithm 1) Iteration N (Section 4.1).** We vary the iteration number N and perform memory retrieval on `U-Hop` with modern Hopfield. We set $\beta = 1, t = 2$ and report the sum-of-square pixel differences. The result shows the retrieval error decays fast with respect to the increase of N.

Time Series Prediction. We also use the `STanHop-Net` (Wu et al., 2024) as our test-bed and observe the performance change with and without `U-Hop`. For this task, we use ETTh1, ETTm1 and WTH datasets. We use the prediction horizon of $\{96, 192, 336, 720\}$ for all datasets. Please see Appendix F for experimental and hyperparameter details.

Baselines, Setting and Metrics. We compare the performance of Modern Hopfield and Sparse Hopfield with and without `U-Hop`. For image classification, we use Vision Transformer (Dosovitskiy et al., 2020) as test-bed and replace the attention mechanism with `Hopfield` (Ramsauer et al., 2020) and `SparseHopfield` layer (Hu et al., 2023). For time series prediction, we compare the performance of `STanHop-Net` (Hu et al., 2023) with and without `U-Hop`.

Results. See Table 1 for convergence results of image classification task, Figure 6 for expressiveness results (Theorem 3.1) and Table 8 for time series prediction.

- For **image classification**, we observe that modern Hop-

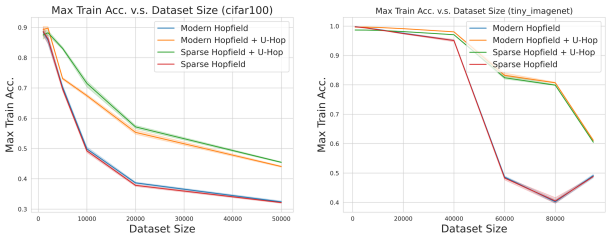

Figure 6. **Maximum Training Accuracy v.s. Training Sample Size (Section 4.2: Expressiveness).** Here we report the train accuracy comparison between modern Hopfield models with and without `U-Hop`. The maximum training accuracy represents how many percentages of samples a model memorizes, which is highly related to model expressiveness and complexity. Note that using `U-Hop` does not increase model complexity, which shows `U-Hop` improves model expressiveness by a large margin.

field models under `U-Hop` consistently outperform other baselines, and the performance gap increases with the sample size growth. Additionally, `U-Hop` models shows superior convergence speed comparing to other baselines on both training and test set. For model generalization, see Table 1, for convergence results, see Appendix G.5.

- For **model expressiveness**, we observe that when the dataset size is small, `U-Hop` has similar memorization capability as `Hopfield`. However, as the dataset size increases, `Hopfield` without `U-Hop` shows a sharp degeneration on training accuracy and struggles to converge well, as evidenced in Figure 6. For more detailed results, see Appendix G.4.
- For **time series prediction**, our results (in Appendix G.6) demonstrate that even on SOTA Hopfield-based time series model, `U-Hop` delivers performance improvement across different datasets and prediction horizons.

4.3. More Discussions on Experimental Results

For memory retrieval tasks, `U-Hop` delivers significant improvements on retrieval error by lowering separation loss over the memory set. For the epochs N required for kernel learning, we demonstrate that low retrieval error has strong correlation with large size of N. This is expected as our separation loss is convex and guaranteed to obtain global optima with a rate of $\mathcal{O}(1/N)$. As showed in Figure 5, separation loss consistently decreased as N goes up.

For classification tasks, `U-Hop` delivers significant improvements in predictive power of the underlying models. Comparing to contrastive self-supervised learning (Wang and Isola, 2020; Chen et al., 2020), where they maximize pairwise distance between samples, `U-Hop` maximizes the pairwise distance between patches. As Saunshi et al. (2022) show maximizing the distance over samples improves class generalization and is beneficial to downstream tasks.

Our experiment results indicate 2 new insights that the separation on the patch/token level also leads to better generalization. Firstly, we hypothesize that the Stage I of `U-Hop` serves as a pre-training step for a better representation with more separated data geometry. Namely, tokens/patches projected to kernel space have higher quality of representation as `U-Hop`'s first iteration leads to better patch separation. Secondly, though " `Hopfield` layers with and without `U-Hop`" and "expressiveness" experiments (Table 1 and Figure 6), we observe that solely increasing embedding dimension do not guarantee to escape from the low-rank bottleneck in attention- and Hopfield-based models (Bhojanapalli et al., 2020). We conclude that this is because these models do not utilize their full expressive power (as in Theorem 3.1), despite of high embedding dimension. This observation supplements the existing "high-dimensional embedding improves low-rank bottleneck" conjecture (Bhojanapalli et al., 2020) with an intuitive yet effective learning scheme.

5. Concluding Remarks

We present a two-stage formulation for memory retrieval of modern Hopfield models, `U-Hop`. Our key contribution is a learnable similarity measure utilizing the stored memory patterns as learning data. Through our analyses, `U-Hop` is theoretically grounded and empirically strong. Experimentally, it improves memory retrieval tasks by an average 30% margin even with only a single separation-maximization iteration and learning tasks by an average 3% margin. These results are benchmarked against STOA similarity measures (ℓ_2- and Manhattan- distance (Millidge et al., 2022)) and existing modern Hopfield models (Wu et al., 2024; Hu et al., 2023; Ramsauer et al., 2020; Krotov and Hopfield, 2016).

Complexity Analysis. Algorithm 1 has a time complexity of $\mathcal{O}(N+T)$. Algorithm 2 has a time complexity of $\mathcal{O}(N_o N_i)$. Although this increases the standard supervised learning training time by a factor of N_i, our experimental results demonstrate that models under `U-Hop` mitigate this issue with a faster convergence speed, requiring fewer epochs to converge. See Appendix G.5 for related empirical results.

Limitation and Future. One notable limitation is that the optimality of separation loss (Definition 2.2) does not guarantee maximal separation for $R := \frac{1}{2}\min_{\mu,\nu\neq\mu\in[M]} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$ for any given $\boldsymbol{\Xi}$. This problem (maximizing R) is inherently a max-min (non-convex) problem and is less straightforward to analyze (Comparison between max and avg. loss is in Appendix G.2). To achieve provably optimal memory capacity, we plan to explore different loss functions or learning schemes in the future.

Impact Statement

This research is theoretical and is not expected to have negative social impacts. As outlined in the introduction and related works, the primary goal of this study is to enhance our understanding of the underlying principles of large Hopfield-based and transformer-based foundation models from an associative memory perspective.

Acknowledgments

JH would like to thank Stephen Cheng, Shang Wu, Dino Feng and Andrew Chen for enlightening discussions, the Red Maple Family for support, and Jiayi Wang for facilitating experimental deployments. The authors would also like to thank the anonymous reviewers and program chairs for their constructive comments.

JH is partially supported by the Walter P. Murphy Fellowship. HL is partially supported by NIH R01LM1372201, NSF CAREER1841569, DOE DE-AC02-07CH11359, DOE LAB 20-2261 and a NSF TRIPODS1740735. This research was supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

References

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

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

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

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

Sergey Bartunov, Jack W Rae, Simon Osindero, and Timothy P Lillicrap. Meta-learning deep energy-based memory models. *Eighth International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1910.02720.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *Thiry-Seventh International conference on machine learning(ICML)*, pages 864–873. PMLR, 2020. URL https://arxiv.org/abs/2002.07028.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization, 2021. URL https://arxiv.org/abs/2109.12948.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023. URL https://arxiv.org/abs/2306.12929.

Thomas F Burns. Semantically-correlated memories in a dense associative model. In *Forty-first International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2404.07123.

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

Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024a. URL https://openreview.net/forum?id=Tzh6xAJSll.

Vivien Cabannes, Berfin Simsek, and Alberto Bietti. Learning associative memories with gradient descent. *arXiv preprint arXiv:2402.18724*, 2024b. URL https://arxiv.org/abs/2402.18724.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17413–17426, 2021a. URL https://arxiv.org/abs/2110.15343.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Thirty-seventh International conference on machine learning (ICML)*, pages 1597–1607. PMLR, 2020. URL https://arxiv.org/abs/2002.05709.

Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with gaussian kernel and nystr\" om method. *Advances in Neural Information Processing*

Systems (NeurIPS), 34:2122–2135, 2021b. URL https://arxiv.org/abs/2111.00035.

Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015*, 2019. URL https://arxiv.org/abs/1909.00015.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Eighth International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/2010.11929.

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

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

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

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

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

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

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

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

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

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

Jerry Yao-Chieh Hu, Thomas Lin, Zhao Song, and Han Liu. On computational limits of modern hopfield models: A fine-grained complexity analysis. In *Forty-first International Conference on Machine Learning (ICML)*, 2024c. URL https://arxiv.org/abs/2402.04520.

Georgios Iatropoulos, Johanni Brea, and Wulfram Gerstner. Kernel memory networks: A unifying framework for memory modeling. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:35326–35338, 2022. URL https://arxiv.org/abs/2208.09416.

Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *Ninth International Conference on Learning Repre- sentations (ICLR)*, 2020. URL https://arxiv.org/abs/2001.04451.

Leo Kozachkov, Ksenia V Kastanenka, and Dmitry Krotov. Building transformers from neurons and astrocytes. *Proceedings of the National Academy of Sciences*, 120(34): e2219150120, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in neural information processing systems (NeurIPS)*, 29, 2016. URL https://arxiv.org/abs/1606.01164.

Dmitry Krotov and John J. Hopfield. Large associative memory problem in neurobiology and machine learning. In *9th*

International Conference on Learning Representations (ICLR), 2021. URL https://arxiv.org/abs/2008.06996.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Andre Martins, Vlad Niculae, and Daniel C McNamee. Sparse modern hopfield networks. In *Associative Memory & Hopfield Networks in 2023*, 2023. URL https://openreview.net/forum?id=zwqlV7HoaT.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal hopfield networks: A general framework for single-shot associative memory models. In *Thirty-ninth International Conference on Machine Learning (ICML)*, pages 15561–15583. PMLR, 2022. URL https://arxiv.org/abs/2202.04557.

Matteo Negri, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, and Enrico Malatesta. Storage and learning phase transitions in the random-features hopfield model. *Physical Review Letters*, 131(25):257301, 2023. URL https://arxiv.org/abs/2303.16880.

Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Conv-coa: Improving open-domain question answering in large language models via conversational chain-of-action. *arXiv preprint arXiv:2405.17822*, 2024. URL https://arxiv.org/abs/2405.17822.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020. URL https://arxiv.org/abs/2008.02217.

Alex Reneau, Jerry Yao-Chieh Hu, Chenwei Xu, Weijian Li, Ammar Gilani, and Han Liu. Feature programming for multivariate time series prediction. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 29009–29029. PMLR, 23–29 Jul 2023. URL https://arxiv.org/abs/2306.06252.

Tommaso Salvatori, Yuhang Song, Yujian Hong, Lei Sha, Simon Frieder, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. Associative memories via predictive coding. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:3874–3886, 2021. URL https://arxiv.org/abs/2109.08063.

Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *Thirty-ninth International Conference on Machine Learning (ICML)*, pages 19250–19286. PMLR, 2022. URL https://arxiv.org/abs/2202.14037.

Rylan Schaeffer, Nika Zahedi, Mikail Khona, Dhruv Pai, Sang Truong, Yilun Du, Mitchell Ostrow, Sarthak Chandra, Andres Carranza, Ila Rani Fiete, et al. Bridging associative memory and probabilistic modeling. *arXiv preprint arXiv:2402.10202*, 2024. URL https://arxiv.org/abs/2402.10202.

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2018.

Kyungwoo Song, Yohan Jung, Dongjun Kim, and Il-Chul Moon. Implicit kernel attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9713–9721, 2021. URL https://arxiv.org/abs/2006.06147.

Bharath K Sriperumbudur and Gert RG Lanckriet. On the convergence of the concave-convex procedure. In *Advances in neural information processing systems (NeurIPS)*, volume 9, pages 1759–1767, 2009. URL https://papers.nips.cc/paper_files/paper/2009/file/8b5040a8a5baf3e0e67386c2e3a9b903-Paper.pdf.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024. URL https://arxiv.org/abs/2402.17762.

Danil Tyulmankov, Ching Fang, Annapurna Vadaparty, and Guangyu Robert Yang. Biological learning in key-value memory networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:22247–22258, 2021. URL https://arxiv.org/abs/2110.13976.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems (NeurIPS)*, 30, 2017. URL https://arxiv.org/abs/1706.03762.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *39th International Conference on Machine Learning (ICML)*, pages 9929–9939. PMLR, 2020.

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

David J Willshaw, O Peter Buneman, and Hugh Christopher Longuet-Higgins. Non-holographic associative memory. *Nature*, 222(5197):960–962, 1969.

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

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

Maria Yampolskaya and Pankaj Mehta. Controlling the bifurcations of attractors in modern hopfield networks. In *Associative Memory {\&} Hopfield Networks in 2023*, 2023.

Jinsoo Yoo and Frank Wood. Bayespcn: A continually learnable predictive coding associative memory. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:29903–29914, 2022. URL https://arxiv.org/abs/2205.09930.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). *Advances in neural information processing systems (NeurIPS)*, 14, 2001.

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

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021. URL https://arxiv.org/abs/2012.07436.

Supplementary Material

A. Table of Notations

Table 2. 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 $\mathbb{R}^d$
δ	Noises in memory patterns in $\mathbb{R}^d$
$\boldsymbol{\Xi}$	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 $\Phi(\cdot) : \mathbb{R}^d \to D_\Phi$
D_Φ	Dimension of the kernel space, i.e., dimension of output of $\Phi(\cdot)$
$\mathbf{W}$	Weighted matrix of the linear affine feature map defined in (2.1) in $\mathbb{R}^{d \times D_f}$
$\mathcal{K}(\cdot, \cdot)$	Kernel function takes the inner product form $\mathcal{K}(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle$: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$
$\mathcal{M}$	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 $m \coloneqq \max_{\mu \in [M]} \|\boldsymbol{\xi}_\mu\|$
R	Minimal Euclidean distance across all possible pairs of memory patterns, denoted as $R \coloneqq \frac{1}{2} \min_{\mu, \nu \in [M]} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$
$\mathcal{S}_\mu$	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., $\mathbf{x}_\mu^\star \in S_\mu$

Table 3. Comparison between uniform memory Hopfield and other existing works.

Model	Overlap Construction	Separation	Adaptivity
Dense Associative Memory (Krotov and Hopfield, 2016)	Dot Product	Polynomial	No
Modern Hopfield Nework (Ramsauer et al., 2020)	Dot Product	Softmax	No
Sparse Modern Hopfield Network (Hu et al., 2023)	Dot Product	Sparsemax	No
`U-Hop +` (Wu et al., 2024; Hu et al., 2023; Ramsauer et al., 2020)	Kernel Function	Not Restricted	Yes

B. Related Work

Hopfield Networks. Associative memory models (Willshaw et al., 1969; Kanerva, 1988) have been widely discussed in both the neuroscience and machine learning fields. The main goal of these models are to store a set of memory patterns where those patterns can be retrieved with respect to a given query. Hopfield models represent a primary category within the class of computational associative memory models (Hopfield, 1982). Starting from the classical Hopfield models (Hopfield, 1982; 1984; Krotov and Hopfield, 2021), these models are able to store and retrieve binary patterns with guaranteed memorization capacity. Their biologically plausible designs provides significant insights to understand both human brains (Yampolskaya and Mehta, 2023; Krotov and Hopfield, 2021) and modern deep learning paradigms (Burns, 2024; Cabannes et al., 2024b;a; Kozachkov et al., 2023; Negri et al., 2023; Ramsauer et al., 2020). Recently, these Hopfield models regain interest in the deep learning field due to its connection to the attention mechanism in transformers. Notably, Ramsauer et al. (2020) propose the Modern Hopfield models (MHMs) whose single-step update is equivalent to the attention mechanism (Vaswani et al., 2017). As a result, this connection (starting from the dense associative memory model (Krotov and Hopfield, 2016)) facilitates the integration of associative memory models into modern deep learning (Hofmann et al., 2024; Hu et al., 2024b; Xu et al., 2024; Wu et al., 2024; Burns and Fukai, 2023; Auer et al., 2024; Widrich et al., 2020) and large foundation models (Hu et al., 2024a; Pan et al., 2024; Fürst et al., 2022).

Theory of Modern Hopfield Models. Beside empirical success, Modern Hopfield Models (MHM) offer a low-assumption theoretical framework for analyzing transformer-based deep learning architectures. Toward their fundamental theory, Hu et al. (2023) and Wu et al. (2024) point out that the energy function of MHM and its sparse variants are actually tied to the convex conjugates of different entropic regularizers. This has led to the Sparse and Generalized Sparse HMHs, which are connected to attention mechanisms with various degrees of sparsity (Correia et al., 2019; Vaswani et al., 2017; Martins and Astudillo, 2016). Extending this foundation, Hu et al. (2024b) further complement this understanding with the principled construction of possible efficient variants from a nonparametric perspective. Furthermore, Hu et al. (2024c) provide a detailed theoretical analysis of all possible efficient variants, through the lens of fine-grained complexity theory.

We would like to comment further on the results of (Hu et al., 2024c). First, it observes that the magnitude of the patterns (i.e., the norms of queries and memories) not only affects retrieval accuracy (as seen in the linear m scaling in (1.3)), but also determines the efficiency of a variant of the modern Hopfield model. This norm-based efficiency criterion, with precision guarantees, echoes the *outlier effect* in the attention heads of transformer models (Hu et al., 2024a). This outlier effect is well-known in pretraining large transformer-based models for its negative impact on model quantization performance (Sun et al., 2024; Bondarenko et al., 2023; 2021). To address this, Hu et al. (2024a) interpret the outlier effect as inefficient *rare* memory retrieval and propose the outlier-efficient Hopfield layer for transformer-based large models, demonstrating strong empirical performance and theoretical guarantees. The benefits of removing outliers in the attention heads of transformer-based large foundation models are also highlighted in (Gu et al., 2024a;b; Alman and Song, 2024a;b; 2023; Gao et al., 2023) from various theoretical perspectives. In this work, the removal of outliers is achieved by the row-wise normalization in `U-Hop` (see line 4 of Algorithm 1).

Learning Associative Memory Models. Another line of research focuses on learning an associative memory model (Tyulmankov et al., 2021; Salvatori et al., 2021) that has the ability to "read" (retrieve) and "write" (store) memories. Particularly, this type of method contains a "readout" network to retrieve/generate memories with a given query. Bartunov et al. (2019) propose a meta learning framework to learn a generative network that treats the retrieval error as their energy function. Yoo and Wood (2022) propose a hierarchical associative memory model that relaxes the requirement of meta learning. Salvatori et al. (2021) propose a hierarchical generative network trained with predictive coding. Instead of deriving the retrieval dynamics from the energy function, these methods normally use a generative model for memory retrieval. With the expressiveness of deep neural networks, such method showed great empirical performances. However, since

the structure of the readout network does not connect or dependent on the energy function, they are not able to preserve appealing theoretical guarantees like Hopfield models.

Kernel Memory Networks. Iatropoulos et al. (2022) propose a kernelized memory network[3]. They formulate the modern Hopfield models with a recurrent SVM model. In particular, their kernel is a single layer feed forward network that is trained to memorize patterns. However, their framework consists of several high assumptions. In comparison, our proposed framework has mild assumption on parameters and pattern distributions. In addition, `U-Hop` has significant practical usage and was validated through extensive experiments in both memory retrieval and supervised learning tasks.

This work bridges two paradigms of associative memory models via a non-singular kernel, such that the kernelized energy function (2.2) still satisfies the defining properties of modern Hopfield models, i.e. attention-included retrieval dynamics (Theorem 2.1). A comparison between Uniform Memory Hopfield and similar models are shown in Table 3.

Kernel Approach in Transformer Attention. The usage of kernels and feature expansions in transformers has been extensively discussed in previous literature. One primary objective of these studies is to reduce the computational complexity associated with attention mechanisms. For instance, Chen et al. (2021b); Kitaev et al. (2020); Chen et al. (2021a) demonstrate empirically and theoretically that these efficient algorithms can effectively approximate SoftMax attention. Song et al. (2021) provides a generalized framework for attention mechanism by decomposing it into two parts, RBF kernel as similarity measure and L_2 norm weighting on tokens. In our paper, we offer a distinct perspective aimed at enhancing memory capacity, drawing inspiration from the construction of the modern Hopfield model. Therefore, our approach differs from attempting to approximate the standard modern Hopfield association. Instead, we focus on relocating memory patterns to facilitate easier retrieval.

[3] In a similar vein, Schaeffer et al. (2024) bridge associative memory models and probabilistic modeling.

C. Connection to Transformer Attentions

Suppose that $\mathbf{X}$ and $\mathbf{\Xi}$ are embedded from the *raw* query $\mathbf{R}$ and $\mathbf{Y}$ memory patterns, respectively, via $\mathbf{X}^\mathsf{T} = \mathbf{R}\mathbf{W}_Q \coloneqq \mathbf{Q}$, and $\mathbf{\Xi}^\mathsf{T} = \mathbf{Y}\mathbf{W}_K \coloneqq \mathbf{K}$, with some projection matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$. Then, taking the transport of $\mathcal{T}$ in (1.2) and multiplying with $\mathbf{W}_V$ such that $\mathbf{V} \coloneqq \mathbf{K}\mathbf{W}_V$, we obtain

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

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

This connection provides a insightful theoretical foundation for the attention mechanism. Specifically, the update step in a Transformer's attention mechanism functions as an inner-loop optimization, minimizing an underlying energy function defined by the queries, keys, and values. Please see (Ramsauer et al., 2020; Hu et al., 2023; Wu et al., 2024).

D. Supplementary Theoretical Backgrounds

D.1. Sparsemax and α-EntMax

Here we quote some known results from (Hu et al., 2023; Martins and Astudillo, 2016).

Let $\mathbf{z}, \mathbf{p} \in \mathbb{R}^M$, and $\Delta^M := \{\mathbf{p} \in \mathbb{R}_+^M \mid \sum_\mu^M p_\mu = 1\}$ be the $(M-1)$-dimensional unit simplex. where α-EntMax$(\cdot) : \mathbb{R}^M \to \Delta^M$ is a finite-domain distribution map defined as follows.

Definition D.1 (α-EntMax)**.** The variational form of α-EntMax is defined by the optimization problem

$$\alpha\text{-EntMax}(\mathbf{z}) := \underset{\mathbf{p}\in\Delta^M}{\operatorname{argmax}}[\langle \mathbf{p}, \mathbf{z}\rangle - \Psi_\alpha(\mathbf{p})], \tag{D.1}$$

where $\Psi_\alpha(\cdot)$ is the Tsallis entropic regularizer given by (D.2).

$$\Psi_\alpha(\mathbf{p}) := \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_{\mu=1}^M \left(p_\mu - p_\mu^\alpha\right), & \alpha \neq 1, \\ -\sum_{\mu=1}^M p_\mu \ln p_\mu, & \alpha = 1, \end{cases}, \quad \text{for } \alpha \geq 1, \tag{D.2}$$

Let $\mathbf{z} \in \mathbb{R}^M$. Denote $[a]_+ := \max\{0, a\}$, $z_{(\nu)}$ the ν'th element in a sorted descending z-sequence $\mathbf{z}_{\text{sorted}} := z_{(1)} \geq z_{(2)} \geq \ldots \geq z_{(M)}$, and $\kappa(\mathbf{z}) := \max\left\{k \in [M] \;\middle|\; 1 + kz_{(k)} > \sum_{\nu\leq k} z_{(\nu)}\right\}$. Sparsemax$(\cdot)$ is defined as (Proposition 1 of (Martins and Astudillo, 2016)) :

$$\text{Sparsemax}(\mathbf{z}) = [\mathbf{z} - \tau(\mathbf{z})\mathbf{1}_M]_+, \tag{D.3}$$

where $\tau : \mathbb{R}^M \to \mathbb{R}$ is the threshold function $\tau(\mathbf{z}) = \left[\left(\sum_{\nu\leq\kappa(\mathbf{z})} z_{(\nu)}\right) - 1\right] / \kappa(\mathbf{z})$, satisfying $\sum_{\mu=1}^M [z_\mu - \tau(\mathbf{z})]_+ = 1$ for all $\mathbf{z}$. Notably, $\kappa(\mathbf{z}) = |S(\mathbf{z})|$ where $S(\mathbf{z}) = \{\mu \in [M] \mid \text{Sparsemax}\mu(\mathbf{z}) > 0\}$ is the support set of Sparsemax$(\mathbf{z})$.

D.2. Separation Loss

With the output vector of Φ is normalized, we have

$$\begin{aligned} \ell_\Phi(\mathbf{u}, \mathbf{v}) &= 2t \cdot \langle \Phi(\mathbf{u}), \Phi(\mathbf{v})\rangle - 2t \\ &= -t \cdot \|\Phi(\mathbf{u}) - \Phi(\mathbf{v})\|^2. \end{aligned} \tag{D.4}$$

Given the query $\mathbf{x} = \boldsymbol{\xi}_\mu + \mathbf{r}$, with $\mathbf{r} \in \mathbb{R}^d$, the uniformity loss satisfies the following:

$$\ell_\Phi(\mathbf{x}, \boldsymbol{\xi}_\mu) = \ell_\Phi(\boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu) + \ell_\Phi(\mathbf{r}, \boldsymbol{\xi}_\mu) + 2t.$$

D.3. Convergence Rate of Gradient Descent

Lemma D.1 ([JH: citation])**.** Suppose a function $f := \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable, and that its gradient is Lipschitz continuous with constant $G > 0$, i.e. $\|\nabla f(x) - \nabla f(y)\|_2 \leq G\|x - y\|$ for any $x, y \in \mathbb{R}^d$. Then if we run gradient descent for k iterations with a fixed step size $t \leq 1/G$, it returns a solution which satisfies

$$f(x^k) - f(x^\star) \leq \frac{\left\|x^0 - x^\star\right\|_2^2}{2tk},$$

where $f(x^\star)$ is the optimal value of f. Intuitively, this means that gradient descent is guaranteed to converge and that it converges with rate $\mathcal{O}(1/k)$.

Remark D.1. From Lemma D.1, to achieve a bound of:

$$f(x^k) - f(x^\star) \leq \epsilon,$$

we must run $k = \mathcal{O}(1/\epsilon)$ iterations of gradient descent, which gives us a sub-linear

E. Proofs of Main Text

E.1. Proof of Lemma 2.1

Proof. Here we introduce a helper lemma from (Sriperumbudur and Lanckriet, 2009, Lemma 5).

Lemma E.1 (Lemma 5 of (Sriperumbudur and Lanckriet, 2009))**.** Following Theorem 2.1, $\mathbf{x}$ is called the fixed point of iteration $\mathcal{T}$ w.r.t. E if $\mathbf{x} = \mathcal{T}(\mathbf{x})$ and is considered as a generalized fixed point of $\mathcal{T}$ if $\mathbf{x} \in \mathcal{T}(\mathbf{x})$. If $\mathbf{x}^\star$ is a generalized fixed point of $\mathcal{T}$, then, $\mathbf{x}^\star$ is a stationary point of the energy minimization problem in Equation (2.2).

Based on Zangwill's global convergence theory (Zangwill, 1969), a set of limit points of $\{x_t\}_{t=0}^{\infty}$ are all generalized fixed points if the retrieval dynamics and energy function satisfies the following conditions:

1. For any sequence $\{x_t\}_{t=0}^{\infty}$ with $x_0 \in S_\mu$ as starting point, all points in $\{x_t\}_{t=0}^{\infty}$ are in the compact set S_μ.
2. $E(\mathbf{x})$ is monotonically decreased by $\mathcal{T}(\mathbf{x})$, where $E(\mathbf{x}_{t+1}) \leq E(\mathbf{x}_t), \forall \mathbf{x}_{t+1} = \mathcal{T}(\mathbf{x}_t)$.
3. For all t, if $E(\mathbf{x}_{t+1}) \leq E(\mathbf{x}_t)$, $\mathcal{T}$ is closed at $\mathbf{x}_t$.

From Definition 2.1, since radius R is bounded and closed, S_μ is a compact set. Thus satisfies the first condition. CCCP (Yuille and Rangarajan, 2001) studied the monotonic decreasing property. With our definition of $E_{\text{convex}}, E_{\text{concave}}$, we have $E_U(\mathbf{x}, \mathbf{y}) = E_{\text{convex}}(\mathbf{x}) + E_{\text{concave}}(\mathbf{y}) + \langle (\mathbf{x} - \mathbf{y}), \nabla_{\mathbf{x}} E_{\text{concave}}(\mathbf{y}) \rangle$ is continuous in $\mathbf{x}, \mathbf{y}$. As a result, by (Hu et al., 2023, Lemma E.1), condition (iii) holds due to the non-empty assumption on the point-to-set map $\mathcal{T}$. Thus, by Zangwill's global convergence theory, all limit points are also the stationary points of the energy minimization problem in (2.2). By the results in Lemma E.1, these fixed points are also the stationary points of the minimization problem. Thus, (2.2) is guaranteed to converge to local minimum. □

E.2. Proof of Theorem 2.1

Proof. Since the lse function is non-decreasing and convex, and $\mathcal{K}$ is convex, the composited function $\operatorname{lse}(\mathcal{K}(\boldsymbol{\Xi}, \mathbf{x}))$ is convex. Thus, the energy function is the sum of a convex function: $\langle \mathbf{W}\mathbf{x}, \mathbf{W}\mathbf{x} \rangle / 2$ and a concave function: $-\operatorname{lse}(\mathcal{K}(\boldsymbol{\Xi}, \mathbf{x}))$.

Therefore, we have $E(\mathbf{x}) = E_{\text{convex}}(\mathbf{x}) + E_{\text{concave}}(\mathbf{x})$. With the Concave-Convex Procedure (CCCP) (Yuille and Rangarajan, 2001), the energy function $E(\mathbf{x})$ is guaranteed to monotonically decrease the energy E as a function of time with the following update rule:

$$\underbrace{\nabla_{\mathbf{x}} E_{\text{convex}}(\mathbf{x}^{t+1})}_{=\mathbf{A}\mathbf{x}^{t+1}} = \underbrace{-\nabla_{\mathbf{x}} E_{\text{concave}}(\mathbf{x}^{\top})}_{=\nabla_x \int \mathcal{T}(\mathbf{x}) \mathrm{d}\mathbf{x}},$$

such that

$$\mathbf{A}\mathbf{x}^{t+1} = \underbrace{\mathbf{A}}_{d \times d} \cdot \underbrace{\boldsymbol{\Xi} \cdot \operatorname{Sep}(\mathcal{K}(\boldsymbol{\Xi}, \mathbf{x}))}_{d \times 1} + \mathbf{c}_1,$$

where $\mathbf{c}_1 \in \mathbb{R}^d$ is a constant vector.

Since the matrix $\mathbf{A}$ is non-singular by Assumption 2.1, the solution of the update rule $\mathbf{A}\mathbf{x}^{t+1} = \mathbf{A}\boldsymbol{\Xi} \cdot \operatorname{Sep}(\mathcal{K}(\boldsymbol{\Xi}, \mathbf{x})) \in \mathbb{R}^d$ is the solution of $\mathbf{x}^{t+1} = \boldsymbol{\Xi} \cdot \operatorname{Sep}(\mathcal{K}(\boldsymbol{\Xi}, \mathbf{x})) \in \mathbb{R}^d$ minimizes the energy function $E(\mathbf{x})$. This completes the proof. □

E.3. Proofs of Theorem 2.2 and Corollary 2.2.1

Proof. Here we use the same proof strategy in (Martins et al., 2023, Proposition 2).

The Fenchel-Young loss (Scholkopf and Smola, 2018) indicates that if a vector $\boldsymbol{\theta} \in \mathbb{R}^d$ for any $d > 1$ satisfies

$$\alpha\text{-EntMax}(\boldsymbol{\theta}) = \boldsymbol{e}_\mu,$$

then $\boldsymbol{\theta}$ must satisfy

$$\theta_\mu - \max_{\nu \neq \mu} \theta_\nu \geq \frac{1}{\alpha - 1}. \tag{E.1}$$

If we have exact memory retrieval of pattern $\boldsymbol{\xi}_\mu$, the following equation holds:

$$\mathbf{e}_\mu = \alpha\text{-EntMax}\left(\beta \mathcal{K}(\boldsymbol{\Xi}, \boldsymbol{\xi}_\mu)\right). \tag{E.2}$$

This is also equivalent to $\boldsymbol{\xi}_\mu$ itself being the fixed point.

By combining (E.1) and (E.2), we have

$$\mathcal{K}(\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu) - \max_{\nu \neq \mu} \mathcal{K}(\boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu) \geq \frac{1}{\beta\left(\alpha - 1\right)}.$$

With $\mathcal{K}(\boldsymbol{\xi}_\mu, \mathbf{x}) = \frac{1}{2t} \cdot \ell_\Phi\left(\boldsymbol{\xi}_\mu, \mathbf{x}\right)$, we have

$$\ell_\Phi\left(\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\mu\right) - \max_{\nu \neq \mu} \ell_\Phi\left(\boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu\right) \geq \frac{2t}{\beta\left(\alpha - 1\right)}. \qquad \text{(By Theorem 2.2)}$$

With the assumption of normalized patterns, we are able to reduce the above condition to

$$\max_{\nu \neq \mu} \ell_\Phi\left(\boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu\right) \leq \frac{2t}{\beta\left(\alpha - 1\right)}. \tag{E.3}$$

Assuming Φ is L-lipschitz[4], we derive another upper bound from (E.3):

$$\begin{aligned} \max_{\nu \neq \mu} \ell_\Phi\left(\boldsymbol{\xi}_\nu, \boldsymbol{\xi}_\mu\right) &= \max_{\nu \neq \mu} -t \cdot \left\|\Phi(\boldsymbol{\xi}_\nu) - \Phi(\boldsymbol{\xi}_\mu)\right\|^2 \\ &\leq \max_{\nu \neq \mu} -t \cdot L^2 \|\boldsymbol{\xi}_\nu - \boldsymbol{\xi}_\mu\|^2. \end{aligned} \tag{E.4}$$

By combining (E.4) and (E.3), we obtain

$$\min_{\nu \neq \mu} \|\boldsymbol{\xi}_\nu - \boldsymbol{\xi}_\mu\|^2 \geq \frac{2}{\beta\left(\alpha - 1\right) \cdot L^2}. \qquad \text{(By Corollary 2.2.1)}$$

This completes the proof. □

[4] Φ is always L-lipschitz for some $L \in \mathbb{N}$ in our construction since a linear affine function always has L-Lipschitzness.

E.4. Proof of Theorem 3.1

Theorem 3.1 (Kernelized Representation Theorem)**.** Let $\overline{\Phi}$ be a feature map such that $\overline{\Phi} \coloneqq \mathbb{R}^d \to \mathbb{R}^{D_{\overline{\Phi}}}$, and $\overline{\mathcal{K}}$ be a $\overline{\Phi}$-induced kernel. Assuming $\overline{\mathcal{K}}$ satisfies: $\bar{\ell}_{\overline{\Phi}}(\mathbf{u}, \mathbf{v}) = -2t$ for any given $\mathbf{u}, \mathbf{v} \in \mathbf{\Xi}$. With $\beta > 0$, input $\mathbf{X} \in \mathbb{R}^{d \times M}$, $M \leq d$, and an arbitrary positive column stochastic matrix $\mathbf{P} \in \mathbb{R}^{M \times M}$, there always exist matrices $\mathbf{W}_Q, \mathbf{W}_K$ such that

$$\text{Softmax}\left(\beta \ \left(\mathbf{W}_K \overline{\Phi}(\mathbf{X})\right)^\mathsf{T} \mathbf{W}_Q \overline{\Phi}(\mathbf{X})\right) = \mathbf{P}.$$

Proof. Let $\mathbf{X}' = \overline{\Phi}(\mathbf{X}) \in \mathbb{R}^{d \times M}$.

By construction, any two columns in $\mathbf{X}'$ satisfies:

$$\left\langle \overline{\Phi}(\mathbf{u}), \overline{\Phi}(\mathbf{v}) \right\rangle = 0, \qquad \left(\text{By } \bar{\ell}_{\overline{\Phi}}(\mathbf{u}, \mathbf{v}) = -2t\right)$$

for all $\mathbf{u}, \mathbf{v} \in \mathbf{X}'$, $\mathbf{u} \neq \mathbf{v}$. Thus, any two columns in $\mathbf{X}'$ are orthogonal to each other, and hence $\mathbf{X}'$ has left inverse $\mathbf{X}^\dagger \in \mathbb{R}^{M \times d}$.

Let $\mathbf{W}_K \coloneqq \widetilde{\mathbf{W}}_K \mathbf{X}^\dagger$, and $\mathbf{W}_Q \coloneqq \widetilde{\mathbf{W}}_Q \mathbf{X}^\dagger$ for some $\widetilde{\mathbf{W}}_K, \widetilde{\mathbf{W}}_Q \in \mathbb{R}^{d \times M}$.

This gives us

$$\left(\mathbf{W}_K \mathbf{X}'\right)^\mathsf{T} \mathbf{W}_Q \mathbf{X}' = \left(\widetilde{\mathbf{W}}_K \mathbf{X}^\dagger \mathbf{X}'\right)^\mathsf{T} \widetilde{\mathbf{W}}_Q \mathbf{X}^\dagger \mathbf{X}' = \widetilde{\mathbf{W}}_K^\mathsf{T} \widetilde{\mathbf{W}}_Q = \widetilde{\mathbf{W}}_{KQ}. \qquad \left(\widetilde{\mathbf{W}}_{KQ} \in \mathbb{R}^{M \times M}\right)$$

With softmax, we have

$$\begin{aligned} \text{Softmax}\left(\beta \ \left(\mathbf{W}_K \mathbf{X}'\right)^\mathsf{T} \mathbf{W}_Q \mathbf{X}'\right) &= \text{Softmax}\left(\beta \ \widetilde{\mathbf{W}}_{KQ}\right) \\ &= \exp\left\{\beta \widetilde{\mathbf{W}}_{KQ}\right\} \cdot \mathbf{D}^{-1}_{\widetilde{\mathbf{W}}_{KQ}}, \end{aligned} \tag{E.5}$$

where $\mathbf{D}^{-1}_{\widetilde{\mathbf{W}}_{KQ}} \in \mathbb{R}^{M \times M}$ is a diagonal matrix which

$$\left(\mathbf{D}_{\widetilde{\mathbf{W}}_{KQ}}\right)_{ii} = \sum_{j=1}^{M} \exp\left\{\beta \left(\widetilde{\mathbf{W}}_{KQ}\right)_{ji}\right\} = \left(\mathbf{1}^\top\right) \sum_{j=1}^{M} \exp\left\{\beta \left(\widetilde{\mathbf{W}}_{KQ}\right)\right\}. \tag{E.6}$$

Now with $\mathbf{P}$, we are able to construct $\widetilde{\mathbf{W}}_{KQ}$ by picking an arbitrary positive diagonal matrix such that

$$\widetilde{\mathbf{W}}_{KQ} = \beta^{-1} \cdot \log\left(\mathbf{P} \cdot \mathbf{D}_0\right). \tag{E.7}$$

By combining (E.6) and (E.7), we have

$$\mathbf{D}_{\widetilde{\mathbf{W}}_{KQ}} = \text{Diag}\left(\mathbf{1}^\top \exp\left\{\beta\beta^{-1} \cdot \log\left(\mathbf{P} \cdot \mathbf{D}_0\right)\right\}\right) = \text{Diag}\left(\mathbf{1}^\top \mathbf{P} \cdot \mathbf{D}_0\right) = \mathbf{D}_0.$$

With $\mathbf{D}_{\widetilde{\mathbf{W}}_{KQ}} = \mathbf{D}_0$, and using (E.5), we obtain

$$\text{Softmax}\left(\beta \ \widetilde{\mathbf{W}}_{KQ}\right) = \exp\left\{\beta \widetilde{\mathbf{W}}_{KQ}\right\} \cdot \mathbf{D}^{-1}_{\widetilde{\mathbf{W}}_{KQ}} = \exp\left\{\log\left(\mathbf{P} \cdot \mathbf{D}_{\widetilde{\mathbf{W}}_{KQ}}\right)\right\} \cdot \mathbf{D}^{-1}_{\widetilde{\mathbf{W}}_{KQ}} = \mathbf{P}.$$

As a result, to construct $\mathbf{W}_K$ and $\mathbf{W}_Q$ such that they satisfy Theorem 3.1, any two matrices must satisfy

$$\mathbf{W}_K^\mathsf{T} \mathbf{W}_Q = \beta^{-1} \cdot \log\left(\mathbf{P} \cdot \mathbf{D}_{\widetilde{\mathbf{W}}_{KQ}}\right)$$

This completes the proof. □

F. Implementation Details

F.1. Data

- **MNIST.** It is a hand written digits image recognition dataset (LeCun et al., 1998) consists of 60000 training samples and 10000 test samples. Each image has the size of 28×28. The label contains digits from 0 to 9.
- **CIFAR10.** It is an image recognition dataset (Krizhevsky et al., 2009) consists of 50000 training samples and 10000 test samples. Each image has the size of 32×32. The dataset contains 10 categories with 6000 samples for each.
- **CIFAR100.** It is an image recognition dataset (Krizhevsky et al., 2009) consists of 50000 training samples and 10000 test samples. Rach image has the size of 32×32. The dataset contains 100 categories with 600 samples for each.
- **TinyImageNet.** It is an image recognition dataset (Le and Yang, 2015) contains 100000 images of 200 classes. Each image is downsized to 64×64 colored images. Each class has 500 training images, 50 validation and test images.
- **ETT (Electricity Transformer Temperature).** ETT (Zhou et al., 2021) records 2 years of data from two counties in China. We use two sub-datasets: ETTh1 (hourly) and ETTm1 (every 15 minutes). Each entry includes the "oil temperature" target and six power load features.
- **WTH (Weather).** WTH records climatological data from approximately 1,600 U.S. sites between 2010 and 2013, measured hourly. Entries include the "wet bulb" target and 11 climate features.

F.2. Memory Capacity

For memory capacity experiment, we follow the settings in (Hu et al., 2023; Wu et al., 2024).

We randomly mask 50% of the pixels in the image, using the masked image as a query for a single-step update with various Hopfield models. In the case of `U-Hop`, we trained the kernel with different numbers of epochs on the memory set and then used it for memory retrieval. We reported the sum-of-square pixel difference between the retrieved image and the ground truth. In each run, we repeated this process for every image in the memory set, conducting the experiment 20 times for each baseline. The range of kernel learning epochs, memory set size can be found in Table 4. For Figure 26, we use $N = 100$ for MNIST and $N = 500$ for CIFAR10.

Table 4. Hyperparameter used in the Memory Retrieval Task.

parameter	**values**
Kernel optimizer	SGD
Kernel learning rate	1
Kernel epoch	$\{1, 10, 20, 50, 100, 200, 500, 1000\}$
Memory set size	$\{10, 20, 30, 50, 100, 200, 500\}$
Noise level	$\{0.0, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1.0, 1.2, 1.4, 2.0\}$

F.3. Noise Robustness

For noise robustness experiment, we follow the settings in (Hu et al., 2023; Wu et al., 2024).

For the noise robustness experiment, we randomly sampled a Gaussian noise vector for each image, varying the norm of the sampled noise to adjust the noise level. We then added the noise to the query image and performed a single-step update with different Hopfield models. For `U-Hop`, we trained the kernel with N iterations on the memory set and then used it for memory retrieval. We set $N = 100$ for MNIST and $N = 200$ for CIFAR10. We reported the sum-of-square pixel difference between the retrieved image and the ground truth. In each run, we repeated this process for every image in the memory set, conducting the experiment 20 times for each baseline.

F.4. Classification

CIFAR10 and CIFAR100. For these two datasets, we consider four different Hopfield layers as encoder:

- `Hopfield` (Ramsauer et al., 2020)
- `SparseHopfield` (Hu et al., 2023)

- `Hopfield` +`U-Hop`
- `SparseHopfield` + `U-Hop`.

We use a fully connected layer right after the encoder for classification. For each image, we follow the same process as introduced in (Dosovitskiy et al., 2020). We split an image into patches and add an additional `CLS` patch for classification. We send patches into the `Hopfield` layer with the `CLS` patch as query and other patches as memory. We then send the output to a fully connected layer for prediction. We use the CrossEntropy loss and Adam optimizer for training. Hyperparameters are in Table 5.

Table 5. Hyperparameter used in the classifciation task on CIFAR10 and CIFAR100.

parameter	**values**
learning rate	$1e-3$
embedding dimension	512
Epoch	25
Batch size	128
Model optimizer	Adam
Kernel optimizer	SGD
Kernel learning rate	0.1
Patch size	32

Table 6. Hyperparameter used in the classifciation task on Tiny ImageNet.

parameter	**values**
learning rate	$1e-4$
embedding dimension	512
Epoch	25
Batch size	128
Model optimizer	Adam
Kernel optimizer	SGD
Kernel learning rate	0.1
Patch size	64

Tiny ImageNet. For this dataset, we use a 3 layered Vision Transformer as backbone (Dosovitskiy et al., 2020), and use `Hopfield` variations to replace attention mechanism in . Other processes are the same as introduced in the above paragraph. For kernel learning, we learn all kernels in different layers by passing a full forward pass. We then send the output to a fully connected layer for prediction. Hyperparameters are in Table 6.

F.5. Hopfield-Based Time Series Prediction with STanHop-Net (Wu et al., 2024) (Table 8)

For this task, we use STanHop-Net (Wu et al., 2024) as backbone, and equip it with `U-Hop`. In addition, we use Algorithm 2 to minimize both separation loss and the MAE loss. For prediction horizon of $96, 192$, we use one layered STanHop-Net, for $336, 720$, we use a two layered STanHop-Net. We use the same settings for with and without `U-Hop` [5].

[5]We thank the authors of (Reneau et al., 2023) for their helpful comments on this part.

Table 7. Hyper-parameter used in the time series prediction.

parameter	**values**
learning rate	$1e-4$
embedding dimension	256
Epoch	50
Patience	25
Batch size	16
Model optimizer	Adam
Kernel optimizer	SGD
Kernel learning rate	0.1
Patch Sequence Length	12
Window Size	2

G. Additional Numerical Experiments

G.1. Relationship between Separation Loss and Retrieval Error

This section is a visualization of relationship between separation loss and retrieval error on MNIST and CIFAR10. The result shows that the retrieval error is highly correlated with respect to the value separation loss.

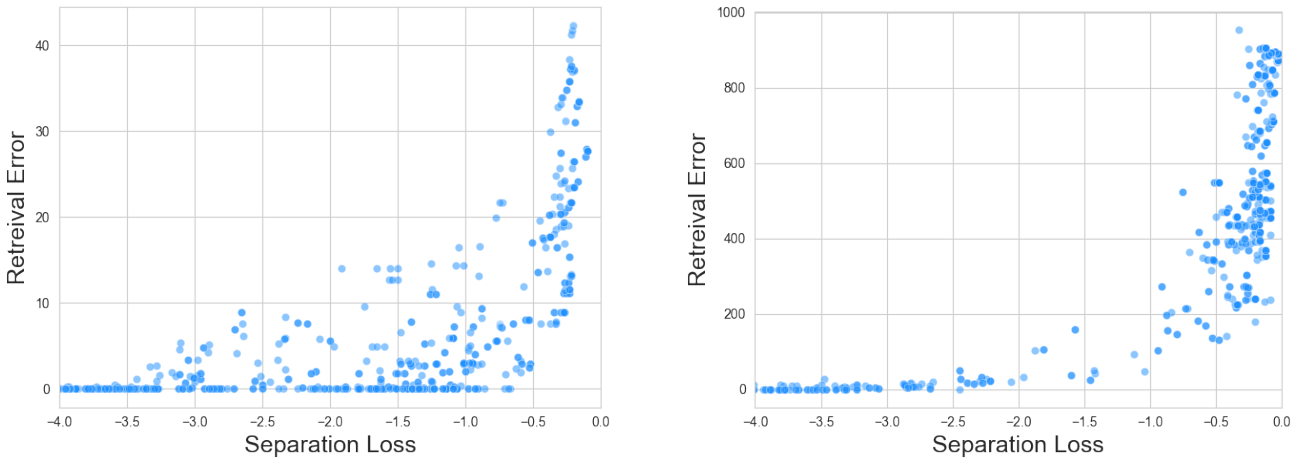

Figure 7. **Memory Retrieval Error v.s. Separation Loss: Left: MNIST, Right: CIFAR10**. We conduct memory retrieval experiment on MNIST and CIFAR10 dataset. We use randomly sampled kernel learning rate in $N(0, 1)$, and uniformly random sampled $N \in [1, 200]$ to obtain diverse separation loss.

G.2. Max. Loss v.s. Avg. Loss

In the main paper, we discuss the differences between minimizing the maximum separation loss and the average separation loss.

Ideally, minimizing the maximum separation loss directly contributes to R_Φ. However, as stated in the main text, such a loss is a max-min problem, which is challenging to optimize. Moreover, it entails an additional quadratic time complexity due to the max operation. On the other hand, the average loss is more time-efficient. It is also convex, thereby guaranteeing convergence to the global optimum at a rate of $\mathcal{O}(1/N)$ under gradient descent, where N is the number of iterations. However, the average loss does not guarantee maximizing R_Φ, nor does it ensure an optimal $\Delta_{\Phi,\mu}$ for any $\mu \in [M]$. Therefore, its theoretical impact on memory capacity and retrieval error bound is difficult to quantify.

As a result, we conduct an analysis comparing the performance of each loss function. We vary the memory size and the kernel learning iteration N and perform memory retrieval on MNIST and CIFAR10 datasets.

The results demonstrate that minimizing the average loss yields a lower retrieval error, and this advantage grows with the size of the memory set. Additionally, the retrieval error decreases more rapidly with respect to N under average loss than under maximum loss. This outcome is anticipated, as minimizing the maximum loss is a non-convex problem and does not guarantee reaching global optima. This empirical finding indicates that in practice, minimizing the average loss leads to better efficiency and retrieval outcomes.

Max. & Avg. Loss Comparison on MNIST. We observe that with the average loss, `U-Hop` achieves almost perfect retrieval outcomes with $N = 200$. However, `U-Hop` using the maximum loss struggles to reach its global minimum during optimization, thus hindering its ability to achieve optimal memory capacity. This is observed in Appendix G.2.

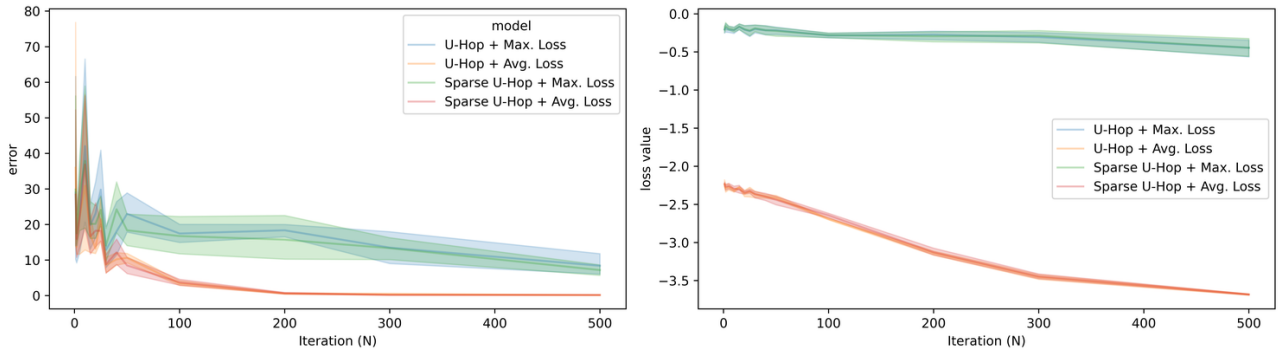

Figure 8. **Loss Value v.s. N and Retrieval Error v.s. N.**

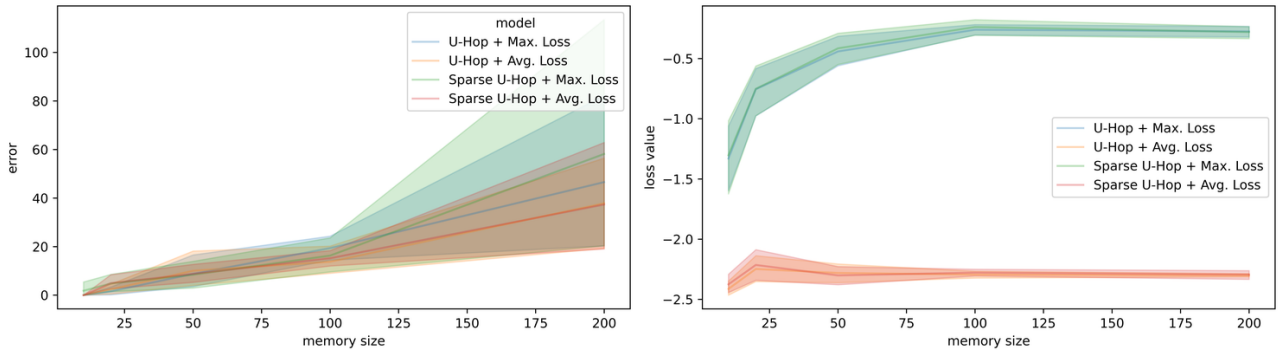

Figure 9. **Max. vs Avg. Loss on MNIST with $N = 10$.**

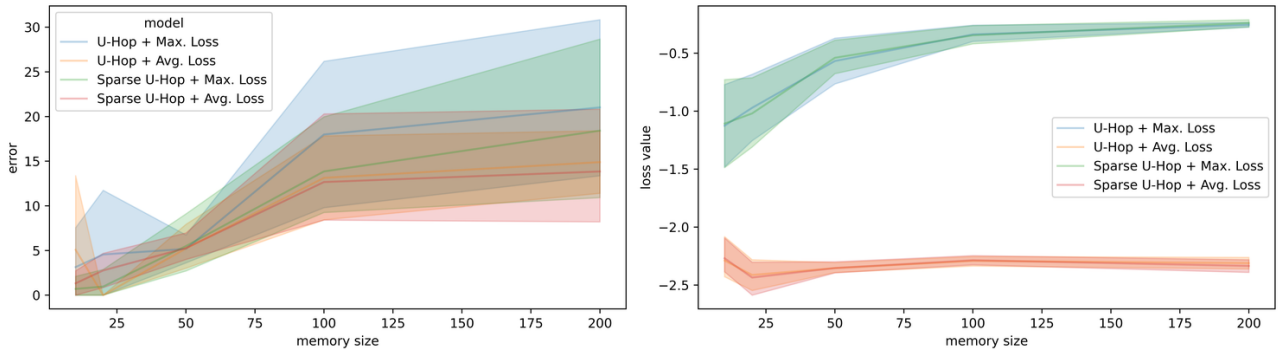

Figure 10. **Max. vs Avg. Loss on MNIST with $N = 20$.**

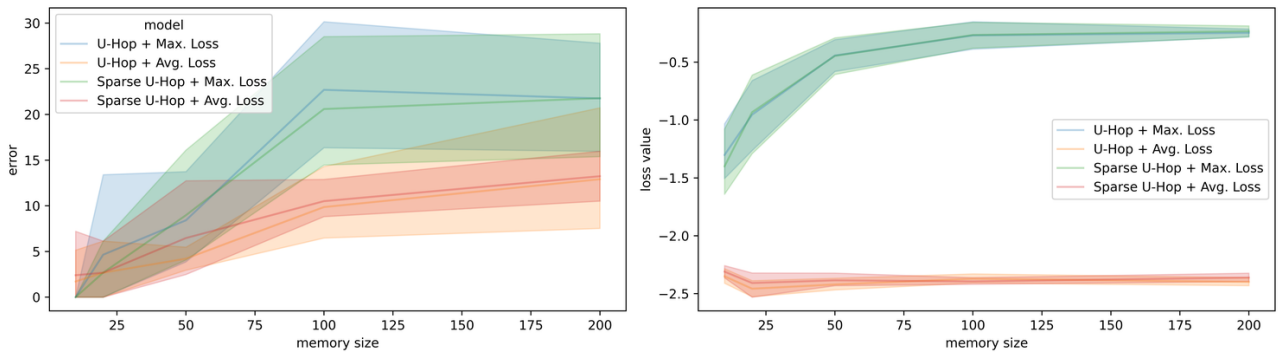

Figure 11. **Max. vs Avg. Loss on MNIST with $N = 30$.**

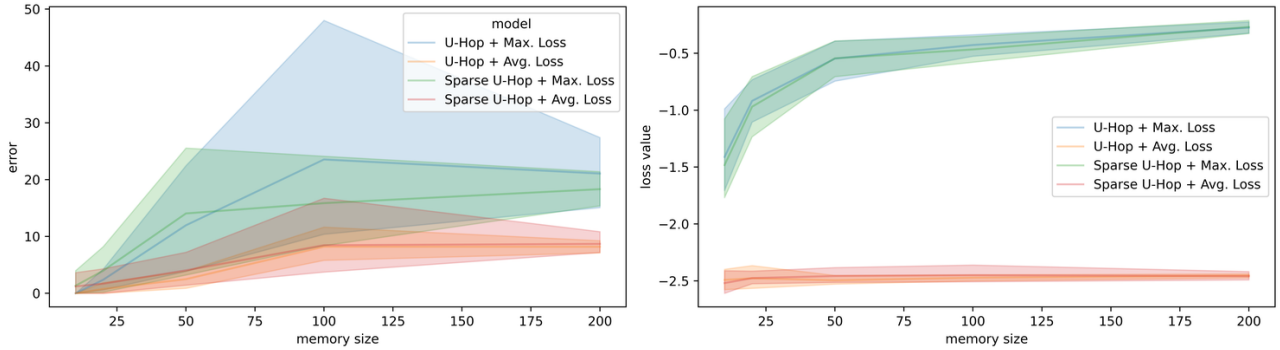

Figure 12. **Max. vs Avg. Loss on MNIST with $N = 50$.**

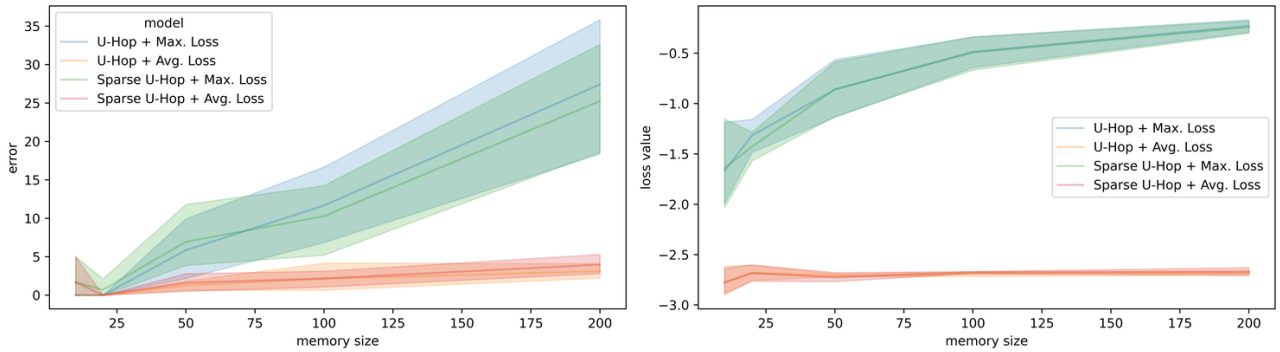

Figure 13. **Max. vs Avg. Loss on MNIST with $N = 100$.**

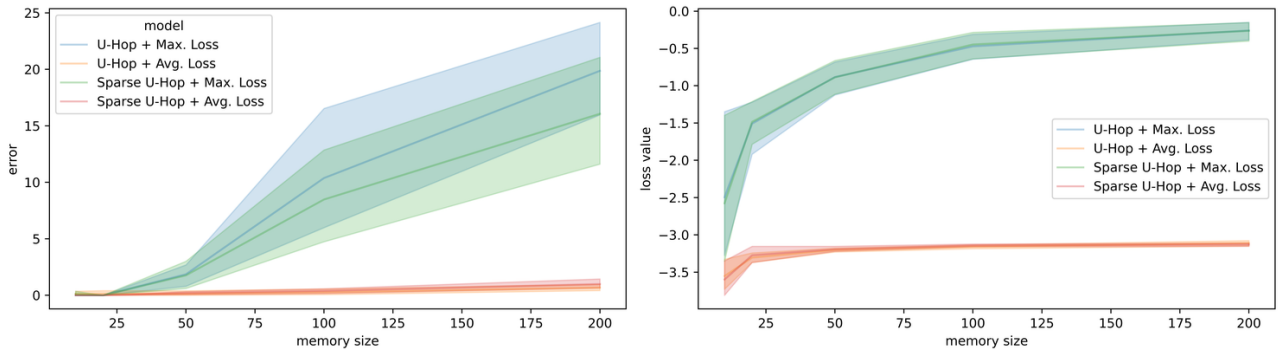

Figure 14. **Max. vs Avg. Loss on MNIST with $N = 200$.**

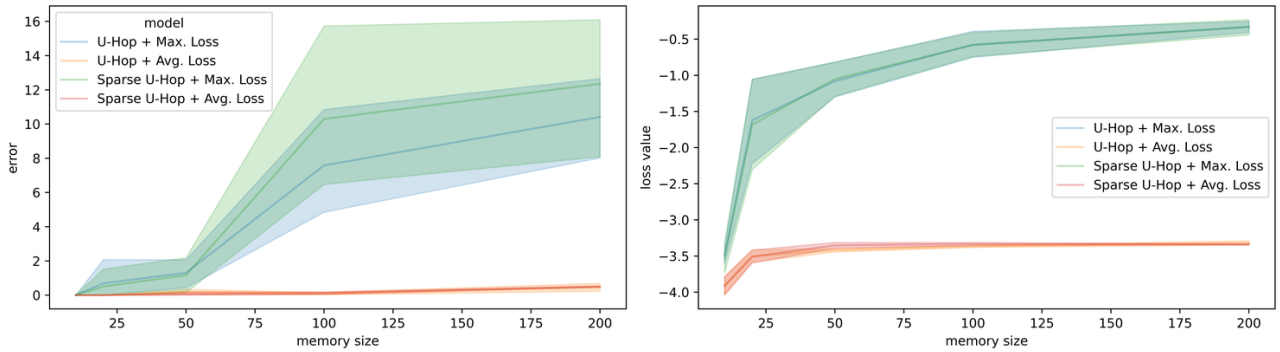

Figure 15. **Max. vs Avg. Loss on MNIST with $N = 10$.**

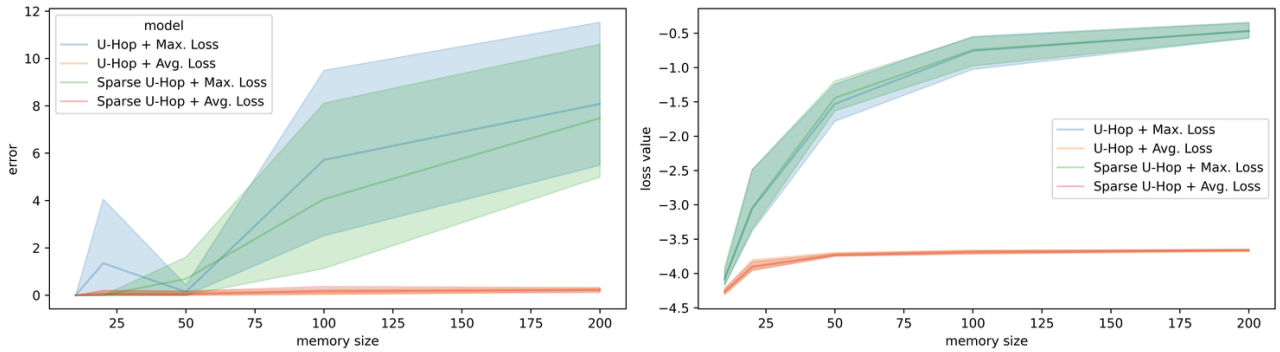

Figure 16. **Max. vs Avg. Loss on MNIST with $N = 500$.**

Max. Avg. Loss Comparison on CIFAR10. With CIFAR10 being more difficult comparing to MNIST, `U-Hop` under average loss still outperforms `U-Hop` under Max. loss.

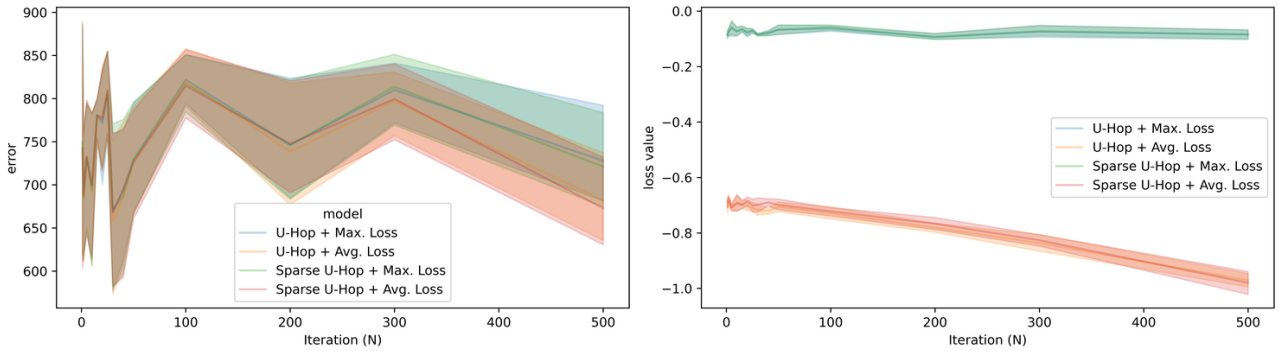

Figure 17. **Loss Value v.s. N and Retrieval Error v.s. N on CIFAR10.**

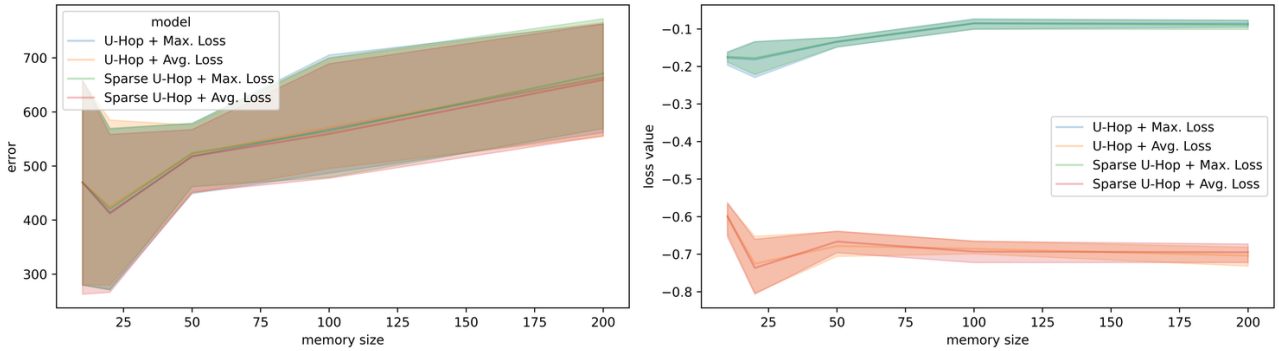

Figure 18. **Max. vs Avg. Loss on CIFAR10 with $N = 10$.**

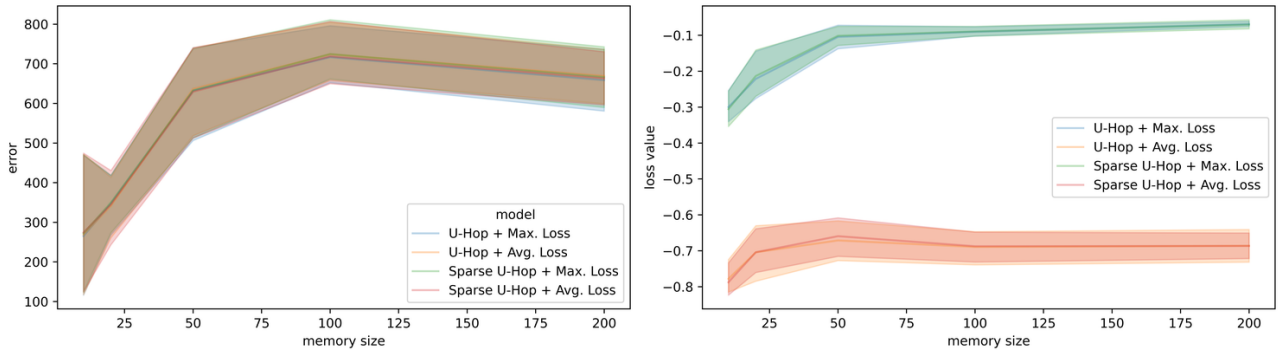

Figure 19. **Max. vs Avg. Loss on CIFAR10 with $N = 20$.**

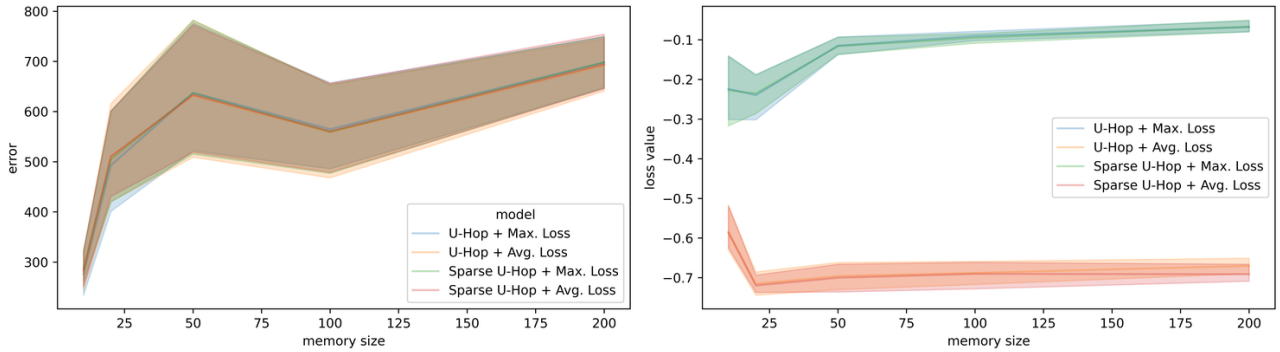

Figure 20. **Max. vs Avg. Loss on CIFAR10 with $N = 30$.**

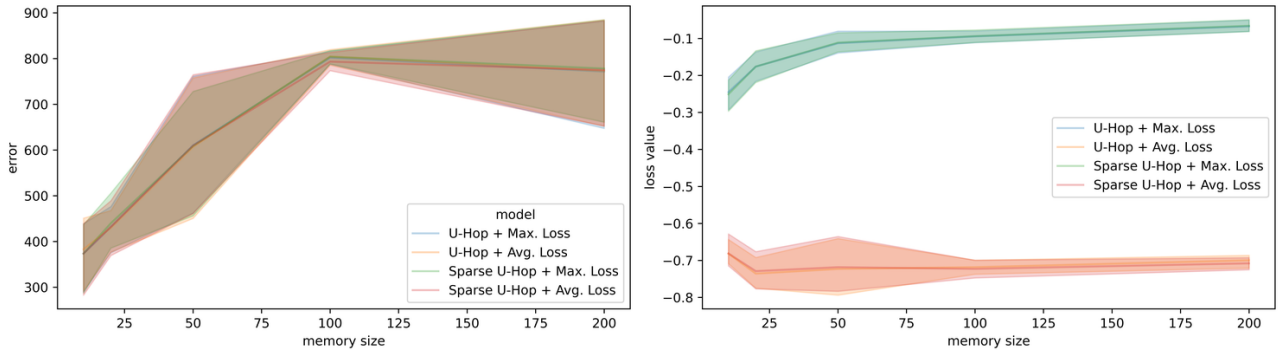

Figure 21. **Max. vs Avg. Loss on CIFAR10 with $N = 50$.**

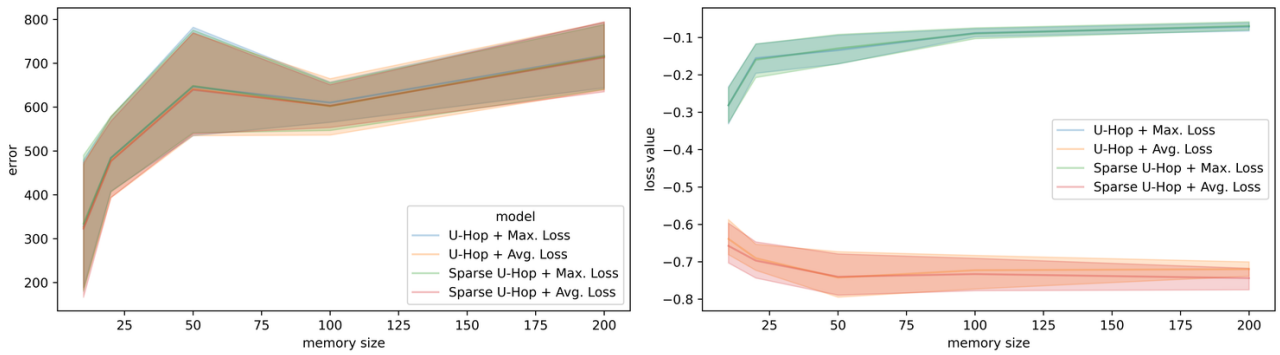

Figure 22. **Max. vs Avg. Loss on CIFAR10 with $N = 100$.**

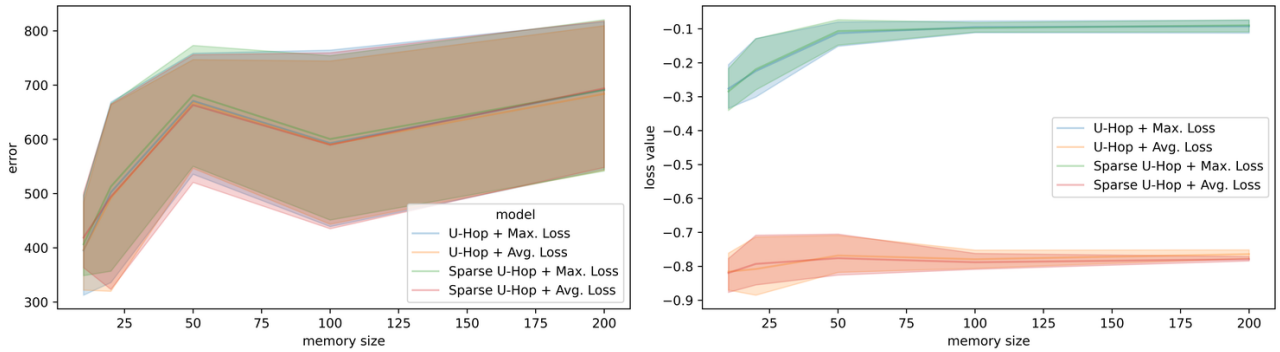

Figure 23. **Max. vs Avg. Loss on CIFAR10 with $N = 200$.**

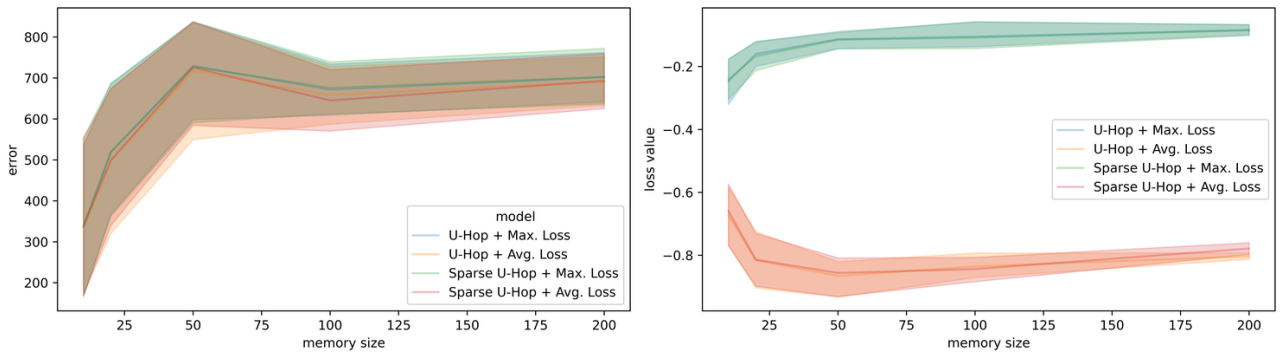

Figure 24. **Max. vs Avg. Loss on CIFAR10 with $N = 250$.**

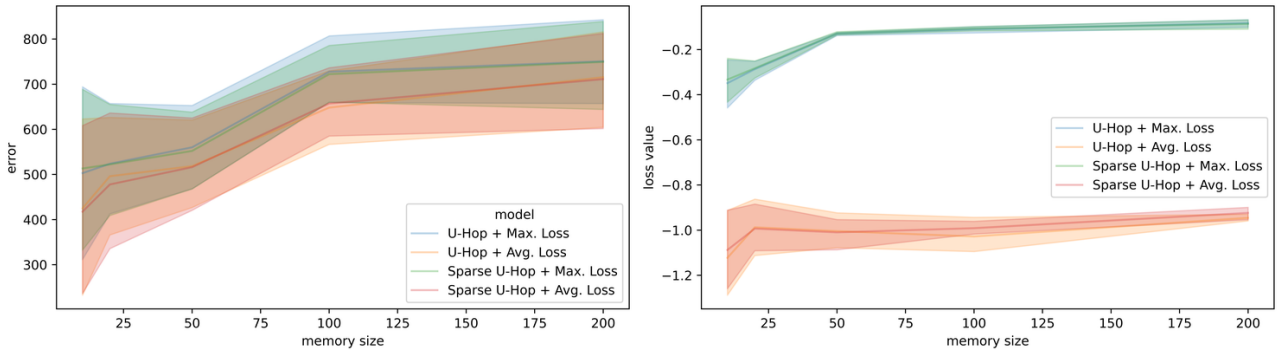

Figure 25. **Max. vs Avg. Loss on CIFAR10 with $N = 500$.**

G.3. Memory Retrieval

Here we again show the memory retrieval results with higher resolution. The result demonstrates with `U-Hop`, modern Hopfield models obtain significant improvement on both datasets.

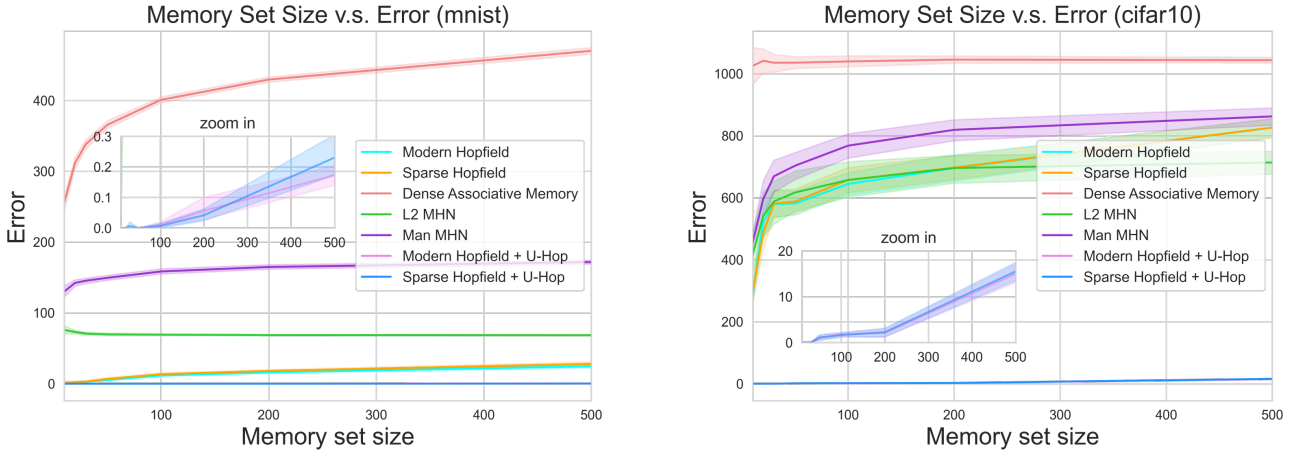

Figure 26. **Memory Retrieval Error v.s. Memory Set Size (M). Left: MNIST, Right: CIFAR10**. We conduct memory retrieval experiment on MNIST and CIFAR10 dataset. We vary the memory size to adjust the difficulty of retrieval process.

G.4. Model Expressiveness

Here we present the model expressiveness on training data. This is a empirical validation for Theorem 3.1. We observe that without `U-Hop`, baseline models suffer from sharp performance drop, which also lead to generalization degradation as show in Appendix G.5.1. In contrast, with `U-Hop`, models show better robustness against sample size increase.

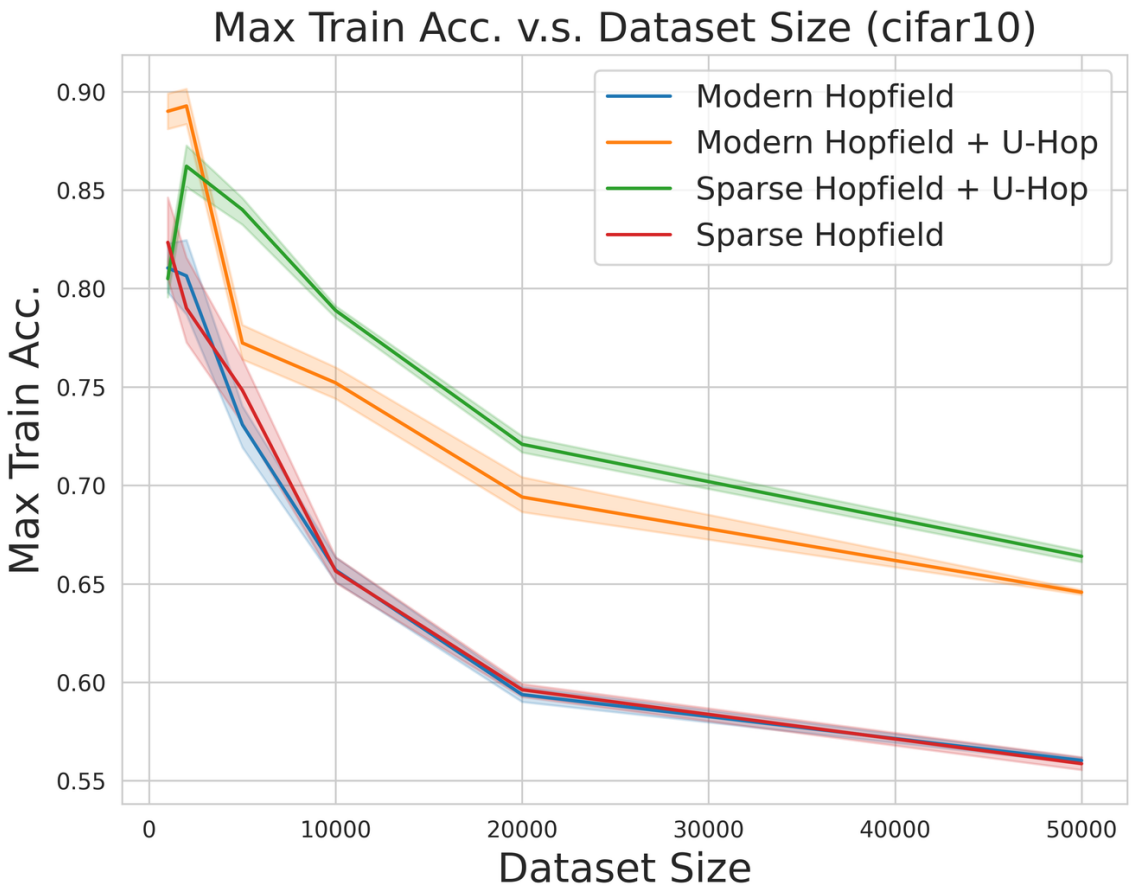

Figure 27. **Max Training Accuracy with respect to Sample Size Increase on CIFAR10**

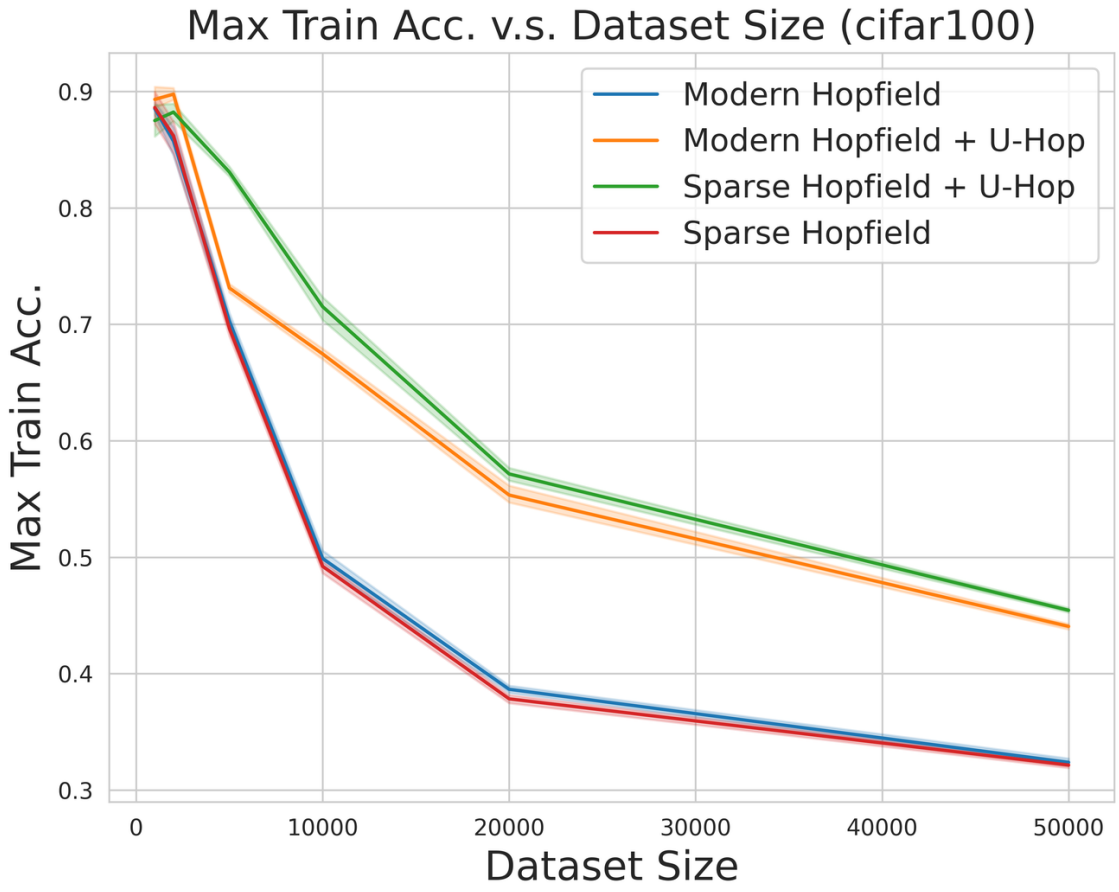

Figure 28. **Max Training Accuracy with respect to Sample Size Increase on CIFAR100**

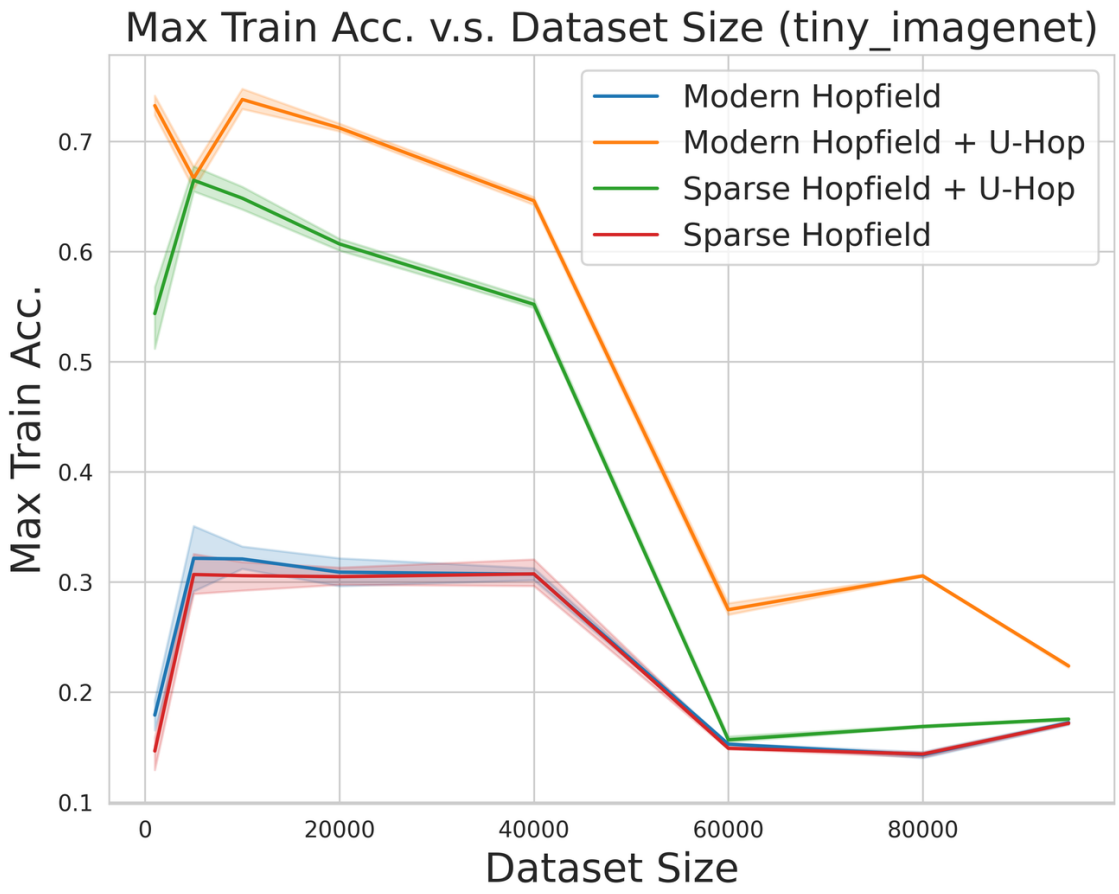

Figure 29. **Max Training Accuracy with respect to Sample Size Increase on Tiny ImageNet.**

G.5. Classification

Here we conduct empirical analysis on the correlation between dataset size and model convergence. In general, it is more difficult to memorize all samples for larger dataset. However, learning from more samples might lead to a better generalization performance, results in higher test accuracy.

G.5.1. CIFAR10

Results. The result demonstrates `U-Hop` significantly improves model's performance on 3 aspects:

- (i) Generalization (test accuracy)
- (ii) Convergence Speed
- (iii) Memorization (training accuracy)

The improvement also became more obvious when the dataset size increases. This is reasonable as the standard `Hopfield` layer is powerful enough to memorize small sample size with and without `U-Hop`.

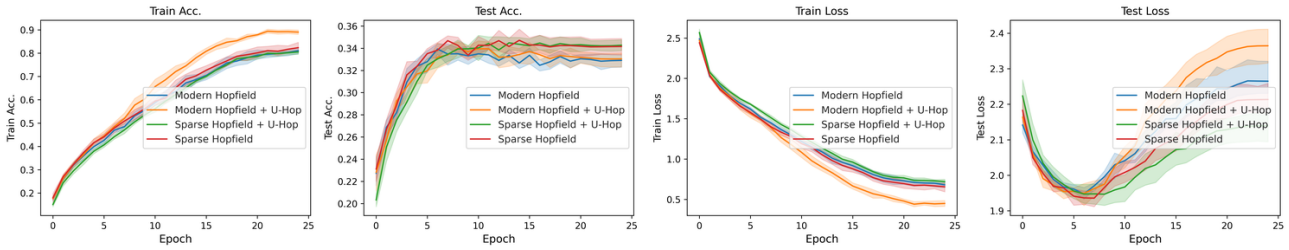

Figure 30. **CIFAR10 Convergence Comparison with Dataset Size=1000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

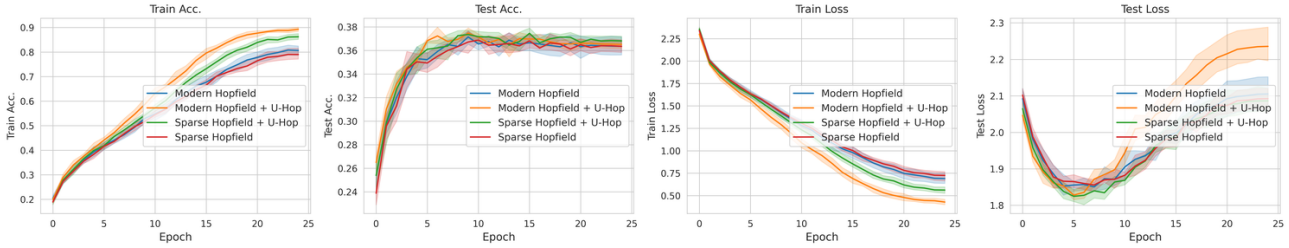

Figure 31. **CIFAR10 Convergence Comparison with Dataset Size=2000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

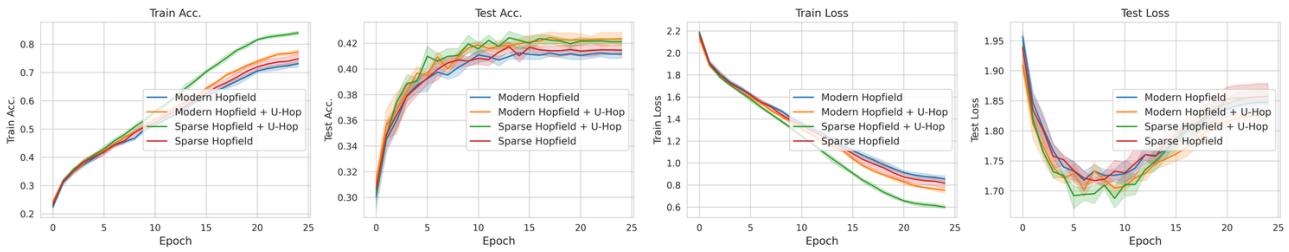

Figure 32. **CIFAR10 Convergence Comparison with Dataset Size=5000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

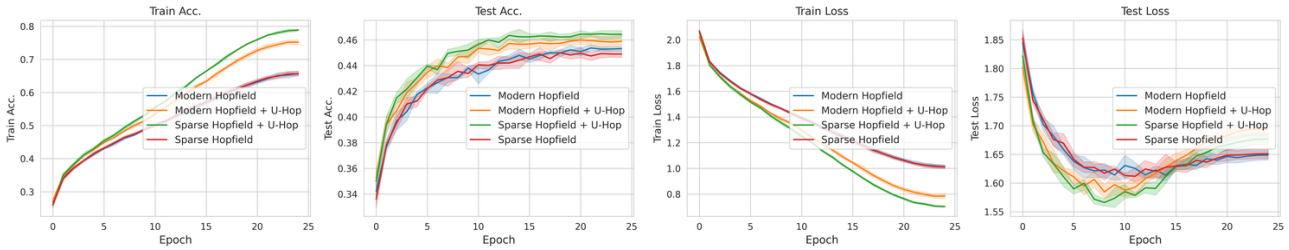

Figure 33. **CIFAR10 Convergence Comparison with Dataset Size=10000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

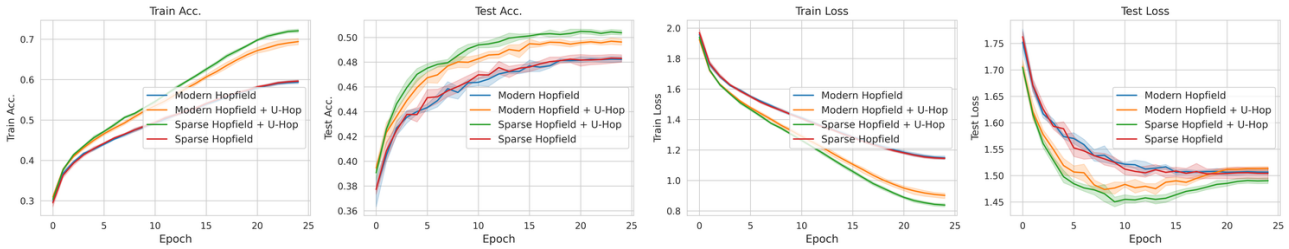

Figure 34. **CIFAR10 Convergence Comparison with Dataset Size=20000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

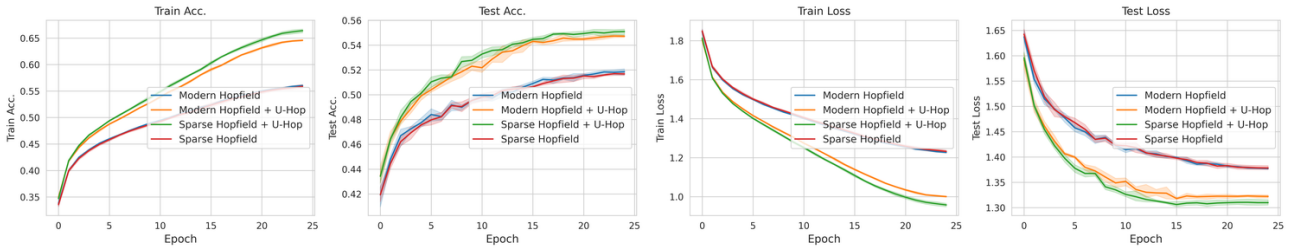

Figure 35. **CIFAR10 Convergence Comparison with Dataset Size=Full** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

G.5.2. CIFAR100

Results. The model behavior was similar comparing to what we observe from CIFAR10. The performance improvement under `U-Hop` became stronger with the increase of dataset size. With `U-Hop`, the model improves on both convergence speed, memorization capacity (training accuracy) and generalization power (test accuracy).

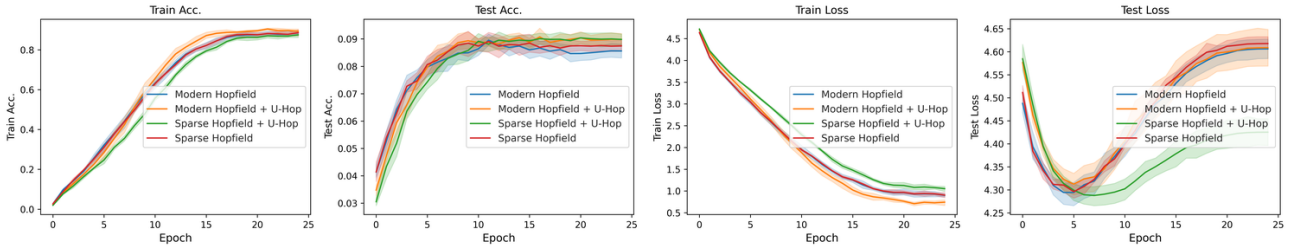

Figure 36. **CIFAR100 Convergence Comparison with Dataset Size=1000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

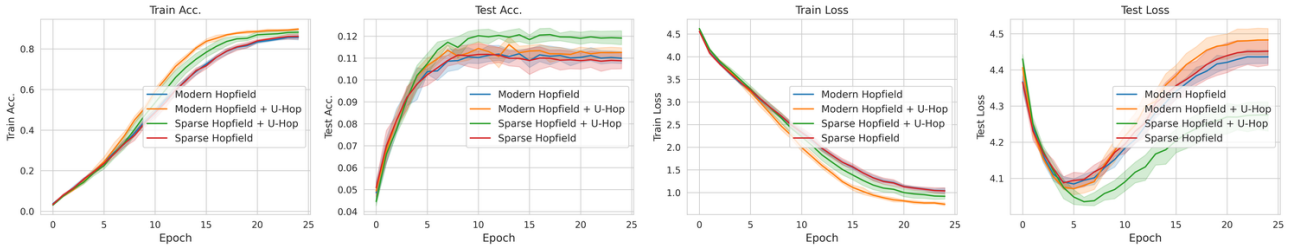

Figure 37. **CIFAR100 Convergence Comparison with Dataset Size=2000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

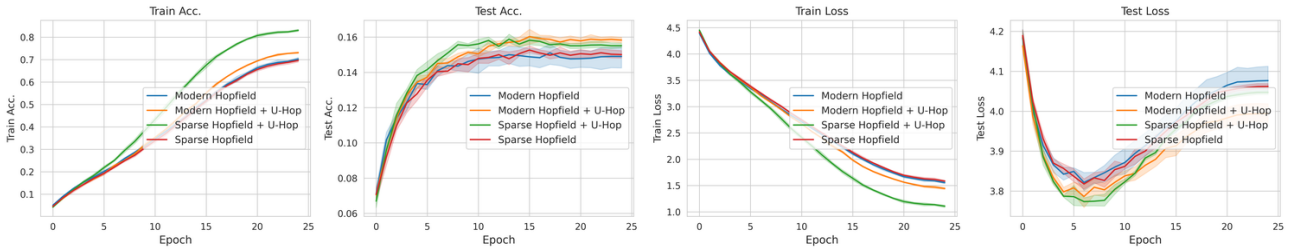

Figure 38. **CIFAR100 Convergence Comparison with Dataset Size=5000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

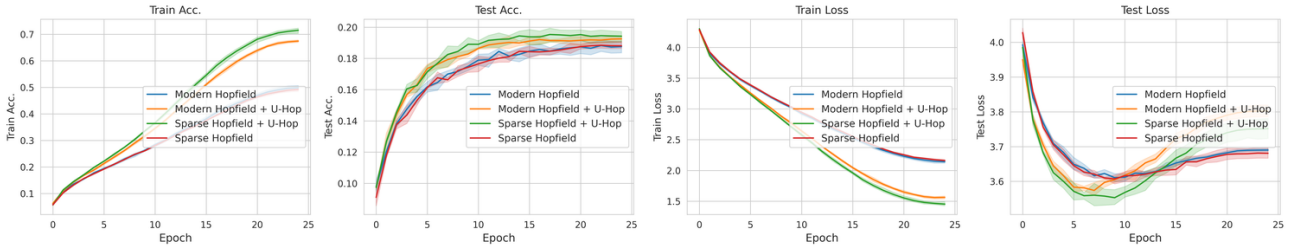

Figure 39. **CIFAR100 Convergence Comparison with Dataset Size=10000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

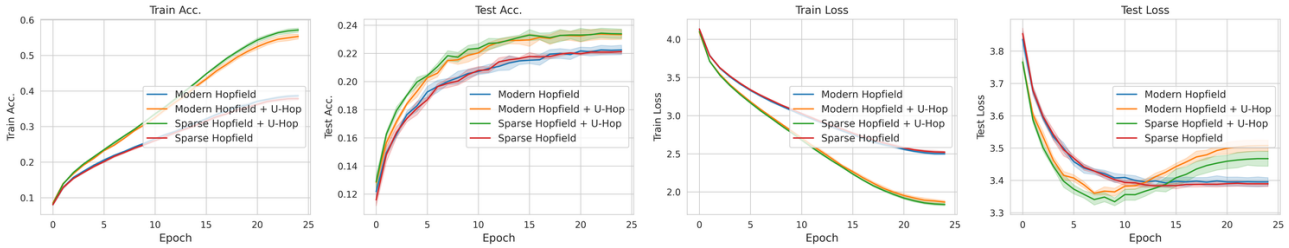

Figure 40. **CIFAR100 Convergence Comparison with Dataset Size=20000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

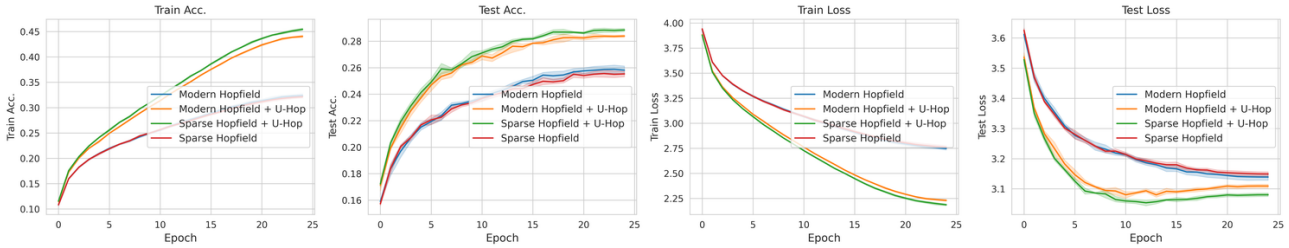

Figure 41. **CIFAR100 Convergence Comparison with Dataset Size=Full** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

G.5.3. TINY IMAGENET

Models under U-Hop continue to show strong performance against baselines on Tiny ImageNet dataset. Notably, we use a 3 layer encoder for this dataset, which provides additional insights ensuring that U-Hop works well under deep neural network architecture.

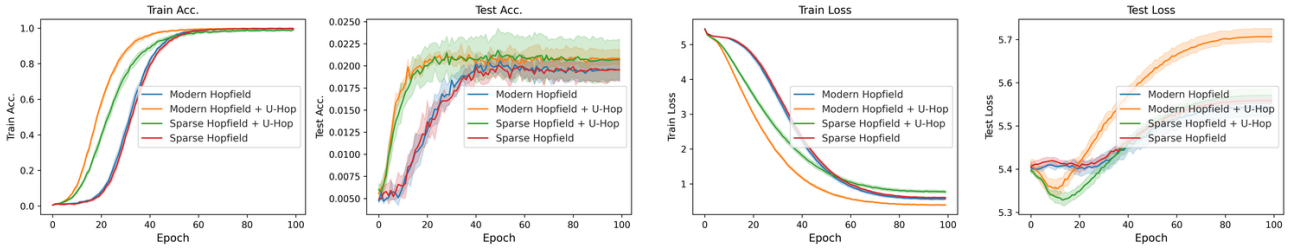

Figure 42. **Tiny ImageNet Convergence Comparison with Dataset Size=1000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

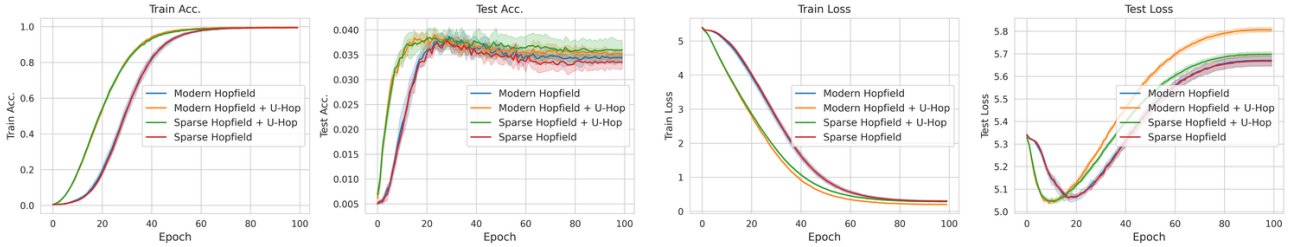

Figure 43. **Tiny ImageNet Convergence Comparison with Dataset Size=5000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

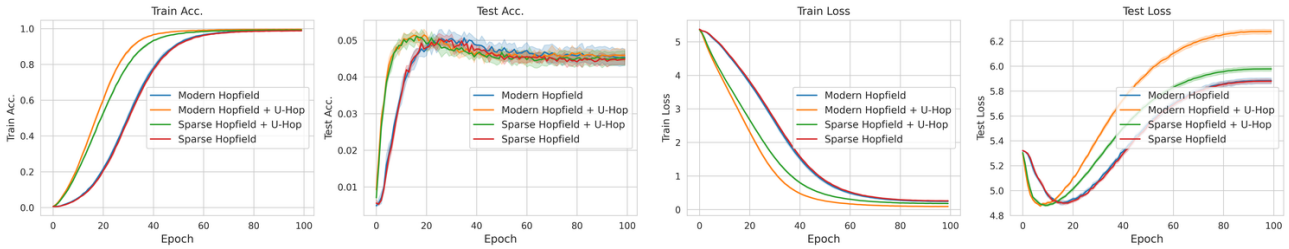

Figure 44. **Tiny ImageNet Convergence Comparison with Dataset Size=10000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

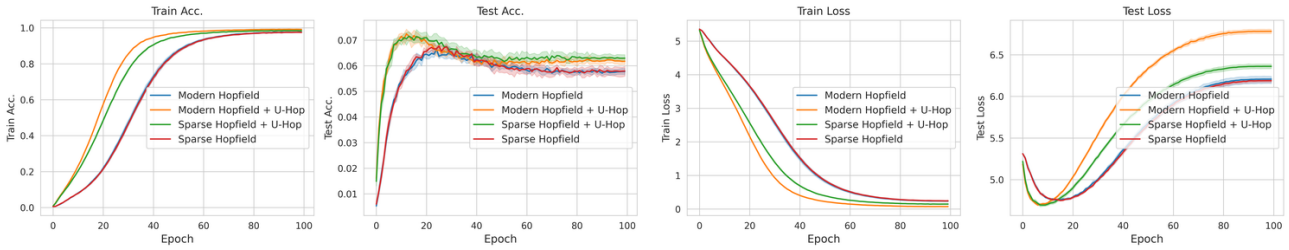

Figure 45. **Tiny ImageNet Convergence Comparison with Dataset Size=20000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

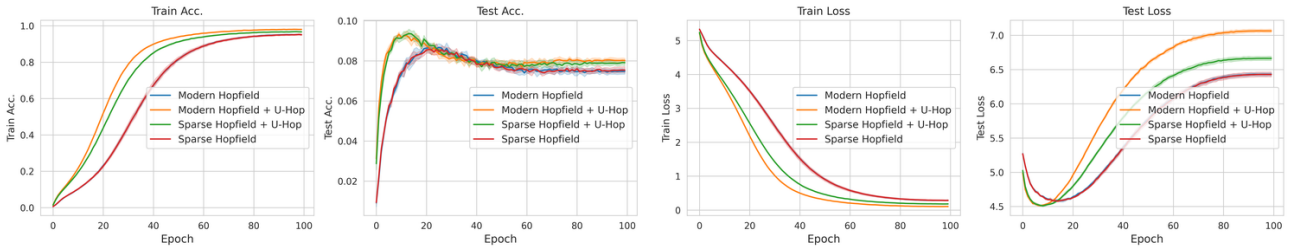

Figure 46. **Tiny ImageNet Convergence Comparison with Dataset Size=40000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

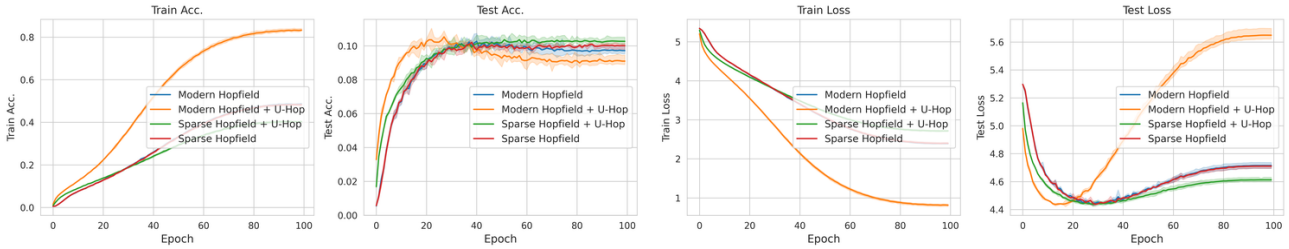

Figure 47. **Tiny ImageNet Convergence Comparison with Dataset Size=60000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

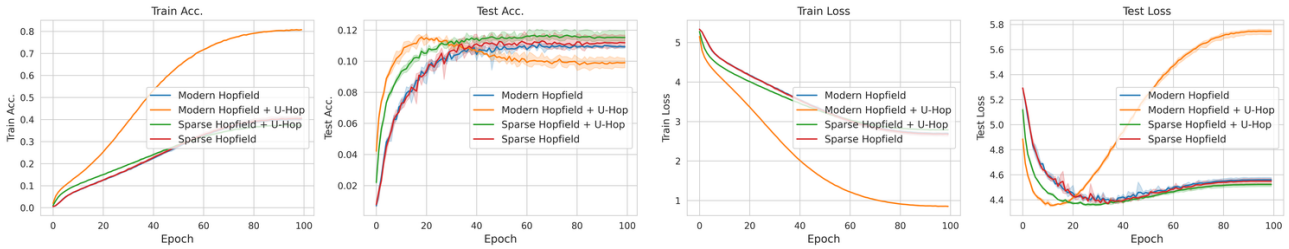

Figure 48. **Tiny ImageNet Convergence Comparison with Dataset Size=80000** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

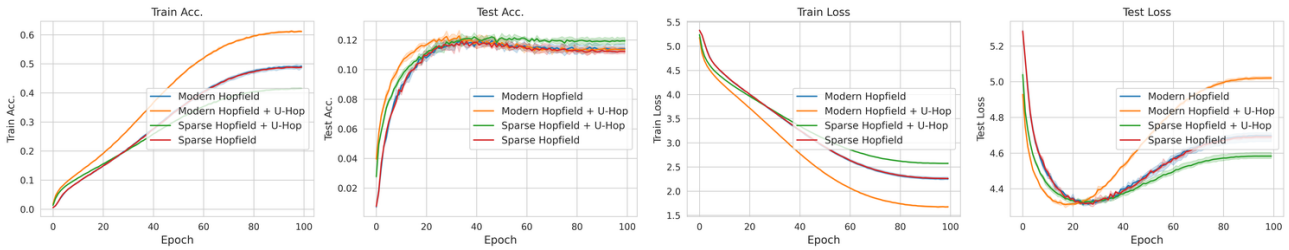

Figure 49. **Tiny ImageNet Convergence Comparison with Dataset Size=Full** Left to right: Training Accuracy, Test Accuracy, Training Loss and Test Loss.

G.6. Time Series Prediction

Here we report the results of our time series prediction experiment. From Table 8, we observe that `U-Hop` obtains improvement in most datasets and prediction horizons.

Table 8. **STanHop (Wu et al., 2024): Multivariate Time Series Predictions.** We compare **STanHop-Net** (Wu et al., 2024) with and without `U-Hop`. We report the average Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics with variance omitted as they are all $\leq 2\%$. We evaluated each dataset with different prediction horizons (shown in the second column). We have the best results **bolded**.

Models		**STanHop-Net**		**STanHop-Net** + `U-Hop`	
Metric		MSE	MAE	MSE	MAE
ETTh1	96	0.395	0.402	**0.392**	**0.400**
	192	0.425	0.432	**0.420**	**0.428**
	336	0.495	0.487	**0.470**	**0.473**
	720	0.631	0.600	**0.572**	**0.559**
ETTm1	96	0.334	0.366	**0.333**	**0.365**
	192	**0.351**	**0.380**	0.355	0.385
	336	**0.391**	**0.393**	0.392	0.399
	720	0.436	0.431	**0.435**	**0.423**
WTH	96	0.494	0.505	**0.483**	**0.498**
	192	**0.513**	**0.526**	0.528	0.536
	336	0.523	0.539	0.523	0.539
	720	**0.601**	0.609	0.603	**0.589**