Hopfield-Fenchel-Young Networks:
A Unified Framework for Associative Memory Retrieval

Saul Santos   SAUL.R.SANTOS@TECNICO.ULISBOA.PT
Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
Instituto de Telecomunicações, Lisbon, Portugal

Vlad Niculae   V.NICULAE@UVA.NL
Language Technology Lab, University of Amsterdam, The Netherlands

Daniel McNamee   DANIEL.MCNAMEE@RESEARCH.FCHAMPALIMAUD.ORG
Champalimaud Research, Lisbon, Portugal

André F. T. Martins   ANDRE.T.MARTINS@TECNICO.ULISBOA.PT
Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
Instituto de Telecomunicações, Lisbon, Portugal
ELLIS Unit Lisbon

Abstract

Associative memory models, such as Hopfield networks and their modern variants, have garnered renewed interest due to advancements in memory capacity and connections with self-attention in transformers. In this work, we introduce a unified framework—*Hopfield-Fenchel-Young networks*—which generalizes these models to a broader family of energy functions. Our energies are formulated as the difference between two Fenchel-Young losses: one, parameterized by a generalized entropy, defines the Hopfield scoring mechanism, while the other applies a post-transformation to the Hopfield output. By utilizing Tsallis and norm entropies, we derive end-to-end differentiable update rules that enable sparse transformations, uncovering new connections between loss margins, sparsity, and exact retrieval of single memory patterns. We further extend this framework to *structured* Hopfield networks using the SparseMAP transformation, allowing the retrieval of pattern associations rather than a single pattern. Our framework unifies and extends traditional and modern Hopfield networks and provides an energy minimization perspective for widely used post-transformations like ℓ_2-normalization and layer normalization—all through suitable choices of Fenchel-Young losses and by using convex analysis as a building block. Finally, we validate our Hopfield-Fenchel-Young networks on diverse memory recall tasks, including free and sequential recall. Experiments on simulated data, image retrieval, multiple instance learning, and text rationalization demonstrate the effectiveness of our approach.

Keywords: Hopfield Networks, Associative Memories, Sparse Transformations, Structured Prediction, Fenchel-Young Losses, Memory Retrieval.

1. Introduction

Hopfield networks are biologically plausible neural networks with associative memory capabilities (Amari, 1972; Nakano, 1972; Hopfield, 1982). Their attractor dynamics, which describe how the networks evolve toward stable states or patterns, make them suitable for

modeling associative memory retrieval in humans and animals (Tyulmankov et al., 2021; Whittington et al., 2021). The limited storage capacity of classical Hopfield networks was recently overcome through the proposal of new energy functions. These energies were initially proposed for dense associative models by Krotov and Hopfield (2016) and Demircigil et al. (2017) and later expanded to continuous states by Ramsauer et al. (2021), resulting in exponential storage capacity and renewed interest in "modern" Hopfield networks. In particular, Ramsauer et al. (2021) revealed connections to attention layers in transformers via an update rule linked to the convex-concave procedure (CCCP; Yuille and Rangarajan 2003). However, this model only *approximates* stored patterns, requiring careful temperature tuning to avoid converging to large metastable states that mix multiple input patterns.

A common element in many recent studies on Hopfield networks is connecting an energy function with a desired update rule, typically by modeling the network's temporal dynamics through differential equations, which are discretized to produce updates involving derivatives of Lagrangian terms in the energy function (Krotov and Hopfield, 2021). However, a comprehensive framework for formulating the equations that govern the dynamics of the entire spectrum of Hopfield networks is lacking—a gap we aim to fill by explicitly using convex analysis and Fenchel-Young duality as building blocks (Rockafellar, 1970; Bauschke and Combettes, 2017). This not only allows for designing new energy functions but also provides a way to understand functionalities in transformer architectures, like multi-head attention (Vaswani et al., 2017) and layer normalization (Ba et al., 2016), as well as other normalization techniques (Nguyen and Salazar, 2019). Developing a generalized framework is crucial for a unified theoretical basis to understand and extend Hopfield networks. This unified view facilitates comparative analysis, reveals underlying principles that govern their dynamics, and potentially leads to improvements in associative memory design and functionality.

Main contributions. The starting point of our paper is establishing a connection between Hopfield energies and **Fenchel-Young losses** (Blondel et al., 2020). Namely, we consider energy functions expressed as the difference of two Fenchel-Young loss terms induced by convex functions Ω and Ψ. These two terms serve distinct purposes: Ω contributes to the Hopfield scoring function, where the aim is to "separate" one pattern from the others, while Ψ acts as a regularizer, relating to the post-transformation applied in the Hopfield updates. Our proposed Hopfield-Fenchel-Young energies recover as particular cases a wide range of associative memory models, such as the classical binary and continuous Hopfield networks (Hopfield, 1982), polynomial and exponential dense associative memories (Krotov and Hopfield, 2016; Demircigil et al., 2017), the modern Hopfield networks from Ramsauer et al. (2021), as well as their sparse counterparts (Hu et al., 2023). Furthermore, we show that the Fenchel-Young loss associated with the Hopfield scoring function, when induced by certain generalized entropy functions Ω, can lead to **sparse update rules** which include as particular cases α-entmax (Peters et al., 2019), γ-normmax (Blondel et al., 2020), and SparseMAP (Niculae et al., 2018). The latter case allows general structural constraints to be incorporated in addition to sparsity, enabling the retrieval of **pattern associations**. We illustrate this with structural constraints which ensure the retrieval of k patterns, as well as a sequential variant which promotes the k patterns to be contiguous in a memorized sequence. One distinctive property of our sparsity-inducing generalized entropies compared to the Hopfield layers of Ramsauer et al. (2021) is that our resulting update rules pave the

way for **exact retrieval**, leading to the emergence of sparse association among patterns while ensuring end-to-end differentiability, a property which relates to the existence of a separation margin in the corresponding Fenchel-Young losses. In addition, our formulation allows for post-transformations in the Hopfield updates via Ψ, such as different kinds of normalization, which may accelerate convergence and have other beneficial properties. While some of these post-transformations have been considered before (Krotov and Hopfield, 2021), their contribution to the energy has often been left implicit. We derive in this paper the explicit energy terms, using classical results from convex duality.

Our endeavour aligns with the strong neurobiological motivation to seek new Hopfield energies capable of **sparse** and **structured** retrieval. Indeed, sparse neural activity patterns forming structured representations underpin core principles of cortical computation (Simoncelli and Olshausen, 2001; Tse et al., 2007; Palm, 2013). With respect to memory formation circuits, the sparse firing of neurons in the dentate gyrus, a distinguished region within the hippocampal network, underpins its proposed role in pattern separation during memory storage (Yassa and Stark, 2011; Severa et al., 2017). Evidence suggests that such sparsified activity aids in minimizing interference, however an integrative theoretical account linking sparse coding and attractor network functionality to clarify these empirical observations is lacking (Leutgeb et al., 2007; Neunuebel and Knierim, 2014).

To sum up, our main contributions are:

- We introduce **Hopfield-Fenchel-Young** energy functions as a generalization of modern and classical Hopfield networks (§3).

- We leverage properties of Fenchel-Young losses which relate **sparsity** to **margins**, obtaining new theoretical results for exact memory retrieval and proving exponential storage capacity in a stricter sense than prior work (§4).

- We propose new **structured** Hopfield networks via the SparseMAP transformation, which return **pattern associations** instead of single patterns. We show that SparseMAP has a structured margin, enabling exact retrieval of pattern associations (§5).

- We use our framework in memory retrieval modeling problems (§6).

Experiments on synthetic and real-world tasks (image retrieval, multiple instance learning, and text rationalization) showcase the usefulness of our proposed models using various kinds of sparse and structured transformations (§7). An overview of the Hopfield scoring functions studied in this paper is shown in Figure 1.[1]

Previous Paper. Our work builds upon a previously published conference paper (Santos et al., 2024) and a non-archival workshop paper (Martins et al., 2023), which we extend significantly in several ways. Santos et al. (2024) fix $\Psi(\cdot) = \frac{1}{2}\|\cdot\|^2$ and focus on Ω corresponding to sparsity-inducing generalized entropies. The current paper examines general Ψ, leading to more general Hopfield energies which are a difference of two Fenchel-Young losses; this step allows this construction to be unified with many other modern and classical Hopfield networks and enables the inclusion of a post-transformation step, such as ℓ_2 and layer normalization (for which we derive, in §3.3.6, an explicit energy minimization interpretation). We show how our expanded framework enables the creation of useful new

---

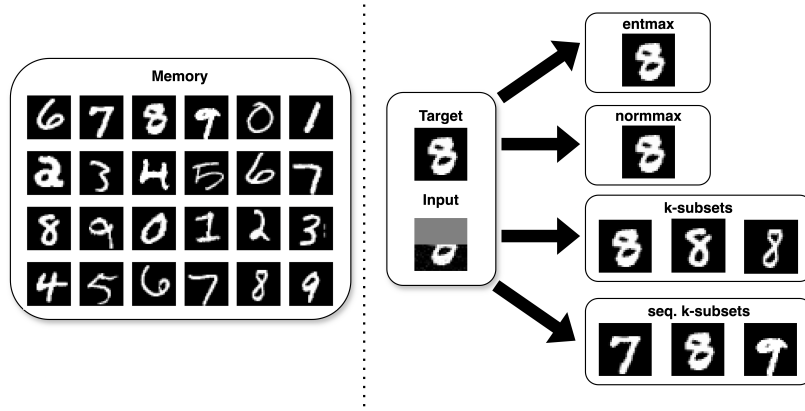1. Our code is made available on https://github.com/deep-spin/HFYN.

Figure 1: Overview of Hopfield scoring functions: sparse transformations (entmax and normmax) aim to retrieve the closest pattern to the query, and they have exact retrieval guarantees. Structured variants find pattern associations. The k-subsets transformation favors a mixture of the top-k patterns, and sequential k-subsets favors contiguous retrieval.

Hopfield networks, with any arbitrary post-transformation defined by a Fenchel-Young loss induced by any convex function. Our theoretical proofs are extended to support this generalization, and we empirically evaluate these variants in multiple instance learning and memory retrieval benchmarks. Overall, §3 and §6 are completely new, §7 contains many new experiments, and §4 and §5 have new proofs due to the inclusion of Ψ.

Notation. We denote by $\triangle_N$ the $(N-1)^{\text{th}}$-dimensional probability simplex, $\triangle_N := \{\boldsymbol{p} \in \mathbb{R}^N : \boldsymbol{p} \geq \boldsymbol{0}, \boldsymbol{1}^\top \boldsymbol{p} = 1\}$. The convex hull of a set $\mathcal{Y} \subseteq \mathbb{R}^M$ is $\operatorname{conv}(\mathcal{Y}) := \{\sum_{i=1}^N p_i \boldsymbol{y}_i : \boldsymbol{p} \in \triangle_N, \boldsymbol{y}_1, ..., \boldsymbol{y}_N \in \mathcal{Y}, N \in \mathbb{N}\}$. We have $\triangle_N = \operatorname{conv}(\{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\})$, where $\boldsymbol{e}_i \in \mathbb{R}^N$ is the ith basis (one-hot) vector. Given a convex function $\Omega : \mathbb{R}^N \to \bar{\mathbb{R}}$, where $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$, we denote its domain by $\operatorname{dom}(\Omega) := \{\boldsymbol{y} \in \mathbb{R}^N : \Omega(\boldsymbol{y}) < +\infty\}$ and its Fenchel conjugate by $\Omega^*(\boldsymbol{\theta}) = \sup_{\boldsymbol{y} \in \mathbb{R}^N} \boldsymbol{y}^\top \boldsymbol{\theta} - \Omega(\boldsymbol{y})$. We denote by $I_{\mathcal{C}}$ the indicator function of a convex set $\mathcal{C}$, defined as $I_{\mathcal{C}}(\boldsymbol{y}) = 0$ if $\boldsymbol{y} \in \mathcal{C}$, and $I_{\mathcal{C}}(\boldsymbol{y}) = +\infty$ otherwise.

Table of Contents

2. Background

2.1 Associative Memories and Hopfield Networks

In associative memories, data patterns are retrieved based on similarity to a given query, rather than through an explicit address. When a noisy cue of the memories is provided as the query, the aim is to retrieve the most similar memory pattern. Hopfield networks (Amari, 1972; Nakano, 1972; Hopfield, 1982) are neural models inspired by statistical physics, specifically by the Ising model (Ising, 1925), which describes a system of magnetic moments or "spins" of particles that can be in one of two states (± 1), interacting with each other to minimize the system's overall energy. In a classical Hopfield network (Hopfield, 1982), states are binary and interact through synaptic weights analogous to the spin-spin couplings in the Ising model. When a query is given as input, the network iteratively adjusts to minimize its energy, leading to the emergence of stable states or attractors, which correspond to stored memory patterns. Pioneering works on classic Hopfield networks, as well as subsequent research by Amit et al. (1985a) and Hertz et al. (1991), has demonstrated the ability of this approach to perform tasks like pattern retrieval and completion for binary data.

Formally, let $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ be a matrix whose rows hold a set of examples $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \in \mathbb{R}^D$ ("memory patterns"), and let $\boldsymbol{q}^{(0)} \in \mathbb{R}^D$ be the query vector (or "state pattern"). Hopfield networks iteratively update $\boldsymbol{q}^{(t)} \mapsto \boldsymbol{q}^{(t+1)}$ for $t \in \{0, 1, ...\}$ according to a certain rule, eventually converging to a fixed point attractor state $\boldsymbol{q}^*$, hopefully corresponding to one of the memorized examples. This update rule corresponds to the minimization of an energy function, which for a classical binary Hopfield network (Hopfield, 1982) takes the form

$$E(\boldsymbol{q}) = -\frac{1}{2}\|\boldsymbol{X}\boldsymbol{q}\|^2 = -\frac{1}{2}\boldsymbol{q}^\top \boldsymbol{W} \boldsymbol{q}, \tag{1}$$

where $\boldsymbol{W} = \boldsymbol{X}^\top \boldsymbol{X} \in \mathbb{R}^{D \times D}$ is a weight matrix and $\boldsymbol{q} \in \{\pm 1\}^D$ is a binary vector, leading to the update rule $\boldsymbol{q}^{(t+1)} = \text{sign}(\boldsymbol{W}\boldsymbol{q}^{(t)})$. In this classical construction, the stored patterns, as well as the queries, are assumed to be binary vectors in $\{\pm 1\}^D$. The discreteness of the update rule (due to the sign transformation) makes this network capable of **exact retrieval** (*i.e.*, it is able to retrieve memorized patterns perfectly, upon convergence). However, its main limitation is that it has only $N = \mathcal{O}(D)$ memory storage capacity. When this capacity is exceeded, the patterns start to interfere (Amit et al., 1985b; McEliece et al., 1987), resulting in the retrieval of metastable states or spurious attractor points.

2.2 Modern Hopfield Networks

More recent work has sidestepped the limitation above through alternative energy functions (Krotov and Hopfield, 2016; Demircigil et al., 2017), paving the way for a class of models known as "modern Hopfield networks" with superlinear (often exponential) memory capacity. In Ramsauer et al. (2021), $\boldsymbol{q} \in \mathbb{R}^D$ is continuous and the following energy is used:

$$E(\boldsymbol{q}) = -\frac{1}{\beta} \log \sum_{i=1}^{N} \exp(\beta \boldsymbol{x}_i^\top \boldsymbol{q}) + \frac{1}{2}\|\boldsymbol{q}\|^2 + \text{const.} \tag{2}$$

[Ramsauer et al. (2021)](#) revealed an interesting relation between the updates in this modern Hopfield network and the attention layers in transformers. Namely, the minimization of the energy (2) using the concave-convex procedure (CCCP; [Yuille and Rangarajan 2003](#)), also known as difference-of-convex (DC) programming ([Horst and Thoai, 1999](#)), leads to the update rule

$$\boldsymbol{q}^{(t+1)} = \boldsymbol{X}^\top \text{softmax}(\beta \boldsymbol{X} \boldsymbol{q}^{(t)}). \tag{3}$$

When $\beta = \frac{1}{\sqrt{D}}$, each update matches the computation performed in the attention layer of a transformer with a single attention head and identity projection matrices ([Ramsauer et al., 2021](#), §2). This triggered interest in developing variants of Hopfield layers which can be used as drop-in replacements for multi-head attention layers ([Hoover et al., 2023](#)).

While [Ramsauer et al. (2021)](#) derived useful theoretical properties of these networks (including their exponential storage capacity under a relaxed notion of retrieval), the use of softmax in (3), along to the fact that these networks now operate on a continuous space, makes retrieval only approximate (*i.e.*, the attractors are not the exact stored patterns, but only approximately close as $\beta \to \infty$), with some propensity for undesirable metastable states (states that mix multiple memory patterns). We overcome these drawbacks in our work by showing that it is possible to work on a continuous space but still obtain exact retrieval (as in binary Hopfield networks), without sacrificing exponential storage capacity. We generalize the energies (1) and (2), as well as several other proposed formulations of Hopfield-like models, and we provide a unified treatment of sparse and structured Hopfield networks along with a theoretical and empirical analysis.

2.3 Regularized Argmax and Fenchel-Young Losses

Our construction and results follow from the properties of regularized argmax functions and Fenchel-Young losses ([Blondel et al., 2020](#)), which we next review.

Given a strictly convex function $\Omega : \mathbb{R}^N \to \bar{\mathbb{R}}$ with compact domain $\text{dom}(\Omega)$, the Ω-regularized argmax transformation ([Niculae and Blondel, 2017](#)), $\hat{\boldsymbol{y}}_\Omega : \mathbb{R}^N \to \text{dom}(\Omega)$ is:

$$\hat{\boldsymbol{y}}_\Omega(\boldsymbol{\theta}) := \nabla \Omega^*(\boldsymbol{\theta}) = \underset{\boldsymbol{y} \in \text{dom}(\Omega)}{\text{argmax}}\ \boldsymbol{\theta}^\top \boldsymbol{y} - \Omega(\boldsymbol{y}), \tag{4}$$

where $\nabla \Omega^*(\boldsymbol{\theta})$ denotes the gradient of the Fenchel conjugate Ω^* at $\boldsymbol{\theta}$.[2] A trivial example of such a regularized argmax function (4) is obtained when $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}\|^2$ with $\text{dom}(\Omega) = \mathbb{R}^N$, which leads to the identity map $\boldsymbol{y}_\Omega(\boldsymbol{\theta}) = \boldsymbol{\theta}$. In general, we are interested in cases where $\text{dom}(\Omega) \subsetneq \mathbb{R}^N$, for example the probability simplex $\text{dom}(\Omega) = \triangle_N$ (studied in §4) or a polytope (studied in §5). A famous instance of the former is the **softmax** transformation, obtained when the regularizer is the Shannon negentropy, $\Omega(\boldsymbol{y}) = \sum_{i=1}^N y_i \log y_i + I_{\triangle_N}(\boldsymbol{y})$. Another instance is the **sparsemax** transformation, obtained when $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}\|^2 + I_{\triangle_N}(\boldsymbol{y})$ ([Martins and Astudillo, 2016](#)), and which corresponds to the Euclidean projection onto the probability simplex. In §4, we analyze a wider set of transformations induced by generalized entropies which include these as particular cases.

The **Fenchel-Young loss** induced by Ω ([Blondel et al., 2020](#)) is the function defined as

$$L_\Omega(\boldsymbol{\theta}, \boldsymbol{y}) := \Omega(\boldsymbol{y}) + \Omega^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \boldsymbol{y}. \tag{5}$$

---

2. Note that Ω^* is guaranteed to be everywhere differentiable under the assumption that $\text{dom}(\Omega)$ is compact and Ω is strictly convex, which together ensure existence and uniqueness of a maximum in (4).

In the trivial case above, where $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}\|^2$ with $\mathrm{dom}(\Omega) = \mathbb{R}^N$, we obtain $\Omega^*(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2$, leading to the squared loss, $L_\Omega(\boldsymbol{\theta}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{y}\|^2$. When Ω is the Shannon's negentropy defined above, we have $\Omega^*(\boldsymbol{\theta}) = \log \sum_{i=1}^N \exp(\theta_i)$, and L_Ω is the **cross-entropy loss**, up to a constant independent of $\boldsymbol{\theta}$. Intuitively, Fenchel-Young losses quantify how "compatible" a score vector $\boldsymbol{\theta} \in \mathbb{R}^N$ (*e.g.*, logits) is to a desired target $\boldsymbol{y} \in \mathrm{dom}(\Omega)$ (*e.g.*, a probability vector). Fenchel-Young losses have important and useful properties, summarized below:

> **Proposition 1 (Properties of Fenchel-Young losses)** *Fenchel-Young losses $L_\Omega(\boldsymbol{\theta}, \boldsymbol{y})$ satisfy the following properties:*
>
> 1. *They are non-negative, $L_\Omega(\boldsymbol{\theta}, \boldsymbol{y}) \geq 0$, with equality if and only if $\boldsymbol{y} = \hat{\boldsymbol{y}}_\Omega(\boldsymbol{\theta})$.*
>
> 2. *They are convex in $\boldsymbol{\theta}$, and their gradient is $\nabla_{\boldsymbol{\theta}} L_\Omega(\boldsymbol{\theta}, \boldsymbol{y}) = -\boldsymbol{y} + \hat{\boldsymbol{y}}_\Omega(\boldsymbol{\theta})$.*

These properties are stated and proved in Blondel et al. (2020, Proposition 2). For certain choices of Ω, Fenchel-Young losses also have an important margin property, which we will elaborate on in §4 and which underpins our proposed Hopfield energies with exact retrieval.

3. Hopfield-Fenchel-Young Energies

We now use Ω-regularized argmax transformations and Fenchel-Young losses to define a new class of energy functions associated to modern Hopfield networks. Historically, the connection between sparse Hopfield energy functions and Fenchel conjugates was first made by Hu et al. (2023) and extended by Martins et al. (2023), Wu et al. (2024b), and Santos et al. (2024).

3.1 Definition

Let $\Omega : \mathbb{R}^N \to \bar{\mathbb{R}}$ and $\Psi : \mathbb{R}^D \to \bar{\mathbb{R}}$ be convex functions, and $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ be the matrix of memory patterns. We consider energy functions of the form $E(\boldsymbol{q}) = -\Omega^*(\boldsymbol{X}\boldsymbol{q}) + \Psi(\boldsymbol{q})$ (up to a constant), defined for $\boldsymbol{q} \in \mathrm{dom}(\Psi) \subseteq \mathbb{R}^D$. These energy functions can be equivalently written as a difference of Fenchel-Young losses (cf. (5)):

$$E(\boldsymbol{q}) = \underbrace{-L_\Omega(\boldsymbol{X}\boldsymbol{q}, \boldsymbol{u})}_{E_{\mathrm{concave}}(\boldsymbol{q})} + \underbrace{L_\Psi(\boldsymbol{X}^\top \boldsymbol{u}, \boldsymbol{q})}_{E_{\mathrm{convex}}(\boldsymbol{q})} + \mathrm{const.}, \tag{6}$$

where $\boldsymbol{u} \in \mathrm{dom}(\Omega) \subseteq \mathbb{R}^N$ is an arbitrary baseline.[3] We call (6) a **Hopfield-Fenchel-Young (HFY) energy**. We will see that energies of this form are general enough to encompass most previously proposed variants of Hopfield energies and to motivate new ones.

The first thing to observe is that HFY energies decompose as the sum of a concave and a convex function of $\boldsymbol{q}$: The concavity of E_{concave} holds from the convexity of Fenchel-Young losses on their first argument, as established in Proposition 1, and from the fact that the composition of a convex function with an affine map is convex (Boyd and Vandenberghe, 2004, §3.2). The convexity of E_{convex} comes from the fact that Fenchel-Young losses are also

---

3. The value of $E(\boldsymbol{q})$ does not depend on the choice of $\boldsymbol{u}$; without loss of generality we define $\boldsymbol{u} = (\nabla\Omega^*)(\boldsymbol{0}) = \mathrm{argmin}_{\boldsymbol{y}} \Omega(\boldsymbol{y})$ assuming that this argmin exists. When $\mathrm{dom}(\Omega) = \triangle_N$ and Ω is a generalized negentropy—the scenario we study in §4—this choice leads to $\boldsymbol{u} = \mathbf{1}/N$ being a uniform distribution.

convex on their second argument when Ψ is strictly convex. These two terms compete when minimizing the energy (6) with respect to $\boldsymbol{q}$:

- Minimizing E_{concave} is equivalent to *maximizing* $L_\Omega(\boldsymbol{Xq}; \boldsymbol{u})$, which pushes for state patterns $\boldsymbol{q}$ such that $\boldsymbol{Xq}$ is as far as possible from the baseline $\boldsymbol{u}$. We will see later that this will encourage $\boldsymbol{q}$ to be close to a single memory pattern.

- Minimizing E_{convex} serves as a regularization, encouraging $\boldsymbol{q}$ to stay close to $\boldsymbol{X}^\top \boldsymbol{u}$. More intuition about this regularization will be provided in §4.4.

3.2 Update rule

The next result, proved in Appendix B.1, leverages Proposition 1 to derive the Hopfield update rule for the energy function (6), generalizing Ramsauer et al. (2021, Theorem A.1).

> **Proposition 2 (Update rule of HFY energies)** *Minimizing* (6) *using the CCCP algorithm (Yuille and Rangarajan, 2003) leads to the updates*
>
> $$\boldsymbol{q}^{(t+1)} = (\nabla\Psi^*)\left(\boldsymbol{X}^\top \nabla\Omega^*(\boldsymbol{Xq}^{(t)})\right) = \hat{\boldsymbol{y}}_\Psi\left(\boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\boldsymbol{Xq}^{(t)})\right). \tag{7}$$

Note that the updates (7) involve the functions Ω and Ψ only via the gradient maps $\nabla\Omega^*$ and $\nabla\Psi^*$, which are the regularized prediction functions $\hat{\boldsymbol{y}}_\Omega$ and $\hat{\boldsymbol{y}}_\Psi$ (cf. 4). Another way of looking into (7) is to unpack these updates in terms of composing four operations: similarity, separation, projection, and post-transformation. This can be expressed by

$$\boldsymbol{q}^{(t+1)} = \underbrace{\text{normalize}}_{\text{Post-transformation}}\left(\underbrace{\boldsymbol{X}^\top}_{\text{Projection}} \cdot \underbrace{\text{sep}}_{\text{Separation}}\left(\underbrace{\text{sim}(\boldsymbol{X}, \boldsymbol{q}^{(t)})}_{\text{Similarity}}\right)\right). \tag{8}$$

The update rule (7) is obtained when the similarity operation is the dot product, the separation corresponds to $\hat{\boldsymbol{y}}_\Omega$, and the post-transformation corresponds to $\hat{\boldsymbol{y}}_\Psi$. This is consistent with the universal Hopfield network of Millidge et al. (2022), where the post-transformation is the identity function. The update rule (7) supports only the dot product as similarity function, although other similarities can be introduced as long as the gradient of the similarity function is also projected.

3.3 Particular cases

We show now that energies of the form (6) recover many known variants of Hopfield networks and suggest new ones. These variants are obtained through particular choices of the Ω and Ψ functions, which lead to the corresponding conjugates Ω^* and Ψ^*. We start by examining key properties of convex duality that are instrumental for developing Hopfield energies. The convex conjugate, also known as the Legendre-Fenchel transform (Fenchel, 1949), is a cornerstone of convex analysis and optimization, allowing to convert complex optimization problems into more manageable dual forms. Table 1 highlights several properties and examples which establish the building blocks for designing new classes of Hopfield networks, as described in the subsequent subsections. Table 2 provides a summary of the several examples arising from the simple construction presented in §3.1 by leveraging the properties in Table 1.

Table 1: Properties/examples of convex conjugates (Boyd and Vandenberghe, 2004, §3.3).

Property	Expression / Description
Biconjugate	$\Omega^{**} = \Omega,$ if Ω is closed and convex
Linear transformations	If $\boldsymbol{A}$ is squared and non-singular, $(\Omega \circ \boldsymbol{A})^* = \Omega^* \circ \boldsymbol{A}^{-\top}$
Scaling	For $a > 0$, $(a\Omega)^*(\boldsymbol{\theta}) = a\Omega^*\left(\frac{\boldsymbol{y}}{a}\right)$
Translation	If $\Omega(\boldsymbol{y}) = \Psi(\boldsymbol{y} - \boldsymbol{b})$, then $\Omega^*(\boldsymbol{\theta}) = \boldsymbol{b}^\top \boldsymbol{\theta} + \Psi^*(\boldsymbol{\theta})$
Separable sum	If $\Omega(\boldsymbol{y}) = \sum_i \Omega_i(y_i)$, then $\Omega^*(\boldsymbol{\theta}) = \sum_i \Omega_i^*(\theta_i)$
Infimal convolution	$(\Omega_1 \square \Omega_2)^*(\boldsymbol{\theta}) = \Omega_1^*(\boldsymbol{\theta}) + \Omega_2^*(\boldsymbol{\theta})$ (see (16))
Dual norms	If $\Omega(\boldsymbol{y}) = \frac{1}{p}\|\boldsymbol{y}\|_p^p$, then $\Omega^*(\boldsymbol{\theta}) = \frac{1}{q}\|\boldsymbol{\theta}\|_q^q$ with $\frac{1}{p} + \frac{1}{q} = 1$
Exponential	If $\Omega(y) = \exp(y)$, then $\Omega^*(\theta) = \theta \log \theta - \theta + I_{\mathbb{R}_+}(\theta)$
Negentropy	If $\Omega(\boldsymbol{y}) = \sum_i y_i \log y_i + I_{\triangle^N}(\boldsymbol{y})$, then $\Omega^*(\boldsymbol{\theta}) = \log \sum_i \exp(\theta_i)$

3.3.1 Classical Hopfield networks

We denote the Fermi-Dirac entropy (sum of independent binary entropies) by $H_{\mathrm{b}}(\boldsymbol{y}) = -\sum_{i=1}^N (y_i \log y_i - (1 - y_i) \log(1 - y_i))$, where $\boldsymbol{y} \in [0, 1]^N$. Classical (continuous) Hopfield networks are recovered in (6) with

$$\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}\|^2 \quad \text{and} \quad \Psi(\boldsymbol{q}) = -\beta^{-1}H_{\mathrm{b}}\left(\frac{1+\boldsymbol{q}}{2}\right) + I_{[-1,1]^D}(\boldsymbol{q}), \tag{9}$$

where $\beta^{-1} \geq 0$ is a temperature parameter. In this case, $\hat{\boldsymbol{y}}_\Omega$ is the identity and $\hat{\boldsymbol{y}}_\Psi$ is the `tanh` transformation with temperature β^{-1}, leading to the update rule $\boldsymbol{q}^{(t+1)} = \tanh\left(\beta \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{q}^{(t)}\right)$. Binary Hopfield networks (Hopfield, 1982) appear as a limit case when $\beta \to +\infty$.[4]

3.3.2 Dense associative memories (DAMs)

DAMs correspond to energy functions $E(\boldsymbol{q}) = -\sum_{i=1}^N F(\boldsymbol{q}^\top \boldsymbol{x}_i)$. Poly-DAMs use $F(z) = |z|^r$ and are constrained to $\boldsymbol{q} \in \{\pm 1\}^D$ (Krotov and Hopfield, 2016). These energies can be equivalently written (up to a scaling factor) as

$$E(\boldsymbol{q}) = -r^{-1}\|\boldsymbol{X}\boldsymbol{q}\|_r^r + I_{[-1,1]^D}(\boldsymbol{q}). \tag{10}$$

This corresponds to choosing $\Omega(\boldsymbol{y}) = -s^{-1}\|\boldsymbol{y}\|_s^s$, where $r^{-1} + s^{-1} = 1$ ($\|.\|_r$ and $\|.\|_s$ are dual norms), and $\Psi(\boldsymbol{q})$ as in classical Hopfield networks. We have $(\nabla\Omega^*)(\boldsymbol{\theta}) = \mathrm{sign}(\boldsymbol{\theta})|\boldsymbol{\theta}|^{r-1} =:$ $\mathrm{spow}(\boldsymbol{\theta}, r-1)$, where `spow` denotes the signed power function. The update rule is

$$\boldsymbol{q}^{(t+1)} = \tanh\left(\beta \boldsymbol{X}^\top \mathrm{spow}(\boldsymbol{X}\boldsymbol{q}^{(t)}, r-1)\right). \tag{11}$$

---

4. We can obtain the same result for the classical binary Hopfield network by letting $\Psi(\boldsymbol{q}) = I_{\|.\|_\infty \leq 1}(\boldsymbol{q})$, in which case $\Psi^*(\boldsymbol{z}) = \|\boldsymbol{z}\|_1$ and $(\nabla\Psi^*)(\boldsymbol{z}) = \mathrm{sign}(\boldsymbol{z})$.

Table 2: **Examples of HFY energies and corresponding update rules.** The top rows show classic Hopfield networks and dense associative memories (we show continuous variants; binary variants are a limit case when $\beta^{-1} \to 0^+$ in $\Psi(\boldsymbol{q})$). In DAMs, we set $r^{-1} + s^{-1} = 1$ and denote by $\mathrm{spow}(\boldsymbol{\theta}, \alpha) = \mathrm{sign}(\boldsymbol{\theta}) \odot |\boldsymbol{\theta}|^\alpha$ the signed power function. The middle rows show modern Hopfield networks (MHNs) associated to probabilistic prediction, where $\mathrm{dom}(\Omega) = \triangle_N$. Our paper focuses on sparse variants of these models (§4). Finally, the last row generalizes the setups above to structured memories over a structured set $\mathcal{Y}$, also addressed in this paper (§5).

	$\Omega(\boldsymbol{y})$	$\mathrm{dom}(\Omega)$	$\Psi(\boldsymbol{q})$	$\mathrm{dom}(\Psi)$	Update rule ($\boldsymbol{q}^{(t+1)} = ...$)
Classic HNs (Hopfield, 1982)	$\frac{1}{2}\|\boldsymbol{y}\|^2$	$\mathbb{R}^N$	$-\frac{1}{\beta}H_{\mathrm{b}}\left(\frac{1+\boldsymbol{q}}{2}\right)$	$[-1,1]^D$	$\tanh(\beta \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{q}^{(t)})$
Poly-DAMs (Krotov and Hopfield, 2016)	$s^{-1}\|\boldsymbol{y}\|_s^s$	$\mathbb{R}^N$	$-\frac{1}{\beta}H_{\mathrm{b}}\left(\frac{1+\boldsymbol{q}}{2}\right)$	$[-1,1]^D$	$\tanh(\beta \boldsymbol{X}^\top \mathrm{spow}(\boldsymbol{X}\boldsymbol{q}^{(t)}, r-1))$
Exp-DAMs (Demircigil et al., 2017)	$\boldsymbol{y}^\top(\log \boldsymbol{y} - \mathbf{1})_+$	$\mathbb{R}_+^N$	$-\frac{1}{\beta}H_{\mathrm{b}}\left(\frac{1+\boldsymbol{q}}{2}\right)$	$[-1,1]^D$	$\tanh(\beta \boldsymbol{X}^\top \exp(\boldsymbol{X}\boldsymbol{q}^{(t)})$
MHNs (Ramsauer et al., 2021)	$\frac{1}{\beta}\boldsymbol{y}^\top \log \boldsymbol{y}$	$\triangle_N$	$\frac{1}{2}\|\boldsymbol{q}\|^2$	$\mathbb{R}^D$	$\boldsymbol{X}^\top \mathrm{softmax}(\beta \boldsymbol{X}\boldsymbol{q}^{(t)})$
Sparse MHNs (Hu et al., 2023)	$\frac{1}{\beta}\frac{1}{2}\|\boldsymbol{y}\|^2$	$\triangle_N$	$\frac{1}{2}\|\boldsymbol{q}\|^2$	$\mathbb{R}^D$	$\boldsymbol{X}^\top \mathrm{sparsemax}(\beta \boldsymbol{X}\boldsymbol{q}^{(t)})$
Entmax MHNs (**this work** and Wu et al. 2024b)	$\frac{1}{\beta}\frac{\|\boldsymbol{y}\|_\alpha^\alpha - 1}{\alpha(\alpha-1)}$	$\triangle_N$	$\frac{1}{2}\|\boldsymbol{q}\|^2$	$\mathbb{R}^D$	$\boldsymbol{X}^\top \alpha\text{-}\mathrm{entmax}(\beta \boldsymbol{X}\boldsymbol{q}^{(t)})$
Normmax MHNs (**this work**)	$\frac{1}{\beta}(\|\boldsymbol{y}\|_\gamma - 1)$	$\triangle_N$	$\frac{1}{2}\|\boldsymbol{q}\|^2$	$\mathbb{R}^D$	$\boldsymbol{X}^\top \gamma\text{-}\mathrm{normmax}(\beta \boldsymbol{X}\boldsymbol{q}^{(t)})$
Structured MHNs (**this work**)	$\frac{1}{\beta}\frac{1}{2}\|\boldsymbol{y}\|^2$	$\mathrm{conv}(\mathcal{Y})$	$\frac{1}{2}\|\boldsymbol{q}\|^2$	$\mathbb{R}^D$	$\boldsymbol{X}^\top \mathrm{SparseMAP}(\beta \boldsymbol{X}\boldsymbol{q}^{(t)})$

Likewise, the Exp-DAM of Demircigil et al. (2017), which uses $F(z) = \exp(z)$, can be written as $E(\boldsymbol{q}) = -\mathbf{1}^\top \exp(\boldsymbol{X}\boldsymbol{q}) + I_{[-1,1]^D}(\boldsymbol{q})$. It corresponds to choosing

$$\Omega(\boldsymbol{y}) = \sum_{j=1}^{D}[y_j \log y_j - y_j]_+ + I_{\mathbb{R}_+^N}(\boldsymbol{y}) \quad \text{and} \quad \Psi(\boldsymbol{q}) = \beta^{-1}H_{\mathrm{b}}\left(\frac{1+\boldsymbol{q}}{2}\right) + I_{[-1,1]^D}(\boldsymbol{q}). \quad (12)$$

We have $(\nabla\Omega^*)(\boldsymbol{\theta}) = \exp(\boldsymbol{\theta})$. The update rule is $\boldsymbol{q}^{(t+1)} = \tanh\left(\beta\boldsymbol{X}^\top \exp(\boldsymbol{X}\boldsymbol{q}^{(t)})\right)$.

3.3.3 MODERN HOPFIELD NETWORKS (MHNS)

The energy (2) of Ramsauer et al. (2021) is recovered when $\Omega(\boldsymbol{y}) = \beta^{-1}\sum_{i=1}^{N} y_i \log y_i + I_{\triangle_N}(\boldsymbol{y})$, the Shannon negentropy with temperature β^{-1}, and when $\Psi(\boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{q}\|^2$. In this case, $\hat{\boldsymbol{y}}_\Psi$ is the identity and $\hat{\boldsymbol{y}}_\Omega$ is the softmax transformation with temperature β^{-1}, leading to the update rule

$$\boldsymbol{q}^{(t+1)} = \boldsymbol{X}^\top \mathrm{softmax}(\beta\boldsymbol{X}\boldsymbol{q}^{(t)}). \quad (13)$$

The sparse MHNs of Hu et al. (2023) modify Ω to $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}\|^2 + I_{\triangle_N}(\boldsymbol{y})$, which leads to similar updates but where softmax is replaced by sparsemax. We present in §4 a generalization using the Tsallis α-negentropy, which is also sparse for $\alpha > 1$ and was proposed independently

by Martins et al. (2023) and Wu et al. (2024b). We also study the γ-normmax negentropy, proposed by Santos et al. (2024), which also leads to sparse MHNs.

3.3.4 HETERO-ASSOCIATIVE MEMORIES

The convex function $\Psi(\boldsymbol{q})$ can be used to induce outer projection operations, resulting in hetero-associativity through a matrix $\boldsymbol{A}$, similar to the hetero-associative memories described by Millidge et al. (2022) and Wu et al. (2024a). To do that, $\Psi(\boldsymbol{q})$ is defined as $\Psi(\boldsymbol{q}) = \frac{1}{2}\boldsymbol{q}^\top \boldsymbol{A}^{-1}\boldsymbol{q}$, where $\boldsymbol{A}$ is a symmetric and positive definite matrix, which leads to the gradient map $(\nabla \Psi^*)(\boldsymbol{z}) = \boldsymbol{A}\boldsymbol{z}$. When $\boldsymbol{z} = \boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\boldsymbol{X}\boldsymbol{q})$, we are effectively using the matrix $\boldsymbol{A}$ to project the associative space into a hetero-associative memory. This approach follows closely the outer projection considered in Ramsauer et al. (2021)'s Hopfield layers.

3.3.5 STRUCTURED ENERGIES

The framework of MHNs can be extended to the case where $\mathrm{dom}(\Omega)$ is a polytope, which is more general than the simplex $\triangle_N$. A vertex of this polytope might indicate an association among memory patterns informed by some desired structure. We develop this scenario in §5.

3.3.6 NORMALIZATION OPERATIONS

The function Ψ can also be used to induce a post-normalization operation, as hinted in the classical and dense associative cases highlighted above. Post-normalization was discussed by Millidge et al. (2022). One way to do this is to define $\Psi(\boldsymbol{q}) = I_{\|.\| \leq r}(\boldsymbol{q})$, whose Fenchel conjugate is $\Psi^*(\boldsymbol{z}) = r\|\boldsymbol{z}\|$ and has gradient map $(\nabla \Psi^*)(\boldsymbol{z}) = \frac{r\boldsymbol{z}}{\|\boldsymbol{z}\|}$, which corresponds to ℓ_2-**normalization**. This normalization technique was explored by Krotov and Hopfield (2021). An alternative **layer normalization** approach was proposed by Hoover et al. (2023), but without explicitly deriving the underlying energy term.

We establish next a result, proved in Appendix B.2, which derives explicitly the energy term (via $\Psi(\boldsymbol{q})$) which gives rise to the layer normalization post-transformation in HFY networks. To the best of our knowledge, this result has never been explicitly derived before.

Proposition 3 (Layer normalization) *Consider the layer normalization map*

$$\mathrm{LayerNorm}(\boldsymbol{z}; \eta, \boldsymbol{\delta}) := \eta \frac{\boldsymbol{z} - \mu_{\boldsymbol{z}}}{\sqrt{\frac{1}{D}\sum_i (z_i - \mu_{\boldsymbol{z}})^2}} + \boldsymbol{\delta}, \tag{14}$$

where $\eta > 0$ and $\boldsymbol{\delta} \in \mathbb{R}^D$ are arbitrary parameters, and where $\mu_{\boldsymbol{z}} := \frac{1}{D}\mathbf{1}^\top \boldsymbol{z}$. If we choose

$$\Psi(\boldsymbol{q}) := I_S(\boldsymbol{q}) \quad with \quad S := \left\{ \boldsymbol{q} \ : \ \|\boldsymbol{q} - \boldsymbol{\delta}\| \leq \eta\sqrt{D} \ \wedge \ \mathbf{1}^\top(\boldsymbol{q} - \boldsymbol{\delta}) = 0 \right\}, \tag{15}$$

then we have that $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = (\nabla \Psi^*)(\boldsymbol{z}) = \mathrm{LayerNorm}(\boldsymbol{z}; \eta, \boldsymbol{\delta})$.

3.3.7 SUM OF ENERGY FUNCTIONS AND INFIMAL CONVOLUTIONS

It may be convenient to consider sums of energy functions, as done recently by Hoover et al. (2023). Let Ω_1 and Ω_2 be two functions and suppose we would like to have $\Omega^* = \Omega_1^* + \Omega_2^*$. The corresponding function Ω is the infimal convolution $\Omega = \Omega_1 \square \Omega_2$ (Bauschke and Combettes,

2017, §12), defined as

$$(\Omega_1 \square \Omega_2)(\boldsymbol{y}) := \inf_{\boldsymbol{z} \in \mathbb{R}^N} \left[\Omega_1(\boldsymbol{y} - \boldsymbol{z}) + \Omega_2(\boldsymbol{z})\right]. \tag{16}$$

This leads to the desired convex conjugate $\Omega^* = (\Omega_1 \square \Omega_2)^* = \Omega_1^* + \Omega_2^*$, and therefore to the sum of the gradient maps $\hat{\boldsymbol{y}}_\Omega = \hat{\boldsymbol{y}}_{\Omega_1} + \hat{\boldsymbol{y}}_{\Omega_2}$.

4. Sparse Hopfield Networks

In this section we assume that $\text{dom}(\Omega) = \triangle_N$ (the probability simplex). We now use the general result derived in Proposition 2 to define a particular instance of sparse energy functions for modern Hopfield networks. Then, we study their properties by making a connection to margins in Fenchel-Young losses.

4.1 Generalized negentropies

A **generalized (neg)entropy** (Grünwald and Dawid, 2004; Amigó et al., 2018), formally defined below, provides a flexible framework for measuring disorder or uncertainty.

Definition 4 $\Omega : \triangle_N \to \mathbb{R}$ *is a generalized negentropy iff it satisfies the conditions*

1. *Zero negentropy:* $\Omega(\boldsymbol{y}) = 0$ *if* $\boldsymbol{y}$ *is a one-hot vector, i.e.,* $\boldsymbol{y} = \boldsymbol{e}_i$ *for any* $i \in \{1, ..., N\}$.

2. *Strict convexity:* $\Omega\left((1-\lambda)\boldsymbol{y} + \lambda\boldsymbol{y}'\right) < (1-\lambda)\Omega(\boldsymbol{y}) + \lambda\Omega(\boldsymbol{y}')$ *for* $\lambda \in]0,1[$ *and* $\boldsymbol{y}, \boldsymbol{y}' \in \triangle_N$ *with* $\boldsymbol{y} \neq \boldsymbol{y}'$.

3. *Permutation invariance:* $\Omega(\boldsymbol{P}\boldsymbol{y}) = \Omega(\boldsymbol{y})$ *for any permutation matrix* $\boldsymbol{P}$ *(i.e., a square matrix with a single 1 in each row and each column, and zero elsewhere).*

This definition implies that $\Omega \leq 0$ and that Ω is minimized when $\boldsymbol{y} = \boldsymbol{1}/N$, the uniform distribution (Blondel et al., 2020, Proposition 4), which justifies the name "generalized negentropy." We next discuss choices of Ω which lead to sparse alternatives to softmax.

4.2 Sparse transformations

In §2.3 we saw sparsemax, which is an example of a sparse transformation. In fact, the softmax and sparsemax transformations are both particular cases of a broader family of **α-entmax transformations** (Peters et al., 2019), parametrized by a scalar $\alpha \geq 0$ (called the entropic index). These transformations correspond to the following choice of regularizer Ω, called the **Tsallis α-negentropy** (Tsallis, 1988):

$$\Omega_\alpha^T(\boldsymbol{y}) = \begin{cases} \frac{-1+\|\boldsymbol{y}\|_\alpha^\alpha}{\alpha(\alpha-1)} + I_{\triangle_N}(\boldsymbol{y}), & \text{if } \alpha \neq 1 \\ \sum_i y_i \log y_i, & \text{if } \alpha = 1. \end{cases} \tag{17}$$

When $\alpha = 1$, the Tsallis α-negentropy Ω_α^T reduces to the Shannon's negentropy, leading to the softmax transformation. When $\alpha = 2$, it becomes the Gini negentropy, which equals the ℓ_2-norm (up to a constant), leading to the sparsemax transformation (Martins and Astudillo, 2016).
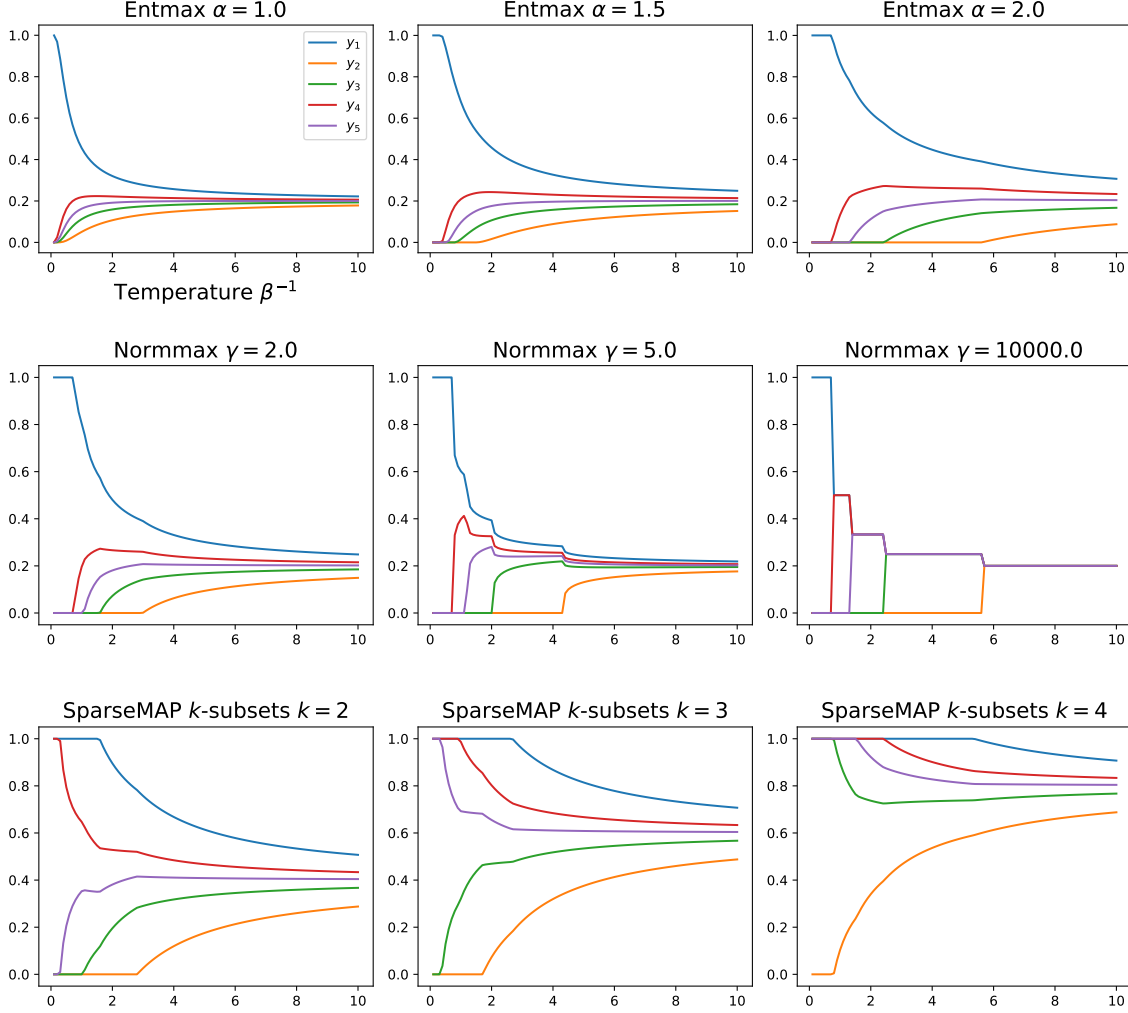
Figure 2: Sparse and structured transformations used in this paper and their regularization path. In each plot, we show $\hat{\boldsymbol{y}}_\Omega(\beta\boldsymbol{\theta}) = \hat{\boldsymbol{y}}_{\beta^{-1}\Omega}(\boldsymbol{\theta})$ as a function of the temperature β^{-1} where $\boldsymbol{\theta} = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425]^\top$.

Another example is the **norm γ-negentropy** (Blondel et al., 2020, §4.3),

$$\Omega_\gamma^N(\boldsymbol{y}) := -1 + \|\boldsymbol{y}\|_\gamma + I_{\triangle_N}(\boldsymbol{y}), \tag{18}$$

which, when $\gamma \to +\infty$, is called the Berger-Parker dominance index (May, 1975), widely used in ecology to measure the diversity of a species within a community. We call the resulting transformation γ-**normmax**. While the Tsallis and norm negentropies have similar expressions and the resulting transformations both tend to be sparse, they have important differences, as suggested in Figure 2: normmax favors distributions closer to uniform over the selected support. We will come back to these properties in the subsequent sections.

The examples above are all instances of transformations induced by generalized negentropies (Blondel et al., 2020): Tsallis negentropies (17) for $\alpha \geq 1$ and norm negentropies (18) for $\gamma > 1$ both satisfy the properties stated in §4.1.

4.3 Sparsity and margins

As seen in §2.3, convex regularizers Ω can be used to define not only a regularized argmax transformation, but also a Fenchel-Young loss. What happens when Ω is a sparsity-inducing generalized negentropy? We next show that, in addition to the general properties mentioned in Proposition 1, Fenchel-Young losses induced by such negentropies also satisfy a **margin** property, which, as we shall see, plays a pivotal role in the convergence and storage capacity of the class of Hopfield networks to be studied in §4 and §5.

> **Definition 5 (Margin)** *A loss function $L(\boldsymbol{\theta}; \boldsymbol{y})$ has a **margin** if there is a finite $m \geq 0$ such that*
>
> $$\forall i \in [N], \quad L(\boldsymbol{\theta}, \boldsymbol{e}_i) = 0 \iff \theta_i - \max_{j \neq i} \theta_j \geq m. \tag{19}$$
>
> *The smallest such m is called the margin of L. If L_Ω is a Fenchel-Young loss, (19) is equivalent to $\hat{\boldsymbol{y}}_\Omega(\boldsymbol{\theta}) = \boldsymbol{e}_i$.*

A famous example of a loss with a margin of 1 is the hinge loss of support vector machines. On the other hand, the cross-entropy loss does not have a margin, as suggested in Figure 3: it never reaches exactly zero for $\boldsymbol{\theta} = [s, 0] \in \mathbb{R}^2$, unlike the α-entmax and γ-normmax losses. We then have the following result, proved by Blondel et al. (2020):

> **Proposition 6 (Margin Properties of Tsallis and Norm-Entropies)** *Tsallis and norm negentropies have the following margins:*
>
> 1. ***Tsallis negentropies** Ω_α^T with $\alpha > 1$ lead to a loss $L_{\Omega_\alpha^T}$ with a margin $m = (\alpha-1)^{-1}$.*
>
> 2. ***Norm-negentropies** Ω_γ^N with $\gamma > 1$ lead to a loss $L_{\Omega_\gamma^N}$ with a margin $m = 1$, independent of γ.*

In fact, the sparsity of $\hat{\boldsymbol{y}}_\Omega$ is a sufficient condition for L_Ω having a margin (Blondel et al., 2020, Proposition 6). By leveraging these established facts from prior propositions, we will prove that our class of Hopfield networks is capable of exact retrieval.

4.4 Sparse Hopfield networks: Definition and update rule

In this section, we study a specialization of the HFY energy defined in §3 by assuming that the regularizer Ω has domain $\text{dom}(\Omega) = \triangle_N$ and that it is a generalized negentropy (see Definition 4). We assume also that $\Psi(\boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{q}\|^2$, and we use $\boldsymbol{u} = \frac{1}{N}$ as the baseline. Using $\tilde{\Omega}(\boldsymbol{\theta}) = \Omega(\beta\boldsymbol{\theta})$, where $\beta^{-1} > 0$ is a temperature parameter, the HFY energy (6) becomes the following, up to an extra constant:

$$E(\boldsymbol{q}) = \underbrace{-\beta^{-1} L_\Omega(\beta \boldsymbol{X}\boldsymbol{q}; \boldsymbol{1}/N)}_{E_{\text{concave}}(\boldsymbol{q})} + \underbrace{\frac{1}{2}\|\boldsymbol{q} - \boldsymbol{\mu_X}\|^2 + \frac{1}{2}(M^2 - \|\boldsymbol{\mu_X}\|^2)}_{E_{\text{convex}}(\boldsymbol{q})}, \tag{20}$$
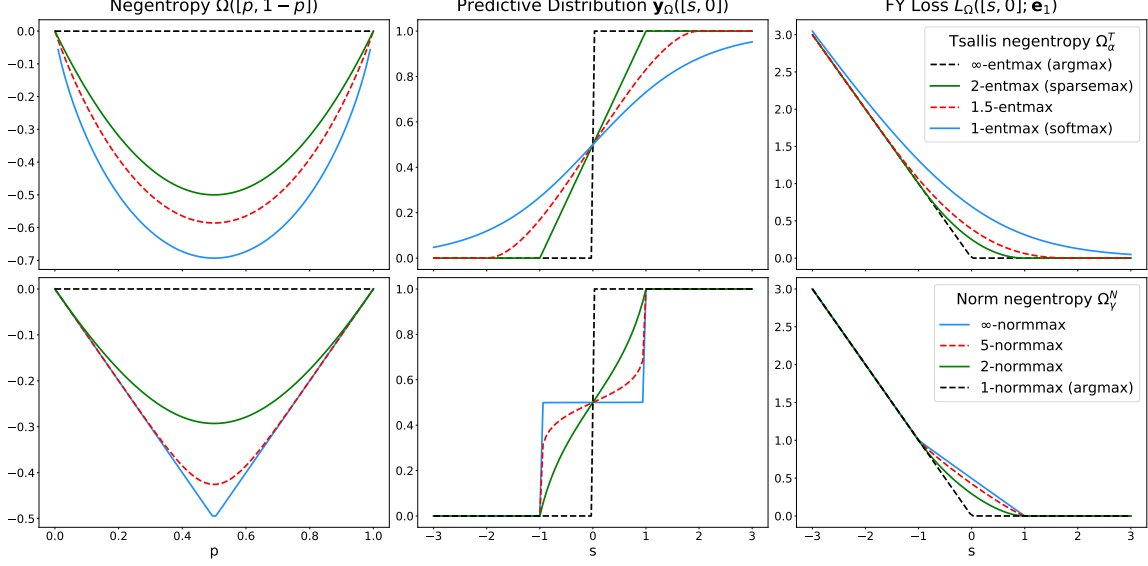
14

Figure 3: Examples of **generalized entropies** (left) are presented alongside their corresponding **prediction distributions** (middle) and **Fenchel-Young losses** (right) for the binary case. Here, $\boldsymbol{y} = [p, 1-p] \in \triangle_2$, $\boldsymbol{\theta} = [s, 0] \in \mathbb{R}^2$, and $\boldsymbol{e}_1$ is the one-hot vector for the first class. Unlike softmax, which never reaches exactly zero and consequently does not have a margin, all other distributions shown in the center can exhibit sparse support.

where $\boldsymbol{\mu_X} := \boldsymbol{X}^\top \boldsymbol{1}/N \in \mathbb{R}^D$ is the empirical mean of the patterns and $M := \max_i \|\boldsymbol{x}_i\|$. This energy extends that of Ramsauer et al. (2021) in (2), which is recovered when Ω is Shannon's negentropy. The E_{convex} and E_{concave} terms compete when minimizing (20):

- Minimizing E_{concave} is equivalent to *maximizing* $L_\Omega(\beta \boldsymbol{X} \boldsymbol{q}; \boldsymbol{1}/N)$, which pushes $\boldsymbol{q}$ to be as far as possible from a uniform average and closer to a single pattern.

- Minimizing E_{convex} serves as proximity regularization, encouraging $\boldsymbol{q}$ to stay close to $\boldsymbol{\mu_X}$.

The next result is a consequence of our Proposition 2, generalizing Ramsauer et al. (2021, Lemma A.1, Theorem A.1). The bounds on the energy are derived in Appendix B.1.

Proposition 7 (Update rule of sparse HFY energies) *Let the query $\boldsymbol{q}$ be in the convex hull of the rows of $\boldsymbol{X}$, i.e., $\boldsymbol{q} = \boldsymbol{X}^\top \boldsymbol{y}$ for some $\boldsymbol{y} \in \triangle_N$. Then, the energy (20) satisfies $0 \le E(\boldsymbol{q}) \le \min\left\{2M^2, \; -\beta^{-1}\Omega(\boldsymbol{1}/N) + \frac{1}{2}M^2\right\}$. Furthermore, minimizing (20) with the CCCP algorithm (Yuille and Rangarajan, 2003) leads to the updates*

$$\boldsymbol{q}^{(t+1)} = \boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q}^{(t)}). \tag{21}$$

Entmax and normmax. When $\Omega = \Omega_\alpha^T$ (the Tsallis α-negentropy (17)), the update (21) corresponds to the adaptively sparse transformer of Correia et al. (2019). The α-entmax

15

transformation can be computed efficiently with sort or top-k algorithms for $\alpha \in \{1.5, 2\}$; for other values of α, an efficient bisection algorithm was proposed by Peters et al. (2019). The case $\alpha = 2$ (sparsemax) corresponds to the sparse modern Hopfield network proposed by Hu et al. (2023). When $\alpha > 1$ and $\alpha \neq 2$, we recover the α-entmax Hopfield network introduced by Martins et al. (2023), Wu et al. (2024b), and Santos et al. (2024). When $\Omega = \Omega_\gamma^N$ (the norm γ-negentropy (18)), we obtain the γ-normmax transformation. This transformation is more challenging since Ω_γ^N is not separable, but Appendix A presents a bisection algorithm which works for any $\gamma > 1$.

ℓ_2-normalization and layer normalization. It is possible to extend the idea above to incorporate a post-transformation as described in §3.3.6. We discuss this scenario in §4.6.

4.5 Properties: Margins, sparsity, and exact retrieval

Prior work on modern Hopfield networks (Ramsauer et al., 2021, Def. 1) defines pattern storage and retrieval in an *approximate* sense: they assume a small neighbourhood around each pattern $\boldsymbol{x}_i$ containing an attractor $\boldsymbol{x}_i^*$, such that if the initial query $\boldsymbol{q}^{(0)}$ is close enough, the Hopfield updates will converge to $\boldsymbol{x}_i^*$, leading to a retrieval error of $\|\boldsymbol{x}_i^* - \boldsymbol{x}_i\|$. For this error to be small, a large β may be necessary. We consider here a stronger definition of **exact retrieval**, where the attractors *coincide* with the actual patterns (rather than being nearby). Our main result is that **zero retrieval error** is possible in HFY networks as long as the corresponding Fenchel-Young loss has a **margin** (Def. 5). Given that $\hat{\boldsymbol{y}}_\Omega$ being a sparse transformation is a sufficient condition for L_Ω having a margin (Blondel et al., 2020, Proposition 6), this is a general statement about sparse transformations.[5]

> **Definition 8 (Exact retrieval)** *A pattern $\boldsymbol{x}_i$ is **exactly retrieved** for query $\boldsymbol{q}^{(0)}$ iff there is a finite number of steps T such that iterating (21) leads to $\boldsymbol{q}^{(T')} = \boldsymbol{x}_i \ \forall T' \geq T$.*

The following result gives sufficient conditions for exact retrieval with $T = 1$ given that patterns are well separated and that the query is sufficiently close to the retrieved pattern. It establishes the **exact autoassociative** property of this class of HFY networks: if all patterns are slightly perturbed, the Hopfield dynamics are able to recover the original patterns exactly. Following Ramsauer et al. (2021, Def. 2), we define the separation of pattern $\boldsymbol{x}_i$ from data as $\Delta_i = \boldsymbol{x}_i^\top \boldsymbol{x}_i - \max_{j \neq i} \boldsymbol{x}_i^\top \boldsymbol{x}_j$.

> **Proposition 9 (Exact retrieval in a single iteration)** *Assume L_Ω has margin m, and let $\boldsymbol{x}_i$ be a pattern outside the convex hull of the other patterns. Then*
>
> *1. $\boldsymbol{x}_i$ is a stationary point of the energy (20) iff $\Delta_i \geq m\beta^{-1}$.*
>
> *2. In addition, if the initial query $\boldsymbol{q}^{(0)}$ satisfies $\boldsymbol{q}^{(0)^\top}(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq m\beta^{-1}$ for all $j \neq i$, then the update rule (21) converges to $\boldsymbol{x}_i$ exactly in one iteration.*

---

5. At first sight, this might seem to be a surprising result, given that both queries and patterns are continuous. The reason why exact convergence is possible hinges crucially on sparsity.

3. *Moreover, if the patterns are normalized, $\|\boldsymbol{x}_i\| = M$ for all i, and well-separated with $\Delta_i \geq m\beta^{-1} + 2M\epsilon$, then any $\boldsymbol{q}^{(0)}$ ϵ-close to $\boldsymbol{x}_i$ ($\|\boldsymbol{q}^{(0)} - \boldsymbol{x}_i\| \leq \epsilon$) will converge to $\boldsymbol{x}_i$ in one iteration.*

The proof is in Appendix B.4. For the Tsallis negentropy case $\Omega = \Omega_\alpha^T$ with $\alpha > 1$ (the sparse case), we have $m = (\alpha - 1)^{-1}$ (cf. Def. 5), leading to the condition $\Delta_i \geq \frac{1}{(\alpha-1)\beta}$. This result is stronger than that of Ramsauer et al. (2021) for their energy (which is ours for $\alpha = 1$), according to which memory patterns are only ϵ-close to stationary points, where a small $\epsilon = \mathcal{O}(\exp(-\beta))$ requires a low temperature (large β). It is also stronger than the retrieval error bound recently derived by Hu et al. (2023, Theorem 2.2) for the case $\alpha = 2$, which has an additive term involving M and therefore does not provide conditions for exact retrieval. For the normmax negentropy case $\Omega = \Omega_\gamma^N$ with $\gamma > 1$, we have $m = 1$, so the condition above becomes $\Delta_i \geq \frac{1}{\beta}$.

Given that exact retrieval is a stricter definition, one may wonder whether requiring it sacrifices storage capacity. Reassuringly, the next result, inspired but stronger than (Ramsauer et al., 2021, Theorem A.3)[6] and proved in our Appendix B.5, shows that HFY networks with exact retrieval also have exponential storage capacity.

Proposition 10 (Storage capacity with exact retrieval) *Assume patterns are optimally placed on the sphere of radius M. The HFY network can store and exactly retrieve $N = \mathcal{O}((2/\sqrt{3})^D)$ patterns in one iteration under an ϵ-perturbation as long as $M^2 > 2m\beta^{-1}$ and $\epsilon \leq \frac{M}{4} - \frac{m}{2\beta M}$.*

Assume patterns are randomly placed on the sphere with uniform distribution. Then, with probability $1 - p$, the HFY network can store and exactly retrieve $N = \mathcal{O}(\sqrt{p}\zeta^{\frac{D-1}{2}})$ patterns in one iteration under a ϵ-perturbation if

$$\epsilon \leq \frac{M}{2}\left(1 - \cos\frac{1}{\zeta}\right) - \frac{m}{2\beta M}. \tag{22}$$

4.6 Extension of previous results for extra normalization step

We now extend the previous results to the scenario where a post-transformation $\hat{\boldsymbol{y}}_\Psi$ is applied, such as ℓ_2-normalization or layer normalization, as described in §3.3.6. In this scenario, the update rule (21) is replaced by

$$\boldsymbol{q}^{(t+1)} = \hat{\boldsymbol{y}}_\Psi\left(\boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\beta\boldsymbol{X}\boldsymbol{q}^{(t)})\right). \tag{23}$$

We consider the image set induced by this transformation $\mathrm{im}(\hat{\boldsymbol{y}}_\Psi) := \{\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) : \boldsymbol{z} \in \mathbb{R}^D\}$. For ℓ_2-normalization, $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \frac{r\boldsymbol{z}}{\|\boldsymbol{z}\|}$, this image set is a $(D-1)^{\text{th}}$ sphere of radius r, and for layer normalization, $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \mathrm{LayerNorm}(\boldsymbol{z}; \eta, \boldsymbol{\delta})$ (14), it is the $(D-2)^{\text{th}}$-dimensional set $\{\boldsymbol{q} : \|\boldsymbol{q} - \boldsymbol{\delta}\| = \eta\sqrt{D} \wedge \mathbf{1}^\top(\boldsymbol{q} - \boldsymbol{\delta}) = 0\}$.

---

6. In fact, there is a mistake in the proof of Theorem A.3 by Ramsauer et al. (2021) regarding their bound on "equidistant patterns on the sphere". We fix their proof in Appendix B.5 by making a connection with existing bounds in sphere packing and kissing numbers, an active problem in combinatorics (Chabauty, 1953; Conway and Sloane, 2013).

Proposition 11 *Assume Ψ is chosen so that the transformation $\hat{\boldsymbol{y}}_\Psi$ is idempotent, i.e., $\hat{\boldsymbol{y}}_\Psi(\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z})) = \hat{\boldsymbol{y}}_\Psi(\boldsymbol{z})$ for all $\boldsymbol{z} \in \mathbb{R}^D$. Then, if all patterns $\boldsymbol{x}_i$ satisfy $\boldsymbol{x}_i \in \mathrm{im}(\hat{\boldsymbol{y}}_\Psi)$, we have that all results in Propositions 9–10, which concern convergence in one iteration, also hold for the Hopfield updates (23).*

Proof Propositions 9–10 guarantee that $\boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q}^{(0)}) = \boldsymbol{x}_i$ for some $i \in [N]$; since $\boldsymbol{x}_i \in \mathrm{im}(\hat{\boldsymbol{y}}_\Psi)$ and $\hat{\boldsymbol{y}}_\Psi$ is idempotent, the subsequent post-transformation in (23) will not change the result, ensuring that $\boldsymbol{q}^{(t+1)} = \boldsymbol{x}_i$. ∎

The idempotence condition is satisfied for both the ℓ_2-normalization and layer normalization transformations mentioned above. The condition $\boldsymbol{x}_i \in \mathrm{im}(\hat{\boldsymbol{y}}_\Psi)$ is satisfied if the patterns in $\boldsymbol{X}$ are pre-normalized with the same post-transformation $\hat{\boldsymbol{y}}_\Psi$ that is applied to the queries. The conditions in Propositions 9–10 which require $\|\boldsymbol{x}_i\| = M$ are satisfied, in the ℓ_2-normalization case, by $r = M$, and in the layer normalization case by $\eta = \frac{M}{\sqrt{D}}$ and $\boldsymbol{\delta} = \boldsymbol{0}$.

Even in situations where the initial query $\boldsymbol{q}^{(0)}$ is not sufficiently close to a pattern to obtain convergence in one step, the inclusion of a post-transformation step under the conditions of Proposition 11 can speed up convergence by projecting the query to a smaller subspace $\mathrm{im}(\hat{\boldsymbol{y}}_\Psi)$ where the patterns are contained. This will be illustrated in §7.2.

5. Structured Hopfield Networks

In §4, we considered the case where $\boldsymbol{y} \in \mathrm{dom}(\Omega) = \triangle_N$, the scenario studied by Ramsauer et al. (2021) and Hu et al. (2023). Since $\triangle_N = \mathrm{conv}(\mathcal{Y})$ with $\mathcal{Y} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\}$, we can see the domain of Ω as the convex relaxation of the set $\mathcal{Y}$ of pattern indicators.

We now go one step further and consider the more general **structured** case, where $\mathrm{dom}(\Omega)$ is a polytope. More specifically, we assume that $\boldsymbol{y} \in \mathrm{dom}(\Omega) := \mathrm{conv}(\mathcal{Y})$ is a vector of "marginals" associated to some given structured set $\mathcal{Y}$. This structure can reflect **pattern associations** that we might want to induce when querying the Hopfield network with $\boldsymbol{q}^{(0)}$. Possible structures include k**-subsets** of memory patterns, potentially leveraging **sequential memory structure**, tree structures, matchings, etc. In these cases, the set of pattern associations we can form is combinatorial, hence it can be considerably larger than the number N of memory patterns.

5.1 Unary scores and structured constraints

As before, we assume N is the number of patterns stored in the memory. Let us start with a simple scenario where there is a predefined set of binary structures $\mathcal{Y} \subseteq \{0, 1\}^N$ and N unary scores $\boldsymbol{\theta} \in \mathbb{R}^N$, one for each memory pattern. We assume that we may have $|\mathcal{Y}| \gg N$ in general. In what follows, we let $\mathrm{dom}(\Omega) = \mathrm{conv}(\mathcal{Y}) \subseteq [0, 1]^N$ denote its convex hull, called the **marginal polytope** associated with the structured set $\mathcal{Y}$ (Wainwright et al., 2008). Later, in §5.2, we generalize this framework to accommodate higher-order interactions modeling soft interactions among patterns.

Example 1 (k-subsets) *We may be interested in retrieving subsets of k patterns, e.g., to take into account a k-ary relation among patterns or to perform top-k retrieval. In this case, we can define, for $k \in [N]$,*

$$\mathcal{Y} := \left\{ \boldsymbol{y} \in \{0,1\}^N \; : \; \boldsymbol{1}^\top \boldsymbol{y} = k \right\}.$$

If $k = 1$, we get $\mathcal{Y} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\}$ and $\mathrm{conv}(\mathcal{Y}) = \triangle_N$, recovering the scenario in §4. For larger k, $|\mathcal{Y}| = \binom{N}{k} \gg N$. With a simple rescaling, the marginal polytope $\mathrm{conv}(\mathcal{Y})$ is equivalent to the capped probability simplex described by Blondel et al. (2020, §7.3).

Given unary scores $\boldsymbol{\theta} \in \mathbb{R}^N$, the structure with the largest score is

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{\theta}^\top \boldsymbol{y} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathrm{conv}(\mathcal{Y})} \boldsymbol{\theta}^\top \boldsymbol{y}, \tag{24}$$

where the last equality comes from the fact that $\mathrm{conv}(\mathcal{Y})$ is a polytope, therefore the maximum is attained at a vertex. The solution of (24) is often called the **maximum a posteriori (MAP)** assignment. As in (4), we consider a regularized prediction version of this problem via a convex regularizer $\Omega : \mathrm{conv}(\mathcal{Y}) \to \mathbb{R}$:

$$\hat{\boldsymbol{y}}_\Omega(\boldsymbol{\theta}) := \operatorname*{argmax}_{\boldsymbol{y} \in \mathrm{conv}(\mathcal{Y})} \boldsymbol{\theta}^\top \boldsymbol{y} - \Omega(\boldsymbol{y}). \tag{25}$$

By choosing $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}\|^2 + I_{\mathrm{conv}(\mathcal{Y})}(\boldsymbol{y})$, we obtain **SparseMAP**, which can be seen as a relaxation of MAP and a structured version of sparsemax (Niculae et al., 2018). The SparseMAP transformation (25) can be computed efficiently via an active set algorithm, as long as an algorithm is available to compute the MAP in (24), as shown in Niculae et al. (2018, §3.2) We will make use of this efficient algorithm in our experiments in §7.

5.2 General case: factor graph, high order interactions

We consider now the more general case where there might be soft interactions among patterns, for example due to temporal dependencies, hierarchical structure, etc. In general, these interactions can be expressed as a bipartite **factor graph** (V, F), where $V = \{1, ..., N\}$ are variable nodes (associated with the patterns) and $F \subseteq 2^V$ are factor nodes representing the interactions (Kschischang et al., 2001).

A structure can be represented as a bit vector $\boldsymbol{y} = [\boldsymbol{y}_V; \boldsymbol{y}_F]$, where $\boldsymbol{y}_V$ and $\boldsymbol{y}_F$ indicate configurations of variable and factor nodes, respectively. Each factor $f \in F$ is linked to a subset of variable nodes $V_f \subseteq V$. We assume each variable $v \in V$ can take one of N_v possible values, and we denote by $\boldsymbol{y}_v \in \{0,1\}^{N_v}$ a one-hot vector indicating a value for this variable. Likewise, each factor $f \in F$ has N_f possible configurations, with $N_f = \prod_{v \in V_f} N_v$, and we associate to it a one-hot vector $\boldsymbol{y}_f \in \{0,1\}^{N_f}$ indicating a configuration for that factor. The global configuration of the factor graph is expressed through the bit vectors $\boldsymbol{y}_V = [\boldsymbol{y}_v \; : \; v \in V] \in \{0,1\}^{N_V}$ and $\boldsymbol{y}_F = [\boldsymbol{y}_f \; : \; f \in F] \in \{0,1\}^{N_F}$, with $N_V = \sum_{v \in V} N_v$ and $N_F = \sum_{f \in F} N_f$. A particular structure is expressed through the bit vector $\boldsymbol{y} = [\boldsymbol{y}_V; \boldsymbol{y}_F] \in \{0,1\}^{N_V + N_F}$. Finally, we define the set of **valid structures** $\mathcal{Y} \subseteq \{0,1\}^{N_V + N_F}$—this set contains all the bit vectors which correspond to valid structures,

which must satisfy consistency between variable and factor assignments, as well as any additional hard constraints.

We associate **unary scores** $\boldsymbol{\theta}_V = [\boldsymbol{\theta}_v : v \in V] \in \mathbb{R}^{N_V}$ to configurations of variable nodes and **higher-order scores** $\boldsymbol{\theta}_F = [\boldsymbol{\theta}_f : f \in F] \in \mathbb{R}^{N_F}$ to configurations of factor nodes. We denote $\boldsymbol{\theta} = [\boldsymbol{\theta}_V; \boldsymbol{\theta}_F] \in \mathbb{R}^{N_V + N_F}$. The MAP inference problem is exactly as in (24) where the objective can be written as $\boldsymbol{\theta}^\top \boldsymbol{y} = \boldsymbol{\theta}_V^\top \boldsymbol{y}_V + \boldsymbol{\theta}_F^\top \boldsymbol{y}_F$. As above, we consider regularized variants of (24) via a convex regularizer $\Omega : \mathrm{conv}(\mathcal{Y}) \to \mathbb{R}$. **SparseMAP** corresponds to $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}_V\|^2 + I_{\mathrm{conv}(\mathcal{Y})}(\boldsymbol{y})$ (note that only the unary variables are quadratically regularized), which leads to the problem (25). The active set algorithm of Niculae et al. (2018) applies also to this general case, requiring only a MAP oracle to solve (24).

Example 2 (sequential k-subsets) *Consider the k-subset problem of Example 1 but now with a sequential structure. This can be represented as a pairwise factor graph (V, F) where $V = \{1, ..., N\}$ and $F = \{(i, i+1)\}_{i=1}^{N-1}$. The budget constraint forces exactly k of the N variable nodes to have the value 1. The set $\mathcal{Y}$ contains all bit vectors satisfying these constraints as well as consistency among the variable and factor assignments.*

- *To each $i \in V$ we assign unary "emission" scores $\boldsymbol{\theta}_i = [0, s_i] \in \mathbb{R}^2$, corresponding to the states "off" and "on", respectively. A positive s_i expresses a preference for the state "on" and a negative s_i for the state "off".*

- *To each factor (edge) $(i, i+1) \in F$ we associate Ising higher-order (pairwise) "transition" scores $\boldsymbol{\theta}_{(i,i+1)} = [0, 0, 0, t] \in \mathbb{R}^4$, corresponding to state pairs "off-off", "off-on", "on-off", and "on-on", respectively. To promote consecutive memory items to be both or neither retrieved, we can define attractive "transition" scores by choosing $t > 0$.*

The MAP inference problem for this model can be solved with dynamic programming in runtime $\mathcal{O}(Nk)$, and the SparseMAP transformation can be computed with the active set algorithm (Niculae et al., 2018) by iteratively calling this MAP oracle.

5.3 Structured Fenchel-Young losses and margins

Fenchel-Young losses are applicable to the structured case outlined in this section by choosing a regularizer with domain $\mathrm{dom}(\Omega) = \mathrm{conv}(\mathcal{Y})$, instead of the probability simplex $\triangle_N$. In the sequel, we focus on the SparseMAP regularizer $\Omega(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{y}_V\|^2 + I_{\mathrm{conv}(\mathcal{Y})}(\boldsymbol{y})$. The notion of margin in Def. 5 can be extended to the structured case (Blondel et al., 2020, Def. 5):

Definition 12 (Structured margin) *A loss $L(\boldsymbol{\theta}; \boldsymbol{y})$ has a **structured margin** if $\exists 0 \le m < \infty$ such that $\forall \boldsymbol{y} \in \mathcal{Y}$:*

$$\boldsymbol{\theta}^\top \boldsymbol{y} \ge \max_{\boldsymbol{y}' \in \mathcal{Y}} \left(\boldsymbol{\theta}^\top \boldsymbol{y}' + \frac{m}{2}\|\boldsymbol{y} - \boldsymbol{y}'\|^2 \right) \;\Rightarrow\; L(\boldsymbol{\theta}; \boldsymbol{y}) = 0.$$

The smallest such m is called the margin of L.

Note that the definition of of margin in Def. 5 is recovered when $\mathcal{Y} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\}$. Note also that, since we are assuming $\mathcal{Y} \subseteq \{0, 1\}^{N_V + N_F}$, the term $\|\boldsymbol{y} - \boldsymbol{y}'\|^2$ is a Hamming distance, which counts how many bits need to be flipped to transform $\boldsymbol{y}'$ into $\boldsymbol{y}$. A well-known example

of a loss with a structured separation margin is the structured hinge loss (Taskar et al., 2003; Tsochantaridis et al., 2005).

We show below that the SparseMAP loss has a structured margin (our result, proved in Appendix B.6, extends that of Blondel et al. (2020), who have shown this only for structures without high order interactions):

Proposition 13 *Let $\mathcal{Y} \subseteq \{0,1\}^{N_V + N_F}$ be contained in a sphere, i.e., for some $r > 0$, $\|\boldsymbol{y}\| = r$ for all $\boldsymbol{y} \in \mathcal{Y}$. Then:*

1. *Without high order interactions, the SparseMAP loss has a structured margin $m = 1$.*

2. *If there are high order interactions and, for some r_V and r_F with $r_V^2 + r_F^2 = r^2$, we have $\|\boldsymbol{y}_V\| = r_V$ and $\|\boldsymbol{y}_F\| = r_F$ for any $\boldsymbol{y} = [\boldsymbol{y}_V; \boldsymbol{y}_F] \in \mathcal{Y}$, then the SparseMAP loss has a structured margin $m \leq 1$.*

The assumptions above are automatically satisfied with the factor graph construction in §5.2, with $r_V^2 = |V|$, $r_F^2 = |F|$, and $r^2 = |V| + |F|$. For the k-subsets example, we have $r^2 = k$, and for the sequential k-subsets example, we have $r_V^2 = N$, $r_F^2 = N - 1$, and $r^2 = 2N - 1$.

5.4 Guarantees for retrieval of pattern associations

We now consider a **structured HFY network** using SparseMAP. Following the same logic as Propositions 2 and 7, we obtain the following update rule:

$$\boldsymbol{q}^{(t+1)} = \boldsymbol{X}^\top \text{SparseMAP}(\beta \boldsymbol{X} \boldsymbol{q}^{(t)}). \tag{26}$$

In this structured case, we aim to retrieve not individual patterns but pattern associations of the form $\boldsymbol{X}^\top \boldsymbol{y}$, where $\boldsymbol{y} \in \mathcal{Y}$. Naturally, when $\mathcal{Y} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\}$, we recover the usual patterns, since $\boldsymbol{x}_i = \boldsymbol{X}^\top \boldsymbol{e}_i$. We define the separation of pattern association $\boldsymbol{y}_i \in \mathcal{Y}$ from data as $\Delta_i = \boldsymbol{y}_i^\top \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{y}_i - \max_{j \neq i} \boldsymbol{y}_i^\top \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{y}_j$. The next proposition, proved in Appendix B.7, states conditions for exact convergence in a single iteration, generalizing Proposition 9.

Proposition 14 (Exact structured retrieval) *Let $\Omega(\boldsymbol{y})$ be the SparseMAP regularizer and assume the conditions of Proposition 13 hold. Let $\boldsymbol{y}_i \in \mathcal{Y}$ be such that $\Delta_i \geq \frac{D_i^2}{2\beta}$, where $D_i = \max \|\boldsymbol{y}_i - \boldsymbol{y}_j\| \leq 2r$. Then, $\boldsymbol{X}^\top \boldsymbol{y}_i$ is a stationary point of the Hopfield energy. In addition, if $\boldsymbol{q}^{(0)\top} \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \geq \frac{D_i^2}{2\beta}$ for all $j \neq i$, then the update rule (26) converges to the pattern association $\boldsymbol{X}^\top \boldsymbol{y}_i$ in one iteration. Moreover, if*

$$\Delta_i \geq \frac{D_i^2}{2\beta} + \epsilon \min\{\sigma_{\max}(\boldsymbol{X}) D_i, M D_i^2\},$$

where $\sigma_{\max}(\boldsymbol{X})$ is the spectral norm of $\boldsymbol{X}$ and $M = \max_k \|\boldsymbol{x}_k\|$, then any $\boldsymbol{q}^{(0)}$ ϵ-close to $\boldsymbol{X}^\top \boldsymbol{y}_i$ will converge to $\boldsymbol{X}^\top \boldsymbol{y}_i$ in one iteration.

Note that the bound above on Δ_i includes as a particular case the unstructured bound in Proposition 9 applied to sparsemax (entmax with $\alpha = 2$, which has margin $m = 1/(\alpha - 1) = 1$), since for $\mathcal{Y} = \triangle_N$ we have $r = 1$ and $D_i = \sqrt{2}$, which leads to the condition $\Delta_i \geq \beta^{-1} + 2M\epsilon$.

For the particular case of the k-subsets problem (Example 1), we have $r = \sqrt{k}$ and $D_i = \sqrt{2k}$, leading to the condition $\Delta_i \geq \frac{k}{\beta} + 2Mk\epsilon$. This recovers sparsemax when $k = 1$.

For the sequential k-subsets problem in Example 2, we have $r = 2N - 1$. Noting that any two distinct $\boldsymbol{y}$ and $\boldsymbol{y}'$ differ in at most $2k$ variable nodes, and since each variable node can affect 6 bits (2 for $\boldsymbol{y}_V$ and 4 for $\boldsymbol{y}_F$), the Hamming distance between $\boldsymbol{y}$ and $\boldsymbol{y}'$ is at most $12k$, therefore we have $D_i = \sqrt{12k}$, which leads to the condition $\Delta_i \geq \frac{6k}{\beta} + 12Mk\epsilon$.

6. Mechanics of Memory Retrieval Modeling

Memory is both a foundational paradigm in human cognitive psychology and a core focus of systems neuroscience in animal models (Eichenbaum, 2017). Ongoing research aims to integrate these levels of investigation into a comprehensive account of memory processing in the brain. From this integrative perspective, we describe how the framework described in §4 and §5 provides a theoretical platform for generatively modeling memory retrieval while retaining formal guarantees regarding memory capacity and convergence. We examine these models in the context of recall paradigms used to study human memory.

Recall tasks for humans are crucial in elucidating the structure of memory and retrieval processes (Tulving, 1972). The most basic paradigm, associative recall, involves the learning of paired items, where one item serves as a cue to retrieve the associated item which is typically a corrupted or partial version of the target item. **Sequential recall** requires learning a sequence of items in a specific order. During retrieval, the initial item serves as a cue, and the individual must recall the subsequent items in the exact order of presentation. **Free recall** involves the learning of a list of items presented without a specific sequence. During retrieval, individuals use a contextual cue to recall the items in any order, though all items must eventually be retrieved. In computational neuroscience, models have been proposed with varying degrees of biological precision which suggest that auto-associative and hetero-associative attractor dynamics in the hippocampal formation subserve memory recall in these paradigms (Naim et al., 2020; Boboeva et al., 2021). However, such models operate in a simplified setting focusing on binary, orthogonalized, and relatively low-dimensional memory patterns without broader theoretic guarantees for memories of the scale and complexity encoded by humans.

In order to address this gap, and inspired by this computational cognitive neuroscience approach for investigating memory recall, we apply our framework in deriving efficient algorithms for modeling memory retrieval with the latter two specified paradigms.

6.1 Free recall

Consider the sparsemax transformation presented in §2:

$$\text{sparsemax}(\boldsymbol{\theta}) := \underset{\boldsymbol{y} \in \triangle_N}{\text{argmax}} \, \boldsymbol{\theta}^\top \boldsymbol{y} - \frac{1}{2}\|\boldsymbol{y}\|^2. \tag{27}$$

These projections often reach the boundary of the simplex, resulting in a sparse probability distribution. However, they are not well-suited for modeling recall paradigms like free recall, as these models primarily retrieve the memory closest to the query without keeping a record of previously attended memories. This behaviour leads to potential repetitions and failure to

---

Algorithm 1 Free recall with constrained sparsemax. $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ represents the memory and $\boldsymbol{q} \in \mathbb{R}^D$ denotes the query, initialized as an arbitrary cue. T is the number of inner associative Hopfield iterations. N is the number of memory patterns, which equals the number of outer iterations during the recall process.

Require: $\boldsymbol{X}$, $\boldsymbol{q}$, β, T

1: $\boldsymbol{u} \leftarrow \mathbf{1}_N$ ▷ Initialize upper bounds
2: **for** $i \leftarrow 1$ to N **do**
3:      **for** $j \leftarrow 1$ to T **do**
4:          $\boldsymbol{\theta} \leftarrow \boldsymbol{X}\boldsymbol{q}$ ▷ Scores
5:          $\boldsymbol{p} \leftarrow \text{csparsemax}(\beta\boldsymbol{\theta}; \boldsymbol{u})$
6:          $\boldsymbol{q} \leftarrow \boldsymbol{X}^\top \boldsymbol{p}$ ▷ Hopfield update
7:      $\boldsymbol{u} \leftarrow \boldsymbol{u} - \boldsymbol{p}$ ▷ Upper bound probabilities

---

Algorithm 2 Free recall with penalized α-entmax. The parameter λ corresponds to the penalty applied to the moving average while τ corresponds to the decay rate.

Require: $\boldsymbol{X}$, $\boldsymbol{q}$, λ, τ, β, T

1: $\boldsymbol{a} \leftarrow \mathbf{0}_N$ ▷ Initial average
2: **for** $i \leftarrow 1$ to N **do**
3:      $\boldsymbol{p} \leftarrow \alpha\text{-entmax}\left(\beta\left(\boldsymbol{X}\boldsymbol{q} - \lambda\boldsymbol{a}\right)\right)$ ▷ Penalized probabilities
4:      $\boldsymbol{a} \leftarrow \tau\boldsymbol{p} + (1-\tau)\boldsymbol{a}$ ▷ Exponentially weighted average
5:      $\boldsymbol{q} \leftarrow \boldsymbol{X}^\top \boldsymbol{p}$ ▷ Outer Hopfield update
6:      **for** $j \leftarrow 1$ to T **do**
7:          $\boldsymbol{p} \leftarrow \alpha\text{-entmax}(\beta\boldsymbol{X}\boldsymbol{q})$
8:          $\boldsymbol{q} \leftarrow \boldsymbol{X}^\top \boldsymbol{p}$ ▷ Inner Hopfield update

---

attend to all distinct memories. One way to address this issue is to set an upper bound on the maximum probability which can be assigned to memories have already been attended to.

In this work, inspired by the concept of **constrained sparsemax** (Malaviya et al., 2018), we show that a modified version of sparse HFY networks can be used for modeling the free recall memory paradigm. Formally, constrained sparsemax is defined as:

$$\text{csparsemax}(\boldsymbol{\theta}; \boldsymbol{u}) := \underset{\boldsymbol{y} \in \triangle_N}{\text{argmax}} \, \boldsymbol{\theta}^\top \boldsymbol{y} - \frac{1}{2}\|\boldsymbol{y}\|^2 \quad \text{s.t.} \quad \boldsymbol{y} \leq \boldsymbol{u}, \tag{28}$$

where $\boldsymbol{u} \in [0,1]^N$ is a vector of upper bounds. This transformation closely resembles the sparsemax function but introduces bounded probabilities defined by $\boldsymbol{u}$. Malaviya et al. (2018) developed efficient forward and backward propagation algorithms for this transformation, making it practical for various applications. It can be useful for tasks such as modeling the free recall memory paradigm with Hopfield models. This process, described in Algorithm 1, involves an inner loop of associative recall and an outer loop where the constrained sparsemax transformation is used for keeping track of the attended memories by upper-bounding how much probability mass can be given to patterns that have already been attended to.

An alternative option (Algorithm 2) is to introduce a penalty mechanism which reduces the scores of attended patterns in subsequent Hopfield iterations. This penalty mechanism discourages the selection of previously chosen memories, aiming to model the non-repetitive functionality of human memory processing during free recall. A potential penalty is the

---

Algorithm 3 Sequential recall using SparseMAP with sequential 2-subsets. $t > 0$ denotes the transition score and $\omega \geq 1$ is a coefficient which promotes sequential order by boosting the emission score of the last recalled pattern.

Require: $\boldsymbol{X}$, $\boldsymbol{q}$, λ, τ, ω, t, β, T

1: $\boldsymbol{a} \leftarrow \boldsymbol{0}_N$
2: **for** $i \leftarrow 1$ to N **do**
3:      $\boldsymbol{y} \leftarrow \text{SEQUENTIAL\_K\_SUBSETS} \left(\beta \left(\boldsymbol{X}\boldsymbol{q} - \lambda \boldsymbol{a} \right), k = 2, t \right)$ ▷ Sequential 2-subsets
4:      $\boldsymbol{q} \leftarrow \boldsymbol{X}^\top \boldsymbol{y} - \boldsymbol{q}$ ▷ Outer Hopfield update
5:      **for** $k \leftarrow 1$ to T **do**
6:          $\boldsymbol{p} \leftarrow \alpha\text{-entmax}(\boldsymbol{X}\boldsymbol{q})$ ▷ Inner Hopfield update
7:          $\boldsymbol{q} \leftarrow \boldsymbol{X}^\top \boldsymbol{p}$
8:      $\boldsymbol{a} \leftarrow \tau(\boldsymbol{y} - \omega \boldsymbol{p}) + (1 - \tau)\boldsymbol{a}$ ▷ Exponentially weighted average

---

exponentially weighted average, which induces forgetting by dynamically balancing the influence of current and past penalties, encouraging diversity in selections.

6.2 Sequential recall

We now consider the sequential recall paradigm, deriving an algorithm inspired by the penalized free recall approach (Algorithm 2), but which leverages the structured Hopfield networks presented in §5. We consider the sequential k-subsets model described in Example 2 with large transition scores (*i.e.*, choosing a large $t > 0$), so that we strongly encourage the retrieval of consecutive memory patterns. This structured transformation is used in the outer loop, operating with $k = 2$, which promotes sequential top-2 retrieval. This encourages retrieving a pattern association involving two adjacent memory patterns, the cue (associated with the initial query) and the succeeding pattern.

The algorithm is presented as Algorithm 3. At each step in the outer loop, the structured Hopfield network with the sequential k-subsets model is first queried with $\boldsymbol{q}$ (which we would like to be close to some pattern, $\boldsymbol{q} \approx \boldsymbol{x}_i$) and returns a pattern association $\boldsymbol{y}$—ideally, this is a two-hot vector indicating the index of the cue pattern and the index of an adjacent pattern, e.g., $\boldsymbol{x}_{i+1}$; that is, $\boldsymbol{y} \approx \boldsymbol{x}_i + \boldsymbol{x}_{i+1}$. In reality it can be a fractional vector satisfying $\boldsymbol{1}^\top \boldsymbol{y} = 2$. In the ideal scenario we have $\boldsymbol{X}^\top \boldsymbol{y} \approx \boldsymbol{x}_i + \boldsymbol{x}_{i+1} \approx \boldsymbol{q} + \boldsymbol{x}_{i+1}$. (Note that in this structured Hopfield update we use a penalty similar to the one used in the penalized free recall algorithm, which we will come back to later.) Then, we subtract the query $\boldsymbol{q}$ from $\boldsymbol{X}^\top \boldsymbol{y}$—in the ideal scenario above, this should be close to $\boldsymbol{q} + \boldsymbol{x}_{i+1} - \boldsymbol{q} = \boldsymbol{x}_{i+1}$. This becomes the query to the inner Hopfield loop, which we expect to yield the attractor $\boldsymbol{x}_{i+1}$—the next pattern to be recalled, and the cue for the next step. The penalties are updated with the difference $\boldsymbol{y} - \omega \boldsymbol{p}$, where we expect $\boldsymbol{y} \approx \boldsymbol{e}_i + \boldsymbol{e}_{i+1}$ and $\boldsymbol{p} \approx \boldsymbol{e}_{i+1}$. If $\omega = 1$, this difference would be close to $\boldsymbol{e}_i$, the indicator of the ith pattern, which will be penalized in subsequent iterations to avoid being retrieved twice. By choosing ω slightly greater than one, we also add a small bonus to the $(i + 1)^{\text{th}}$ pattern, which we would like to be retrieved as part of the pattern association in the next step—this avoids memory jumps.
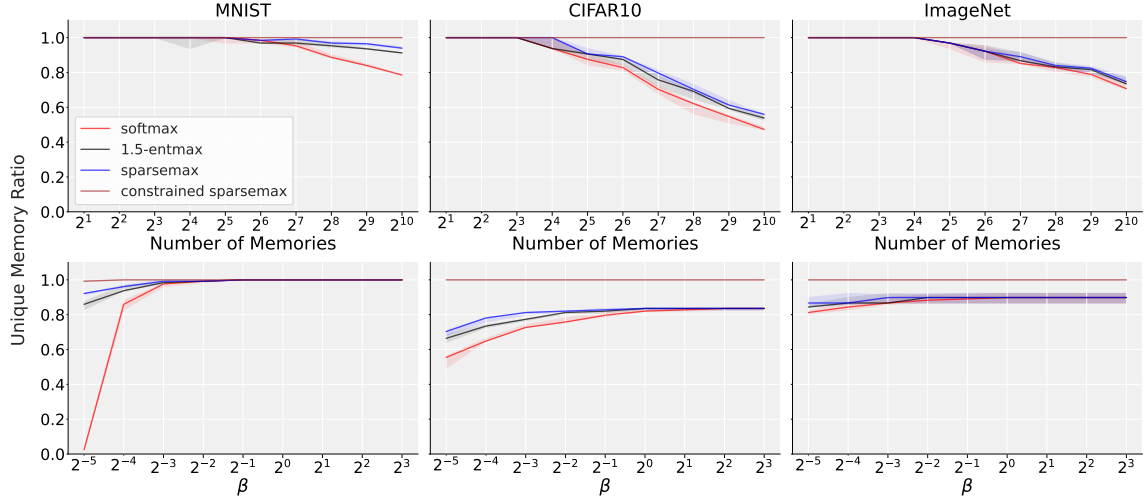
Figure 4: (Top) Memory capacity in terms of unique, non-repeated memories using various free recall methods for different numbers of stored memories with $\beta = 0.1$. (Bottom) Unique memory ratio as a function of β for a memory size of 128. Plotted are the medians over 5 runs with different memories and the interquartile range.

6.3 Empirical evaluation

We evaluate the algorithms derived in the current section using three datasets: MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky, 2009), and Tiny ImageNet (Le et al., 2015). We use a maximum of 20 Hopfield (or inner) iterations. For the penalized free and sequential recall, we employ a penalty of $\lambda = 10^9$ and a decay rate of $\tau = 0.001$. For the sequential recall algorithm, we use a transition score of 10^8 with the $k = 2$ for the sequential k-subsets and $\omega = 1.1$. The metric used to evaluate the algorithms is the unique memory ratio, which measures the proportion of distinct memories recalled. We illustrate the effectiveness of the constrained and penalized free recall methods in Figure 4. As expected, performance decreases as the number of stored memory items increases, with this effect being particularly noticeable for softmax due to its dense nature. Performance also improves with higher values of β, as the transformation becomes sparser. Despite this behavior, constrained sparsemax shows near-optimal performance across different numbers of memories and β values. The penalized α-entmax transformations, which are biologically more plausible, work effectively for smaller memory sizes, since they are able to forget the already attended memories through the penalty using the exponentially weighted average, but they degrade as the number of memories increases (with the sparse transformations performing better than softmax). Figure 5 which shows the perfect behavior of constrained sparsemax and the brief repetitiveness of penalized sparsemax. In Figure 6, a similar behavior is observed for the sequential recall paradigm in the first row, where the α-entmax methods show competitive performance, for $\alpha \in \{1, 1.5, 2\}$. In the second row, we evaluate the quality of the generated sequence by measuring the Levenshtein coefficient as a function of the number of memories.
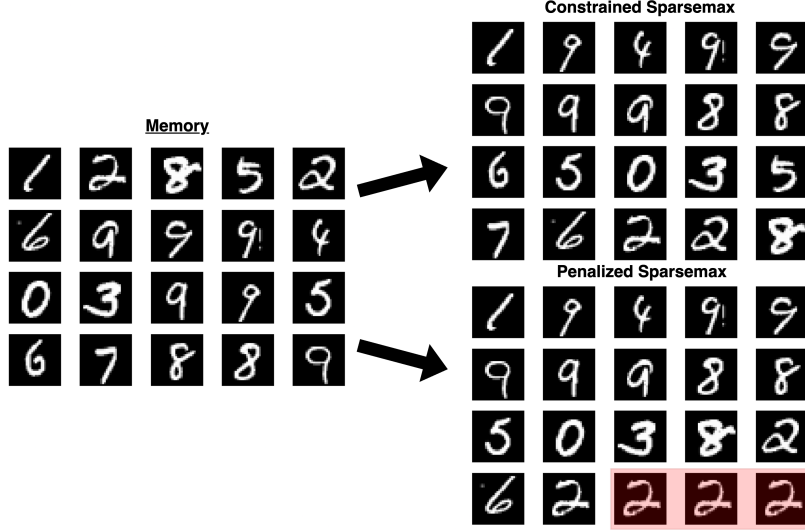
Figure 5: Simulation of **free recall** using our two methods on MNIST (LeCun et al., 1998): **constrained sparsemax** (Algorithm 1) and **penalized sparsemax** (Algorithm 2). For both methods, we set the number of Hopfield iterations to $T = 5$. In the penalized free recall method, we apply a penalty of $\lambda = 10^8$ and a decay rate of $\tau = 0.001$. In both case, we set $\beta = 0.1$. Red highlight corresponds to repeated memories.

This coefficient is computed as $1 - \frac{D}{C}$, where D is the Levenshtein distance and C the sequence length. Even with the inclusion of the parameter ω, the method still exhibits a tendency to jump between positions in memory, especially for larger memory sizes, which leads to the generation of multiple subsequences rather than reconstructing the full original sequence, as can be seen in Figure 7. Indeed, such "jumpy" dynamics are reminiscent of superdiffusive forms of hippocampal replay observed when animals thought to be reflective of parsimonious algorithms for sampling large memory structures (McNamee et al., 2021). This behavior results in fragmented outputs where the model captures and rearranges parts of the sequence as distinct units. Such "block" jumps, where the model effectively skips over certain parts of the sequence, are not adequately handled by the Levensthein distance and other known metrics. As expected, we observe a decrease in performance as the number of memories increases, despite empirical verification that the individual elements within the subsequence blocks are retrieved in the correct order. Nonetheless, softmax tends to be worse than the remaining methods, as expected.

7. Experiments

We now present experiments using both synthetic and real-world datasets to validate our theoretical findings in §4 and §5. These experiments demonstrate the practical benefits of our $\hat{\boldsymbol{y}}_\Omega$ functions, which result in sparse and structured Hopfield networks, and our $\hat{\boldsymbol{y}}_\Psi$ functions, which enable post-transformations like normalization and layer normalization.
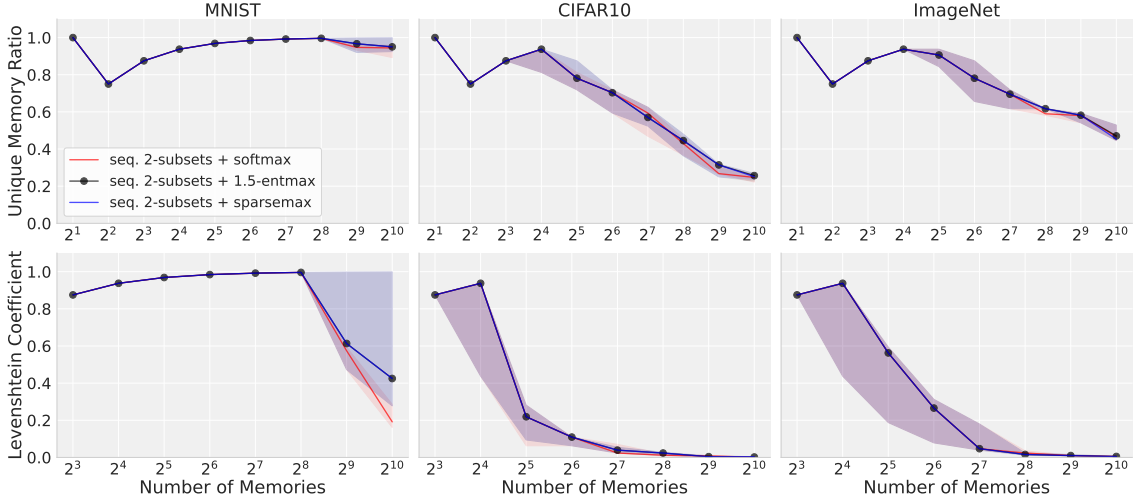
Figure 6: (Top) Memory capacity in terms of unique, non-repeated memories using the sequential recall for different numbers of stored memories with $\beta = 0.1$. (Bottom) Levenshtein coefficient as a function of number of memories. Plotted are the medians over 5 runs with different memories and the interquartile range.
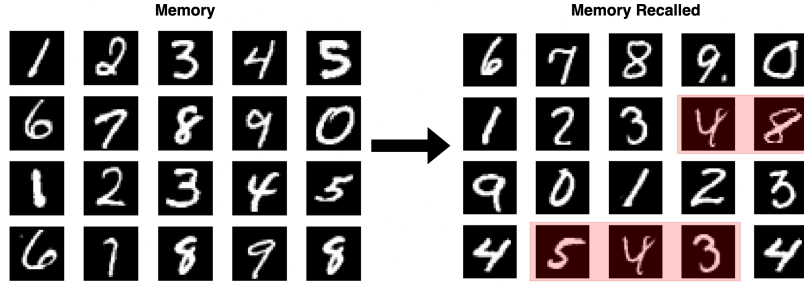

Figure 7: We illustrate **sequential recall** using Algorithm 3 on the MNIST dataset (LeCun et al., 1998) for sparsemax. We set the transition scores to $t = 10^5$, the number of inner iterations to $T = 100$, the penalty coefficient to $\lambda = 10^9$, and the decay rate to $\tau = 0.001$. We set $\beta = 0.1$ and $\omega = 1.1$. Red highlight corresponds to memory jumps.

7.1 Metastable state distributions in MNIST

We start by investigating how often our Hopfield networks converge to metastable states, an important aspect for understanding the network's dynamics. To elucidate this, we examine $\hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q}^{(t)})$ for the MNIST dataset (LeCun et al., 1998), probing the number of nonzeros in these vectors. We set a threshold > 0.01 for the softmax method (1-entmax). For the sparse transformations we do not need a threshold, since they have exact retrieval.

Table 3: Distribution of metastable state (in %) in MNIST. The training set is memorized and the test set is used as queries.

| Metastable State Size | $\beta = 0.1$ | | | | | | | | $\beta = 1$ | | | | | | | |
| | α-entmax | | | γ-normmax | | k-subsets | | | α-entmax | | | γ-normmax | | k-subsets | | |
	1	1.5	2	2	5	2	4	8	1	1.5	2	2	5	2	4	8
1	3.5	69.2	88.1	81.4	51.4	0.0	0.0	0.0	97.8	99.9	100.0	100.0	99.8	0.0	0.0	0.0
2	2.1	8.6	5.2	6.7	31.4	87.3	0.0	0.0	0.9	0.1	0.0	0.0	0.2	99.9	0.0	0.0
3	1.6	3.9	2.6	1.9	7.0	6.1	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.1	0.0	0.0
4	1.2	2.3	1.6	1.0	2.1	2.5	80.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	99.3	0.0
5	1.2	1.6	1.1	0.9	1.5	2.0	11.9	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.7	0.0
6	0.9	0.9	0.8	0.5	1.5	1.1	4.4	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0
7	1.1	0.6	0.4	0.4	1.3	0.6	2.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.8	0.6	0.1	0.8	1.0	0.2	1.0	60.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	95.0
9	1.0	0.3	0.0	0.5	0.8	0.1	0.4	26.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.7
10	1.1	0.1	0.0	0.5	0.6	0.0	0.1	9.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.2
10^+	85.5	11.9	0.1	5.4	1.4	0.1	0.0	4.8	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1

Results in Table 3 suggest that α-entmax is capable of retrieving single patterns for higher values of α. Despite γ-normmax's ability to induce sparsity, we observe that as γ increases, the method tends to stabilize in small but persistent metastable states. This behavior aligns with theoretical expectations, as it favors a uniform distribution over some patterns. On the other hand, SparseMAP with k-subsets is capable of retrieving sparse pattern associations of k patterns, as expected. For $\beta = 1$, we observe that all methods yield sparse distributions, which can be attributed to the inherently sparse nature of the MNIST dataset, where the majority of the pixels are background (represented by zeros), resulting in a high degree of sparsity in the data.

7.2 Hopfield dynamics and basins of attraction

Figures 8, 9 and 10 illustrate the optimization trajectories and basins of attraction across different queries and artificially generated memory pattern configurations for two families of sparse transformations: α-entmax and γ-normmax. We use the post-transformations $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \boldsymbol{z}$ (identity), $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \frac{\boldsymbol{z}}{||\boldsymbol{z}||}$ (normalization) and $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = (\boldsymbol{z} - \mu_{\boldsymbol{z}})/\sqrt{\sigma_{\boldsymbol{z}}^2 + \epsilon}$ (layer normalization), respectively, which were covered in §3.3. We used $\epsilon = 10^{-8}$, as is commonly done in layer normalization for numerical stability. We use $\alpha \in \{1, 1.5, 2\}$ for α-entmax and $\gamma \in \{2, 5\}$ for γ-normmax (where we apply the bisection algorithm described in Appendix A).

In Figure 8, as α increases, α-entmax converges more often to a single pattern, whereas γ-normmax tends to converge towards an attractor which is a uniform average of some patterns. This behavior is also observable in the basins of attraction (right plot), where larger values of α result in fewer regions converging to metastable states. In Figure 9, it is observed that the converged patterns consistently align along the circle with infinite energy outside. This observation is in line with expectations, considering the function $\hat{\boldsymbol{y}}_\Psi$, derived from $\Psi(\boldsymbol{q}) = I_{\|.\| \leq r}(\boldsymbol{q})$, with $r = 1$, that reflects the projected space constraint performed by normalization. In this figure, the local minima of the energy function tend to cluster around a set of memories, whereas in the basins of attraction, the trends closely resemble those in Figure 8, with the exception of the softmax and 1.5-entmax cases, where more attraction
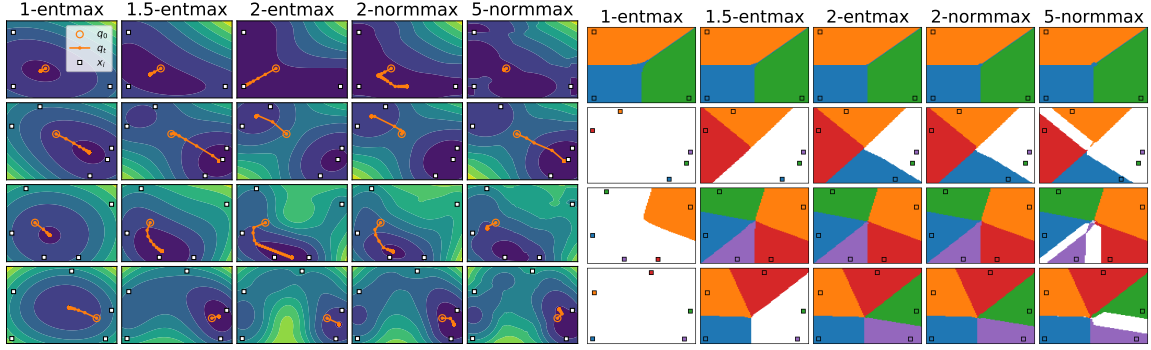
Figure 8: Left: contours of the energy function and optimization trajectory of the CCCP iteration ($\beta = 1$) for $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \boldsymbol{z}$. Right: attraction basins associated with each pattern ($\beta = 10$; a larger β is needed to allow for the 1-entmax to get T-close to a single pattern). White sections converge to a metastable state; for $\alpha = 1$ we allow a tolerance of $T = .01$).
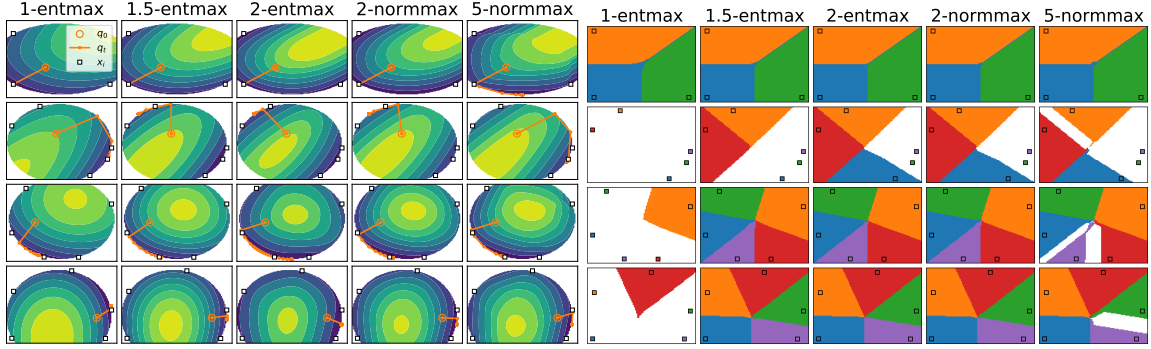

Figure 9: Left: contours of the energy function and optimization trajectory of the CCCP iteration ($\beta = 1$) for $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}$. The white regions correspond to infinite energies, due to the hard constraints. Right: attraction basins associated with each pattern ($\beta = 10$).

areas are present. In Figure 10, we generate synthetic 3D data and plot the contours and trajectories obtained through the Hopfield update rules (CCCP iterations). In this case, we have $\Psi(\boldsymbol{q}) = I_S(\boldsymbol{q})$, where $S = \{\boldsymbol{q} \mid \|\boldsymbol{q}\| \leq \sqrt{D-1} \text{ and } \mathbf{1}^\top \boldsymbol{q} = 0\}$, which corresponds to layer normalization, and therefore we can represent a query $\boldsymbol{q} = (q_1, q_2, q_3)$ in the 2D plane through coordinates (q_1, q_2), with $q_3 = -q_1 - q_2$. After the first iteration, the points converge to the conditions specified by this indicator function, *i.e.*, $\mathbf{1}^\top \boldsymbol{q} = 0$ plane and $\sqrt{D-1}$ radius sphere.[7] This is a special case of the scenario discussed in §3 where no trainable parameters

---

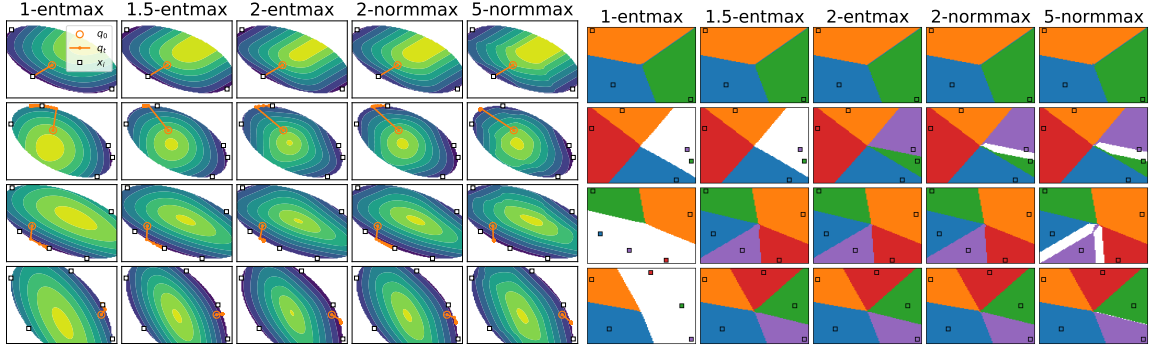7. The $D-1$ term arises because the layer normalization operation uses an unbiased standard deviation.

Figure 10: Left: contours of the energy function and optimization trajectory of the CCCP iteration ($\beta = 1$) for $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = (\boldsymbol{z} - \mu_{\boldsymbol{z}})/\sqrt{\sigma_{\boldsymbol{z}}^2 + \epsilon}$. Here we show trajectories for 3D data points and contours according with the $\Psi(\boldsymbol{q})$ restrictions mentioned in §3, where the contours lie in the plane $z = -x - y$, intersected with the sphere of radius $\sqrt{D-1}$. White regions correspond to infinite energy. Right: attraction basins associated with each pattern ($\beta = 10$).

are present. Similarly to normalization, trajectories tend to converge to clusters of memories, and the basins of attraction exhibit a greater number of attraction areas.

7.3 Retrieval capacity

We next assess the ability of HFY networks to handle growing quantities of stored memories (Figure 11), and noise (Figure 12), across various choices of $\hat{\boldsymbol{y}}_\Omega$ and $\hat{\boldsymbol{y}}_\Psi$, including the specific cases in Table 2. We assess the retrieval capacity on three image datasets: MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky, 2009), and Tiny ImageNet (Le et al., 2015). Prior to inputting the images as queries to the network, we normalize all pixel values to the interval $[-1, 1]$. The images are flattened into a single vector when fed to the Hopfield network. During the masking process, pixels outside the mask were set to 0. When introducing Gaussian noise to the images, we ensured that pixel values were clipped, preserving all values within the $[-1, 1]$ interval. A query is successfully retrieved when its cosine similarity falls above a predefined threshold of $\epsilon > 0.9$. Plotted are the medians of the ratio of successfully retrieved patterns (success retrieval rate) and the interquartile range for 5 runs with different memories for the methods described in Table 2.

In Figure 11 and Figure 12, we can observe that the classic Hopfield networks of Hopfield (1982) and the dense associative models of Krotov and Hopfield (2016) and Demircigil et al. (2017) struggle to successfully retrieve patterns, most noticeable for the former, even for a low number of memories or for low levels of noise. Notably, in modern Hopfield networks with $\beta = 1$ (first row of Figure 11), all variants—α-entmax and γ-normmax—show ideal behavior on the MNIST dataset. They also demonstrate accurate behavior on the other datasets, with or without the normalization and layer normalization post-transformations. Additionally, these methods exhibit graceful degradation as the number of stored memories increases.
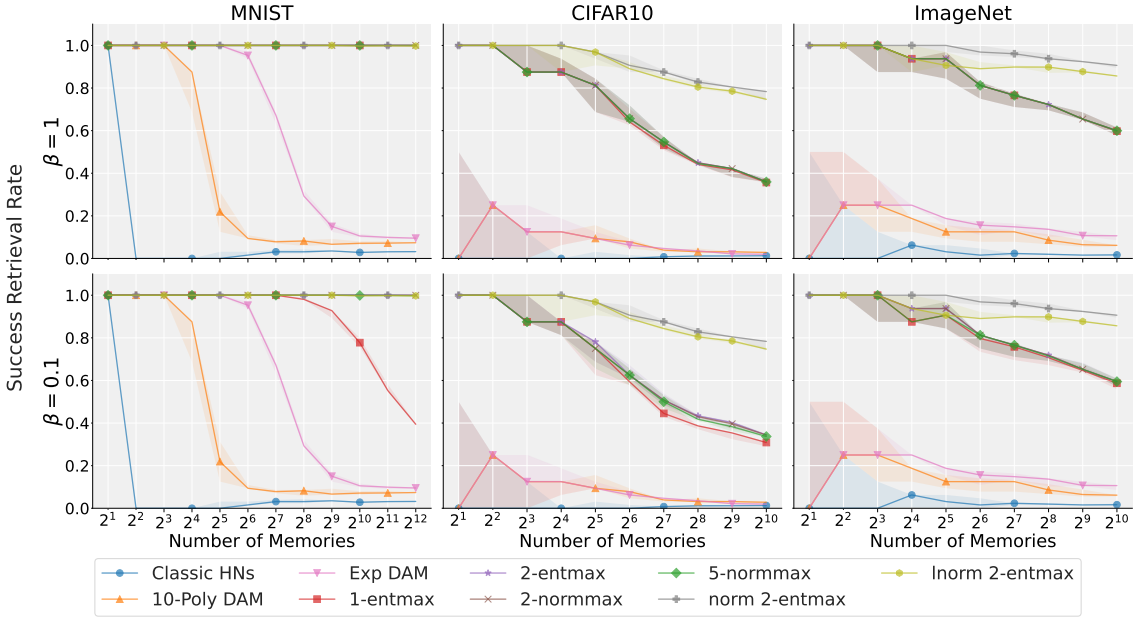
Figure 11: Memory capacity for different numbers of stored memories for $\beta = 0.1$ (bottom) and $\beta = 1$ (top). For $\beta = 1$, entmax and normmax lines intersect. Norm stands for ℓ_2 normalization, which corresponds to $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \boldsymbol{z}/\|\boldsymbol{z}\|$, while lnorm, short for layer normalization, corresponds to $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = (\boldsymbol{z} - \mu_{\boldsymbol{z}})/\sqrt{\sigma_{\boldsymbol{z}}^2 + \epsilon}$. Plotted are the medians over 5 runs with different memories and the interquartile range.

A similar behavior is observed for $\beta = 0.1$ (second row), with performance improving as α increases, for α-entmax methods, as γ decreases for γ-normmax. One can also see that 2-entmax with $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \boldsymbol{z}/\|\boldsymbol{z}\|$ (normalization) demonstrates even better performance across all datasets, indicating the positive contribution of this specific $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z})$. Superior performance is also observed with $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = (\boldsymbol{z} - \mu_{\boldsymbol{z}})/\sqrt{\sigma_{\boldsymbol{z}}^2 + \epsilon}$ (layer normalization), although not as good as the former. Similar behavior can be observed in Figure 12 but now in terms of the noise standard deviation. Detailed plots of the HFY networks, using $\hat{\boldsymbol{y}}_\Omega$ as either α-entmax or γ-normmax for different α and γ values and different $\hat{\boldsymbol{y}}_\Psi$, can be found in Appendix C.1.

7.4 Sparse and structured transformations in multiple instance learning

In multiple instance learning (MIL), instances are grouped into bags and the goal is to predict the label of each bag based on the instances it contains. If a bag contains at least one item from a given class, we consider it positive. This holds true even when we only know whether that particular instance is present in the bag, without needing to know the quantity. This framework is particularly useful in situations where annotating individual instances is challenging or impractical, while bag-level labels are easier to obtain. Instances of such scenarios include medical imaging, where a bag might represent an image, instances could
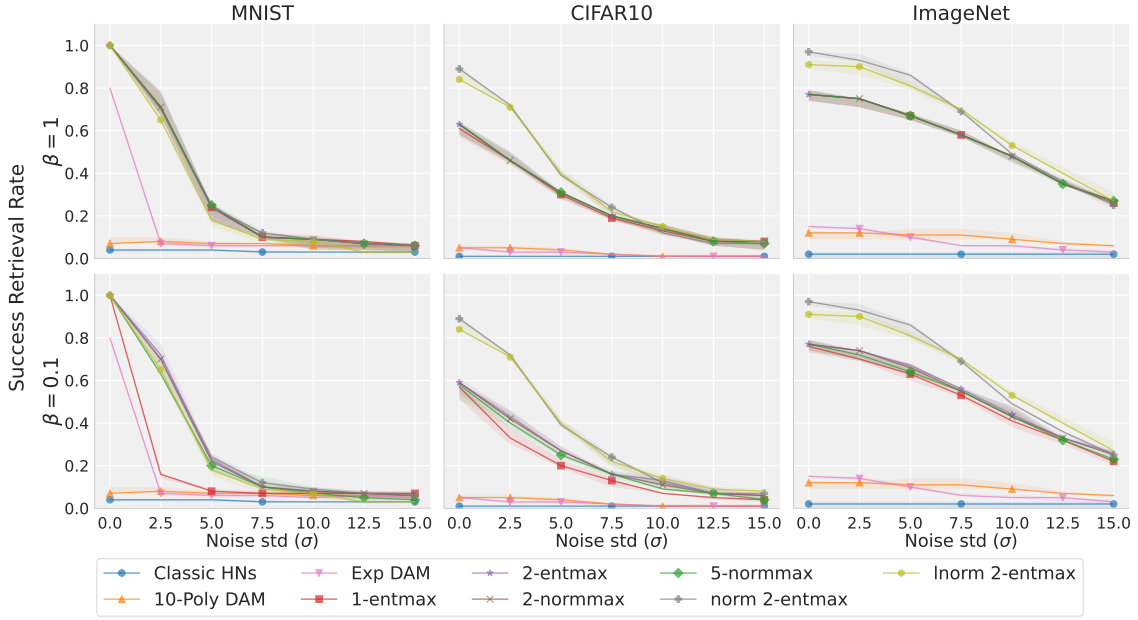
31

Figure 12: Memory robustness against different levels of noise for $\beta = 0.1$ (bottom) and $\beta = 1$ (top). For $\beta = 1$ entmax and normmax lines intersect.

manifest as patches within the image, and the label signifies the presence or absence of a disease. We also consider a extended variant, denoted K-MIL, where bags are considered positive if they contain K or more positive instances; MIL is recovered when $K = 1$. K-MIL with $K > 1$ can be useful in scenarios where instance labels or uncertain, or where precision in bag labels is more important than recall. Our k-subsets method is particularly suitable for this problem, due to its ability to retrieve k patterns.

Ramsauer et al. (2021) tackle MIL via a Hopfield pooling layer, where the query $\boldsymbol{q}$ is learned and the keys $\boldsymbol{X}$ are instance embeddings. This approach closely resembles transformer self-attention with pre- and post-operations such as layer normalization, which differs slightly from a pure Hopfield layer. We experiment with the sparse variants of the Hopfield pooling layer introduced by Ramsauer et al. (2021), as these layers contain more parameters, making them stronger pooling approximators. We use our proposed α-entmax and γ-normmax transformations (§4), as well as structured variants using SparseMAP with k-subsets (§5), varying α, γ and k in each case. We run these models for K-MIL problems in the MNIST dataset (choosing '9' as target as it can be easily misunderstood with '7' or '4'), and on three MIL benchmarks: the Elephant, Fox, and Tiger datasets (Ilse et al., 2018). We experiment with $K \in \{2, 3, 5\}$. Further details can be found in Appendix C.2 and C.3.

Table 4 shows the results. We observe that for MNIST with $K = 1$, 1-entmax outperforms the remaining methods. Normmax shows consistent results across datasets achieving near-optimal performance, arguably due to its ability to adapt to near-uniform metastable states of varying size. We also observe that, for $K > 1$, the k-subsets approach achieves top

Table 4: Results for MIL. We show accuracies for MNIST and ROC AUC for MIL benchmarks, averaged across 5 runs.

Methods	MNIST				MIL benchmarks		
	$K{=}1$	$K{=}2$	$K{=}3$	$K{=}5$	Fox	Tiger	Elephant
1-entmax (softmax)	$\mathbf{98.4 \pm 0.2}$	94.6 ± 0.5	91.1 ± 0.5	89.0 ± 0.3	66.4 ± 2.0	87.1 ± 1.6	92.6 ± 0.6
1.5-entmax	97.6 ± 0.8	96.0 ± 0.9	90.4 ± 1.1	92.4 ± 1.4	66.3 ± 2.0	87.3 ± 1.5	92.4 ± 1.0
2.0-entmax (sparsemax)	97.9 ± 0.2	96.7 ± 0.5	92.9 ± 0.9	91.6 ± 1.0	66.1 ± 0.6	$\mathbf{87.7 \pm 1.4}$	91.8 ± 0.6
2.0-normmax	97.9 ± 0.3	96.6 ± 0.6	93.9 ± 0.7	92.4 ± 0.7	66.1 ± 2.5	86.4 ± 0.8	92.4 ± 0.7
5.0-normmax	98.2 ± 0.5	97.2 ± 0.3	95.8 ± 0.4	93.2 ± 0.5	66.4 ± 2.3	85.5 ± 0.6	93.0 ± 0.7
SparseMAP, $k = 2$	97.9 ± 0.3	$\mathbf{97.7 \pm 0.3}$	95.1 ± 0.5	92.6 ± 1.1	66.8 ± 2.7	85.3 ± 0.5	$\mathbf{93.2 \pm 0.7}$
SparseMAP, $k = 3$	98.0 ± 0.6	96.1 ± 1.0	$\mathbf{96.5 \pm 0.5}$	92.2 ± 1.2	$\mathbf{67.4 \pm 2.0}$	86.1 ± 0.8	92.6 ± 1.7
SparseMAP, $k = 5$	98.2 ± 0.4	96.2 ± 1.4	95.1 ± 1.1	$\mathbf{95.1 \pm 1.5}$	67.0 ± 2.0	86.3 ± 0.8	91.2 ± 1.0

performance when $k = K$, as expected. We conjecture that this is due to the ability of SparseMAP with k-subsets for $k = K$ to retrieve exactly the K positive instances in the bag, whereas other values of k might under or over-retrieve. In the MIL benchmarks, SparseMAP pooling surpasses sparse pooling variants for 2 out of 3 datasets.

7.5 Post-transformations in multiple instance learning

In the previous experiment, we worked with extended variants of the Hopfield pooling layers from Ramsauer et al. (2021), which are designed to resemble self-attention mechanisms in transformers, incorporating distinct pre- and post-layer normalization for the queries and memories. These layers optionally normalize queries and keys with layer normalization, project them, and then layer-normalize them again each with different learnable parameters leading to different keys and values, as shown in the following pipeline, where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$ are projection matrices (see Ramsauer et al. 2021, Figure A.7):

- **Queries:** $\boldsymbol{q} \mapsto \boldsymbol{q}' = \boldsymbol{W}_Q^\top \text{LayerNorm}(\boldsymbol{q}) \mapsto \boldsymbol{q}^{(0)} = \text{LayerNorm}(\boldsymbol{q}')$

- **Keys:** $\boldsymbol{x}_i \mapsto \boldsymbol{x}'_i = \boldsymbol{W}_K^\top \text{LayerNorm}(\boldsymbol{x}_i) \mapsto \boldsymbol{k}_i = \text{LayerNorm}(\boldsymbol{x}'_i)$

- **Values:** $\boldsymbol{x}_i \mapsto \boldsymbol{x}'_i = \boldsymbol{W}_V^\top \text{LayerNorm}(\boldsymbol{x}_i) \mapsto \boldsymbol{v}_i = \text{LayerNorm}(\boldsymbol{x}'_i)$

These operations are followed by the Hopfield update $\boldsymbol{q}^{(t+1)} = \boldsymbol{V}^\top \hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{K} \boldsymbol{q}^{(t)})$, where $\boldsymbol{K} = [\boldsymbol{k}_1^\top; ...; \boldsymbol{k}_N^\top] \in \mathbb{R}^{N \times D}$ and $\boldsymbol{V} = [\boldsymbol{v}_1^\top; ...; \boldsymbol{v}_N^\top] \in \mathbb{R}^{N \times D}$. Despite its higher expressiveness, which motivated its use by Ramsauer et al. (2021) and in our previous experiment, this approach also contrasts with "pure" Hopfield layers, where keys must equal the values. Therefore, we experiment also with different post-transformations in the pure Hopfield scenario, which matches precisely the derived theoretical framework. We experiment in Table 5 with pure Hopfield layers using different $\hat{\boldsymbol{y}}_\Psi$ functions, namely the identity, ℓ_2-normalization, and layer normalization (see §3). Note that, for the post-transformation identity, 1-entmax recovers Ramsauer et al. (2021) without extra parametrizations and 2-entmax recovers Hu et al. (2023). Identity is represented by $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \boldsymbol{z}$. For ℓ_2-normalization, we use $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \frac{r\boldsymbol{z}}{\|\boldsymbol{z}\|}$ with $r = 1$. In the case of layer normalization, we apply $\hat{\boldsymbol{y}}_\Psi(\boldsymbol{z}) = \eta \frac{\boldsymbol{z} - \mu_{\boldsymbol{z}}}{\sqrt{\sigma_{\boldsymbol{z}}^2 + \epsilon}} + \boldsymbol{\delta}$, where η and $\boldsymbol{\delta}$ are learnable parameters. The post-transformations are applied to both the initial query and memory, projecting them into the space of the Hopfield output.

Table 5: Results for MIL. We show ROC AUC, averaged across 5 runs. We bold the top performing model for each dataset and underline the best $\hat{\boldsymbol{y}}_\Psi$ for each method.

Methods	Post-Transformation	Fox	Tiger	Elephant
1-entmax (softmax)	Identity	63.6 ± 1.7	86.9 ± 1.0	91.3 ± 1.0
	ℓ_2 normalization	$\underline{64.3 \pm 2.4}$	$\underline{87.0 \pm 0.8}$	$\underline{91.6 \pm 0.4}$
	LayerNorm	62.1 ± 2.3	$\underline{87.0 \pm 0.9}$	91.2 ± 1.0
1.5-entmax	Identity	61.6 ± 3.8	86.7 ± 0.9	$\underline{92.0 \pm 0.4}$
	ℓ_2 normalization	$\underline{64.2 \pm 2.4}$	86.7 ± 0.4	91.5 ± 0.7
	LayerNorm	63.4 ± 1.6	$\underline{87.0 \pm 0.9}$	$\underline{92.0 \pm 0.4}$
2-entmax (sparsemax)	Identity	63.7 ± 1.7	86.8 ± 0.9	91.6 ± 0.5
	ℓ_2 normalization	63.4 ± 2.7	$\underline{87.6 \pm 1.0}$	90.6 ± 0.8
	LayerNorm	$\underline{63.4 \pm 1.6}$	85.0 ± 1.3	$\underline{91.7 \pm 0.5}$
2-normmax	Identity	63.7 ± 1.7	86.7 ± 0.9	92.0 ± 0.4
	ℓ_2 normalization	$\underline{64.2 \pm 2.4}$	$\mathbf{87.7 \pm 0.6}$	$\mathbf{92.6 \pm 0.8}$
	LayerNorm	63.4 ± 1.6	87.0 ± 0.9	91.9 ± 0.4
5-normmax	Identity	61.9 ± 1.7	86.9 ± 1.0	$\underline{91.9 \pm 0.6}$
	ℓ_2 normalization	64.2 ± 2.4	$\underline{87.5 \pm 0.7}$	91.3 ± 0.7
	LayerNorm	$\mathbf{64.6 \pm 3.1}$	87.0 ± 0.9	$\underline{91.9 \pm 0.6}$

Table 5 displays the results for the MIL benchmarks. We see that both ℓ_2-normalization and layer normalization post-transformations lead to clear benefits for all methods: across the three datasets and five models, ℓ_2-normalization outperforms or matches the other post-transformations in 10 out of 15 cases, whereas layer normalization outperforms or matches the other post-transformations in 7 out of 15 entries.

7.6 Structured Rationalizers

Finally, we experiment with rationalizer models in sentiment prediction tasks, where the inputs are sentences or documents in natural language and the rationales are text highlights (see Figure 13 for an illustration). These models, sometimes referred as select-predict or explain-predict models (Jacovi and Goldberg, 2021; Zhang et al., 2021), consist of a rationale generator and a predictor. The generator processes the input text and extracts the rationale as a subset of words to be highlighted, and the predictor classifies the input based solely on the extracted rationale, which generally involves concealing non-rationale words through the application of a binary mask. Rationalizers are usually trained end-to-end, and the discreteness of the latent rationales is either handled with stochastic methods via score function estimators or the reparametrization trick (Lei et al., 2016; Bastings et al., 2019), or with deterministic methods via structured continuous relaxations (Guerreiro and Martins, 2021). In either case, the model imposes sparsity and contiguity penalties to ensure rationales are short and tend to extract adjacent words.

Our model architecture is adapted from SPECTRA (Guerreiro and Martins, 2021), but the combination of the generator and predictor departs from prior approaches (Lei et al., 2016; Bastings et al., 2019; Guerreiro and Martins, 2021) in which the predictor does not

Figure 13: Examples of human rationale overlap for the aspect "appearance". The yellow highlight indicates the model's rationale, while *italicized and bold font* represents the human rationale. Red font identifies mismatches with human annotations. SparseMAP with sequential k-subsets prefers more contiguous rationales, which better match humans.

"mask" the input tokens; instead, it takes as input the pooled vector that results from the Hopfield pooling layer (either a sequential or non-sequential SparseMAP k-subsets layer). By integrating this Hopfield pooling layer into the predictor, we transform the sequence of word embeddings into a single vector from which the prediction is made. The rationale is formed by the pattern associations (word tokens) extracted by the Hopfield layer. We use the same hyperparameters as Guerreiro and Martins (2021). We use a head dimension of 200, to match the dimensions of the encoder vectors (the size of the projection matrices associated to the static query and keys) and a head dropout of 0.5 (applied to the output of the Hopfield layer). We use a transition score of 0.001 and a train temperature of 0.1.

Table 6 shows the results on the downstream task (classification for SST, AgNews, IMDB; regression for BeerAdvocate) and the F_1 overlap with human rationales for the BeerAdvocate dataset (McAuley et al., 2012). Compared to strong baselines (Bastings et al., 2019; Guerreiro and Martins, 2021), our methods achieve equal or slightly superior performance for all datasets. Moreover, our sequential k-subsets model outperforms the baselines in terms of overlap with human rationales, arguably due to the fact that human rationales tend to contain adjacent words, which is encouraged by our sequential model.

Table 6: Text rationalization results. We report mean and min/max F_1 scores across five random seeds on test sets for all datasets but Beer, where we report MSE. All entries except SparseMAP are taken from Guerreiro and Martins (2021). We also report human rationale overlap (HRO) as F_1 score. We bold the best-performing rationalized model(s).

Method	Rationale	SST↑	AgNews↑	IMDB↑	Beer↓	Beer(HRO)↑
SFE	top-k	.76 (.71/.80)	.92 (.92/.92)	.84 (.72/.88)	.018 (.016/.020)	.19 (.13/.30)
	contiguous	.71 (.68/.75)	.86 (.85/.86)	.65 (.57/.73)	.020 (.019/.024)	.35 (.18/.42)
SFE w/Baseline	top-k	.78 (.76/.80)	.92 (.92/.93)	.82 (.72/.88)	.019 (.017/.020)	.17 (.14/.19)
	contiguous	.70 (.64/.75)	.86 (.84/.86)	.76 (.73/.80)	.021 (.019/.025)	.41 (.37/.42)
Gumbel	top-k	.70 (.67/.72)	.78 (.73/.84)	.74 (.71/.78)	.026 (.018/.041)	.27 (.14/.39)
	contiguous	.67 (.67/.68)	.77 (.74/.81)	.72 (.72/.73)	.043 (.040/.048)	.42 (.41/.42)
HardKuma	-	.80 (.80/.81)	.90 (.87/.88)	.87 (.90/.91)	.019 (.016/.020)	.37 (.00/.90)
Sparse Attention	sparsemax	**.82** (.81/.83)	**.93** (.93/.93)	.89 (.89/.90)	.019 (.016/.021)	.48 (.41/.55)
	fusedmax	.81 (.81/.82)	.92 (.91/.92)	.88 (.87/.89)	.018 (.017/.019)	.39 (.29/.53)
SPECTRA	seq. k-subsets	.80 (.79/.81)	.92 (.92/.93)	**.90** (.89/.90)	**.017** (.016/.019)	.61 (.56/.68)
SparseMAP	k-subsets	.81 (.81/.82)	**.93** (.92/.93)	**.90** (.90/.90)	**.017** (.017/.018)	.42 (.29/.62)
	seq. k-subsets	.81 (.80/.83)	**.93** (.93/.93)	**.90** (.90/.90)	.020 (.018/.021)	**.63** (.49/.70)

8. Related Work

Hopfield networks trace their origins to the works of Amari (1972), Nakano (1972), Amari and Maginu (1988), and Hopfield (1982). A pivotal moment spurring increased research interest occurred in Krotov and Hopfield (2016), which introduced a novel polynomial energy function, followed by exponential dense associative memories (Demircigil et al., 2017). While these models were initially designed for binary cases, the work by Ramsauer et al. (2021) generalized them to continuous states, resembling attention mechanisms in transformers. These works were further extended to induce sparsity by Hu et al. (2023), the first work which proposed sparse Hopfield networks and which derived retrieval error bounds tighter than the dense analog. This was followed by two lines of work which developed in parallel: Wu et al. (2024b) proposed a "generalized sparse Hopfield model" based on α-entmax with learnable α, which they successfully applied to time series prediction problems; and Martins et al. (2023) and Santos et al. (2024) considered α-entmax and γ-normmax, along with analysis and conditions for exact retrieval. Our work generalizes these previous approaches as specific instances of a broader family of energy functions presented in §3 which are expressed as a difference between two Fenchel-Young losses (see equation 7). Additionally, neither Hu et al. (2023) nor Wu et al. (2024b) explored the potential of achieving exact retrieval through sparse transformations. Our work addresses this gap by presenting a unified framework for sparse Hopfield networks with enhanced theoretical guarantees for retrieval and coverage (see Propositions 9–10). Furthermore, the derived framework extends their constructions and broadens the applicability to new families, including γ-normmax. An effective bisection algorithm for γ-normmax (Algorithm 4) is introduced. The link with Fenchel-Young losses allowed the derivation of many results, such as the margin conditions, which were found to have direct application to sparse Hopfield networks. We leveraged this

framework to accommodate structure where we explored the structured margin of SparseMAP (Proposition 13), key to establish exact retrieval of pattern associations (Proposition 14).

A related approach to structure was later explored in Hu et al. (2024), where a particular instance of their structure involves a top-k modern Hopfield network, which relates to our k-subsets example. Our approach simplifies the process by adjusting $\hat{\boldsymbol{y}}_\Omega$ as part of an optimization problem. Our k-subsets example also relates to the top-k retrieval model introduced by Davydov et al. (2023). Their model diverges from ours as they employ an entropic regularizer that does not support sparsity, thereby making exact retrieval impossible. Our sparse and structured Hopfield layers in §5, with sparsemax and SparseMAP, involve a quadratic regularizer, which relates to the differentiable layers of Amos and Kolter (2017). The use of SparseMAP and its active set algorithm (Niculae et al., 2018) allows to exploit the structure of the problem to ensure efficient Hopfield updates and implicit derivatives.

Millidge et al. (2022) introduced universal Hopfield networks, unpacking associative memory models into three operations: similarity, separation, and projection. This framework closely aligns with the general Fenchel-Young framework proposed in this paper (Proposition 2), which can also accommodate various alternative similarity metrics. This extension differs from their work where we can incorporate an additional optional operation: the post-transformation $\hat{\boldsymbol{y}}_\Psi$ in (7). This operation can implement ℓ_2 normalization or layer normalization over the produced Hopfield result, bridging the gap with transformers. The first case aligns with the normalization approach in Nguyen and Salazar (2019), while the second corresponds to the layer normalization used in Vaswani et al. (2017).

Memory retrieval has garnered significant attention in computational neuroscience, based on foundational work by early researchers (Anderson and Bower, 1972; Tulving and Thomson, 1973; Tulving, 1985), and is a crucial paradigm for understanding how neural systems access and use stored information. However, a gap remains in machine learning approaches that effectively model memory retrieval paradigms, such as free and sequential retrieval. A pioneering study by Recanatesi et al. (2015) used Hopfield networks to model free recall, but these models are limited to binary states. Recently, Naim et al. (2020) introduced a parameter-free, graph-based model to predict recall based on associative memory structure, but experiments are limited to word recall. Our work provides an alternative by incorporating continuous states through constrained and penalized sparse transformations.

9. Conclusions

We presented a unified framework for Hopfield networks that accommodates not only sparse and structured Hopfield networks but also recovers many known methods, such as classic Hopfield networks, dense associative memories, and modern Hopfield networks. Our framework hinges on a broad family of energy functions, based on convex duality, written as a difference of two Fenchel-Young losses, one parametrized by a generalized negentropy function and the other can be any convex function that relates with the post-transformation. By incorporating additional operations such as ℓ_2 normalization and layer normalization, we bridge the gap between Hopfield networks and transformer architectures, providing a theoretically grounded approach to more robust Hopfield-based attention mechanisms. A central result of our paper is the connection between the margin properties of certain Fenchel-Young losses and sparse Hopfield networks, establishing provable conditions for

exact retrieval. Moreover, we extend this framework to incorporate structure via the SparseMAP transformation, allowing for the retrieval of pattern associations favored by top-k or sequential top-k retrieval, rather than a single pattern. Finally, we apply and validate our broad family of energies on different memory recall paradigms. We also validate the effectiveness of our approach on image retrieval, multiple instance learning, and text rationalization tasks.

Acknowledgments

Appendix A. Bisection Algorithm for the Normmax Transformation

We derive here expressions for the normmax transformation along with a bisection algorithm to compute this transformation for general γ.

Letting $\Omega(\boldsymbol{y}) = -1 + \|\boldsymbol{y}\|_{\gamma} + I_{\triangle_N}(\boldsymbol{y})$ be the norm entropy, we have $(\nabla\Omega^*)(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{y}\in\triangle_N} \boldsymbol{\theta}^{\top}\boldsymbol{y} - \|\boldsymbol{y}\|_{\gamma}$. The Lagrangian function is $L(\boldsymbol{y}, \boldsymbol{\lambda}, \mu) = -\boldsymbol{\theta}^{\top}\boldsymbol{y} + \|\boldsymbol{y}\|_{\gamma} - \boldsymbol{\lambda}^{\top}\boldsymbol{y} + \mu(\mathbf{1}^{\top}\boldsymbol{y} - 1)$. Equating the gradient to zero and using $\nabla\|\boldsymbol{y}\|_{\gamma} = (\boldsymbol{y}/\|\boldsymbol{y}\|_{\gamma})^{\gamma-1}$, we get:

$$\mathbf{0} = \nabla_{\boldsymbol{y}} L(\boldsymbol{y}, \boldsymbol{\lambda}, \mu) = -\boldsymbol{\theta} + (\boldsymbol{y}/\|\boldsymbol{y}\|_{\gamma})^{\gamma-1} - \boldsymbol{\lambda} + \mu\mathbf{1}. \tag{29}$$

The complementarity slackness condition implies that, if $y_i > 0$, we must have $\lambda_i = 0$, therefore, we have for such $i \in \text{supp}(\boldsymbol{y})$:

$$-\theta_i + (y_i/\|\boldsymbol{y}\|_{\gamma})^{\gamma-1} + \mu = 0 \quad \Rightarrow \quad y_i = (\theta_i - \mu)^{\frac{1}{\gamma-1}}\|\boldsymbol{y}\|_{\gamma}. \tag{30}$$

Since we must have $\sum_{i\in\text{supp}(\boldsymbol{y})} y_i = 1$, we obtain $\|\boldsymbol{y}\|_{\gamma}^{-1} = \sum_{i\in\text{supp}(\boldsymbol{y})}(\theta_i - \mu)^{\frac{1}{\gamma-1}}$. Plugging into (30) and noting that, from (29), we have $\theta_i < \mu_i$ for $i \notin \text{supp}(\boldsymbol{y})$, we get, for $i \in [N]$:

$$y_i = \frac{(\theta_i - \mu)_{+}^{\frac{1}{\gamma-1}}}{\sum_{j\in\text{supp}(\boldsymbol{y})}(\theta_j - \mu)_{+}^{\frac{1}{\gamma-1}}}. \tag{31}$$

Moreover, since $\sum_{i\in\text{supp}(\boldsymbol{y})} y_i^{\gamma} = \|\boldsymbol{y}\|_{\gamma}^{\gamma}$, we obtain from (30):

$$\|\boldsymbol{y}\|_{\gamma}^{\gamma} = \sum_{i\in\text{supp}(\boldsymbol{y})} (\theta_i - \mu)^{\frac{\gamma}{\gamma-1}}\|\boldsymbol{y}\|_{\gamma}^{\gamma} \quad \Rightarrow \quad \sum_{i\in\text{supp}(\boldsymbol{y})} (\theta_i - \mu)^{\frac{\gamma}{\gamma-1}} = 1. \tag{32}$$

In order to compute the solution (31) we need to find μ satisfying (32). This can be done with a simple bisection algorithm if we find a lower and upper bound on μ.

We have, from (31), that $\mu = \theta_i - (y_i/\|\boldsymbol{y}\|_{\gamma})^{\gamma-1}$ for any $i \in \text{supp}(\boldsymbol{y})$. Letting $\theta_{\max} = \max_i \theta_i$ and $y_{\max} = \max_i y_i$, we have in particular that $\mu = \theta_{\max} - (y_{\max}/\|\boldsymbol{y}\|_{\gamma})^{\gamma-1}$. We also have that $y_{\max} = \|\boldsymbol{y}\|_{\infty} \leq \|\boldsymbol{y}\|_{\gamma}$, which implies $y_{\max}/\|\boldsymbol{y}\|_{\gamma} \leq 1$. Since $1/N \leq y_{\max} \leq 1$ and $\|\boldsymbol{y}\|_{\gamma} \leq 1$ for any $\boldsymbol{y} \in \triangle_N$, we also obtain $y_{\max}/\|\boldsymbol{y}\|_{\gamma} \geq (1/N)/1 = N^{-1}$. Therefore we have

$$\underbrace{\theta_{\max} - 1}_{\mu_{\min}} \leq \mu \leq \underbrace{\theta_{\max} - N^{1-\gamma}}_{\mu_{\max}}. \tag{33}$$

The resulting algorithm is shown as Algorithm 4.

Appendix B. Proofs of Main Text

B.1 Proof of Proposition 2

Recall that the energy is written as a difference of two Fenchel-Young losses:

$$E(\boldsymbol{q}) = \underbrace{-L_{\Omega}(\boldsymbol{X}\boldsymbol{q}, \boldsymbol{u})}_{E_{\text{concave}}(\boldsymbol{q})} + \underbrace{L_{\Psi}(\boldsymbol{X}^{\top}\boldsymbol{u}, \boldsymbol{q})}_{E_{\text{convex}}(\boldsymbol{q})} + \text{constant}. \tag{34}$$

Algorithm 4 Compute γ-normmax by bisection.

1: **Input:** Scores $\boldsymbol{\theta} = [\theta_1, ..., \theta_N]^\top \in \mathbb{R}^N$, parameter $\gamma > 1$, number of bisection iterations T
2: **Output:** Probability vector $\boldsymbol{y} = [y_1, ..., y_N]^\top \in \triangle_N$.
3: Define $\theta_{\max} \leftarrow \max_i \theta_i$
4: Set $\mu_{\min} \leftarrow \theta_{\max} - 1$ and $\mu_{\max} \leftarrow \theta_{\max} - N^{1-\gamma}$
5: **for** $t \in 1, \ldots, T$ **do**
6:     Set $\mu \leftarrow (\mu_{\min} + \mu_{\max})/2$ and $Z \leftarrow \sum_j (\theta_j - \mu)_+^{\frac{\gamma}{\gamma-1}}$
7:     **if** $Z < 1$ **then** $\mu_{\max} \leftarrow \mu$ **else** $\mu_{\min} \leftarrow \mu$
8: Return $\boldsymbol{y} = [y_1, ..., y_N]^\top$ with $y_i = (\theta_i - \mu)_+^{\frac{1}{\gamma-1}} / \sum_j (\theta_j - \mu)_+^{\frac{1}{\gamma-1}}$.

The CCCP algorithm works as follows: at the tth iteration, it linearizes the concave function E_{concave} by using a first-order Taylor approximation around $\boldsymbol{q}^{(t)}$,

$$E_{\text{concave}}(\boldsymbol{q}) \approx \tilde{E}_{\text{concave}}(\boldsymbol{q}) := E_{\text{concave}}(\boldsymbol{q}^{(t)}) + \left(\frac{\partial E_{\text{concave}}(\boldsymbol{q}^{(t)})}{\partial \boldsymbol{q}} \right)^\top (\boldsymbol{q} - \boldsymbol{q}^{(t)}).$$

Then, it computes a new iterate by solving the convex optimization problem $\boldsymbol{q}^{(t+1)} := \arg\min_{\boldsymbol{q}} E_{\text{convex}}(\boldsymbol{q}) + \tilde{E}_{\text{concave}}(\boldsymbol{q})$, which leads to $\nabla E_{\text{convex}}(\boldsymbol{q}^{(t+1)}) = -\nabla E_{\text{concave}}(\boldsymbol{q}^{(t)})$. Using the fact, from Proposition 1, that $\nabla L_\Omega(\boldsymbol{\theta}, \boldsymbol{y}) = \hat{\boldsymbol{y}}_\Omega(\boldsymbol{\theta}) - \boldsymbol{y}$ and the chain rule leads to

$$\nabla E_{\text{concave}}(\boldsymbol{q}) = -\nabla_{\boldsymbol{q}} L_\Omega(\boldsymbol{X}\boldsymbol{q}; \boldsymbol{u}) = \boldsymbol{X}^\top \boldsymbol{u} - \boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\boldsymbol{X}\boldsymbol{q}),$$
$$\nabla E_{\text{convex}}(\boldsymbol{q}) = -\boldsymbol{X}^\top \boldsymbol{u} + \nabla \Psi(\boldsymbol{q}), \tag{35}$$

leading to the equation $\nabla \Psi(\boldsymbol{q}^{(t+1)}) = \boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\boldsymbol{X}\boldsymbol{q}^{(t)})$. Using the property that $\nabla \Psi(\boldsymbol{q}) = \boldsymbol{\eta}$ is equivalent to $\boldsymbol{q} = \nabla \Psi^*(\boldsymbol{\eta})$, *i.e.*, that $(\nabla \Psi)^{-1} = \nabla \Psi^*$, we finally obtain:

$$\boldsymbol{q}^{(t+1)} = \nabla \Psi^* \left(\boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\boldsymbol{X}\boldsymbol{q}^{(t)}) \right) = \hat{\boldsymbol{y}}_\Psi \left(\boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\boldsymbol{X}\boldsymbol{q}^{(t)}) \right), \tag{36}$$

which leads to the update equation (7).

B.2 Proof of Proposition 3

Let $\Psi(\boldsymbol{q}) = I_S(\boldsymbol{q})$ with $S := \{\boldsymbol{q} : \|\boldsymbol{q} - \boldsymbol{\delta}\| \leq \eta\sqrt{D} \wedge \mathbf{1}^\top (\boldsymbol{q} - \boldsymbol{\delta}) = 0\}$. We start by showing that, if $f(\boldsymbol{q}) := I_F(\boldsymbol{q})$ with $F := \{\|\boldsymbol{q}\| \leq 1 \wedge \mathbf{1}^\top \boldsymbol{q} = 0\}$, we have $(\nabla f^*)(\boldsymbol{z}) = \frac{\boldsymbol{z} - \mu_{\boldsymbol{z}}\mathbf{1}}{\|\boldsymbol{z} - \mu_{\boldsymbol{z}}\mathbf{1}\|}$. By definition, we have $f^*(\boldsymbol{z}) = \max_{\boldsymbol{q}} \boldsymbol{z}^\top \boldsymbol{q} = -\min_{\boldsymbol{q}} -\boldsymbol{z}^\top \boldsymbol{q}$ subject to $\mathbf{1}^\top \boldsymbol{q} = 0$ and $\|\boldsymbol{q}\| \leq 1$. Introducing Lagrange multipliers μ and $\lambda \geq 0$, we obtain the Lagrangian function $L(\boldsymbol{q}, \mu, \lambda) = -\boldsymbol{z}^\top \boldsymbol{q} + \mu\mathbf{1}^\top \boldsymbol{q} + \lambda(\|\boldsymbol{q}\| - 1)$. We have

$$\mathbf{0} = \nabla L(\boldsymbol{q}, \mu, \lambda) = -\boldsymbol{z} + \mu\mathbf{1} + \lambda\boldsymbol{q}/\|\boldsymbol{q}\|, \tag{37}$$

which implies $0 = -\mathbf{1}^\top \boldsymbol{z} + D\mu + \lambda\mathbf{1}^\top \boldsymbol{q}/\|\boldsymbol{q}\|$. Since we must have $\mathbf{1}^\top \boldsymbol{q} = 0$, this implies that $\mu = \frac{\mathbf{1}^\top \boldsymbol{z}}{D} = \mu_{\boldsymbol{z}}$. Therefore, we can write (37) as $\boldsymbol{z} - \mu_{\boldsymbol{z}}\mathbf{1} = \lambda\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}$. Taking the norm in both sides, we obtain $|\lambda| = \|\boldsymbol{z} - \mu_{\boldsymbol{z}}\mathbf{1}\|$; since $\lambda \geq 0$, we have $\lambda = \|\boldsymbol{z} - \mu_{\boldsymbol{z}}\mathbf{1}\|$. Next, we observe that, while $\boldsymbol{q}$ is constrained as $\|\boldsymbol{q}\| \leq 1$, the objective $\boldsymbol{z}^\top \boldsymbol{q}$ is maximized when $\|\boldsymbol{q}\| = 1$, and

therefore we obtain $\boldsymbol{q}^\star = (\nabla f^*)(\boldsymbol{z}) = \frac{\boldsymbol{z}-\mu_{\boldsymbol{z}}\mathbf{1}}{\|\boldsymbol{z}-\mu_{\boldsymbol{z}}\|}$. We also obtain

$$f^*(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{q}^\star = \frac{\|\boldsymbol{z}\|^2 - d\mu_{\boldsymbol{z}}}{\|\boldsymbol{z}-\mu_{\boldsymbol{z}}\|} = \|\boldsymbol{z}-\mu_{\boldsymbol{z}}\|. \tag{38}$$

Now, consider $g(\boldsymbol{q}) := I_G(\boldsymbol{q})$ with $G := \{\|\boldsymbol{q}\| \leq r \ \wedge \ \mathbf{1}^\top\boldsymbol{q} = 0\}$ for some $r > 0$. We can write $g(\boldsymbol{q}) = f(\boldsymbol{q}/r)$, and using the linear transformation property in Table 1, we have $g^*(\boldsymbol{z}) = f^*(r\boldsymbol{z})$, and therefore $(\nabla g^*)(\boldsymbol{z}) = r(\nabla f^*)(r\boldsymbol{z}) = r\frac{\boldsymbol{z}-\mu_{\boldsymbol{z}}\mathbf{1}}{\|\boldsymbol{z}-\mu_{\boldsymbol{z}}\|}$. When $r = \eta\sqrt{D}$ this becomes $(\nabla g^*)(\boldsymbol{z}) = \eta\frac{\boldsymbol{z}-\mu_{\boldsymbol{z}}\mathbf{1}}{\sigma_{\boldsymbol{z}}}$, where $\sigma_{\boldsymbol{z}} := \sqrt{\frac{1}{D}\sum_i(z_i-\mu_{\boldsymbol{z}})^2}$. Finally, observe that we can write $\Psi(\boldsymbol{q}) = g(\boldsymbol{q} - \boldsymbol{\delta})$. From the translation property in Table 1, we then have $\Psi^*(\boldsymbol{z}) = \boldsymbol{\delta}^\top \boldsymbol{z} + g^*(\boldsymbol{z})$. This leads to $(\nabla\Psi^*)(\boldsymbol{z}) = \boldsymbol{\delta} + (\nabla g^*)(\boldsymbol{z}) = \text{LayerNorm}(\boldsymbol{z}; \eta, \boldsymbol{\delta})$.

B.3 Proof of Proposition 7

We start by proving that $E(\boldsymbol{q}) \geq 0$. We show first that for any Ω satisfying conditions 1–3 above, we have

$$L_\Omega(\boldsymbol{\theta}; \mathbf{1}/N) \leq \max_i \theta_i - \mathbf{1}^\top\boldsymbol{\theta}/N. \tag{39}$$

From the definition of Ω^* and the fact that $\Omega(\boldsymbol{y}) \geq \Omega(\mathbf{1}/N)$ for any $\boldsymbol{y} \in \triangle_N$, we have that, for any $\boldsymbol{\theta}$, $\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{y}\in\triangle_N}\boldsymbol{\theta}^\top\boldsymbol{y} - \Omega(\boldsymbol{y}) \leq \max_{\boldsymbol{y}\in\triangle_N}\boldsymbol{\theta}^\top\boldsymbol{y} - \Omega(\mathbf{1}/N) = \max_i\theta_i - \Omega(\mathbf{1}/N)$, which leads to (39).

Let now $k = \arg\max_i \boldsymbol{q}^\top\boldsymbol{x}_i$, i.e., $\boldsymbol{x}_k$ is the pattern most similar to the query $\boldsymbol{q}$. We have

$$\begin{aligned}
E(\boldsymbol{q}) &= -\beta^{-1}L_\Omega(\beta\boldsymbol{X}\boldsymbol{q}; \mathbf{1}/N) + \frac{1}{2}\|\boldsymbol{q}-\boldsymbol{\mu_X}\|^2 + \frac{1}{2}(M^2 - \|\boldsymbol{\mu_X}\|^2) \\
&\geq -\beta^{-1}(\beta\max_i\boldsymbol{q}^\top\boldsymbol{x}_i - \beta\mathbf{1}^\top\boldsymbol{X}\boldsymbol{q}/N) + \frac{1}{2}\|\boldsymbol{q}-\boldsymbol{\mu_X}\|^2 + \frac{1}{2}(M^2 - \|\boldsymbol{\mu_X}\|^2) \\
&= -\boldsymbol{q}^\top\boldsymbol{x}_k + \boldsymbol{q}^\top\boldsymbol{\mu_X} + \frac{1}{2}\|\boldsymbol{q}-\boldsymbol{\mu_X}\|^2 + \frac{1}{2}(M^2 - \|\boldsymbol{\mu_X}\|^2) \\
&= -\boldsymbol{q}^\top\boldsymbol{x}_k + \frac{1}{2}\|\boldsymbol{q}\|^2 + \frac{1}{2}\underbrace{M^2}_{\geq\|\boldsymbol{x}_k\|^2} \geq \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{q}\|^2 \geq 0.
\end{aligned}$$

The zero value of energy is attained when $\boldsymbol{X} = \mathbf{1}\boldsymbol{q}^\top$ (all patterns are equal to the query), in which case $\boldsymbol{\mu_X} = \boldsymbol{q}$, $M = \|\boldsymbol{q}\| = \|\boldsymbol{\mu_X}\|$, and we get $E_{\text{convex}}(\boldsymbol{q}) = E_{\text{concave}}(\boldsymbol{q}) = 0$.

Now we prove the two upper bounds. For that, note that, for any $\boldsymbol{y} \in \triangle_N$, we have $0 \leq L_\Omega(\boldsymbol{\theta}, \boldsymbol{y}) = L_\Omega(\boldsymbol{\theta}, \mathbf{1}/N) - \Omega(\mathbf{1}/N) + \Omega(\boldsymbol{y}) - (\boldsymbol{y} - \mathbf{1}/N)^\top\boldsymbol{\theta} \leq L_\Omega(\boldsymbol{\theta}, \mathbf{1}/N) - \Omega(\mathbf{1}/N) - (\boldsymbol{y} - \mathbf{1}/N)^\top\boldsymbol{\theta}$, due to the assumptions 1–3 which ensure Ω is non-positive. That is, $L_\Omega(\boldsymbol{\theta}, \mathbf{1}/N) \geq \Omega(\mathbf{1}/N) + (\boldsymbol{y} - \mathbf{1}/N)^\top\boldsymbol{\theta}$. Therefore, with $\boldsymbol{q} = \boldsymbol{X}^\top\boldsymbol{y}$, we get

$$E_{\text{concave}}(\boldsymbol{q}) \leq -\beta^{-1}\Omega(\mathbf{1}/N) - \boldsymbol{y}^\top\boldsymbol{X}\boldsymbol{q} + \boldsymbol{q}^\top\boldsymbol{\mu_X} = -\beta^{-1}\Omega(\mathbf{1}/N) - \|\boldsymbol{q}\|^2 + \boldsymbol{q}^\top\boldsymbol{\mu_X},$$

and $E(\boldsymbol{q}) = E_{\text{concave}}(\boldsymbol{q}) + E_{\text{convex}}(\boldsymbol{q}) \leq -\beta^{-1}\Omega(\mathbf{1}/N) - \|\boldsymbol{q}\|^2 + \boldsymbol{q}^\top\boldsymbol{\mu_X} + \frac{1}{2}\|\boldsymbol{q} - \boldsymbol{\mu_X}\|^2 + \frac{1}{2}(M^2 - \|\boldsymbol{\mu_X}\|^2) = -\beta^{-1}\Omega(\mathbf{1}/N) - \frac{1}{2}\|\boldsymbol{q}\|^2 + \frac{1}{2}M^2 \leq -\beta^{-1}\Omega(\mathbf{1}/N) + \frac{1}{2}M^2$.

To show the second upper bound, use the fact that $E_{\text{concave}}(\boldsymbol{q}) \leq 0$, which leads to $E(\boldsymbol{q}) \leq E_{\text{convex}}(\boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{q} - \boldsymbol{\mu_X}\|^2 + \frac{1}{2}(M^2 - \|\boldsymbol{\mu_X}\|^2) = \frac{1}{2}\|\boldsymbol{q}\|^2 - \boldsymbol{q}^\top\boldsymbol{\mu_X} + \frac{1}{2}M^2$. Note that $\|\boldsymbol{q}\| = \|\boldsymbol{X}^\top\boldsymbol{y}\| \leq \sum_i y_i\|\boldsymbol{x}_i\| \leq M$ and that, from the Cauchy-Schwarz inequality, we have $-\boldsymbol{q}^\top\boldsymbol{\mu_X} \leq \|\boldsymbol{\mu_X}\|\|\boldsymbol{q}\| \leq M^2$. Therefore, we obtain $E(\boldsymbol{q}) \leq \frac{1}{2}\|\boldsymbol{q}\|^2 - \boldsymbol{q}^\top\boldsymbol{\mu_X} + \frac{1}{2}M^2 \leq \frac{1}{2}M^2 + M^2 + \frac{1}{2}M^2 = 2M^2$.

B.4 Proof of Proposition 9

A stationary point is a solution of the equation $-\nabla E_{\text{concave}}(\boldsymbol{q}) = \nabla E_{\text{convex}}(\boldsymbol{q})$. Using the expression for gradients (35), this is equivalent to $\boldsymbol{q} = \boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q})$. If $\boldsymbol{x}_i = \boldsymbol{X}^\top \boldsymbol{e}_i$ is not a convex combination of the other memory patterns, $\boldsymbol{x}_i$ is a stationary point iff $\hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{x}_i) = \boldsymbol{e}_i$. We now use the margin property of sparse transformations (19), according to which the latter is equivalent to $\beta \boldsymbol{x}_i^\top \boldsymbol{x}_i - \max_{j \neq i} \beta \boldsymbol{x}_i^\top \boldsymbol{x}_j \geq m$. Noting that the left hand side equals $\beta \Delta_i$ leads to the desired result.

If the initial query satisfies $\boldsymbol{q}^{(0)\top}(\boldsymbol{x}_i - \boldsymbol{x_j}) \geq \frac{m}{\beta}$ for all $j \neq i$, we have again from the margin property that $\hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q}^{(0)}) = \boldsymbol{e}_i$, which combined to the previous claim ensures convergence in one step to $\boldsymbol{x}_i$. Finally, note that, if $\boldsymbol{q}^{(0)}$ is ϵ-close to $\boldsymbol{x}_i$, we have $\boldsymbol{q}^{(0)} = \boldsymbol{x}_i + \epsilon \boldsymbol{r}$ for some vector $\boldsymbol{r}$ with $\|\boldsymbol{r}\| = 1$. Therefore, we have

$$(\boldsymbol{q}^{(0)})^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) = (\boldsymbol{x}_i + \epsilon \boldsymbol{r})^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq \Delta_i + \epsilon \boldsymbol{r}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq \Delta_i - \epsilon \underbrace{\|\boldsymbol{r}\|}_{=1} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|,$$

where we invoked the Cauchy-Schwarz inequality in the last step. Since the patterns are normalized (with norm M),[8] we have from the triangle inequality that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \|\boldsymbol{x}_i\| + \|\boldsymbol{x}_j\| = 2M$; using the assumption that $\Delta_i \geq \frac{m}{\beta} + 2M\epsilon$, we obtain $\boldsymbol{q}^{(0)\top}(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq \frac{m}{\beta}$, which from the previous points ensures convergence to $\boldsymbol{x}_i$ in one iteration.

B.5 Proof of Proposition 10

For the first statement, we follow a similar argument as the one made by Ramsauer et al. (2021) in the proof of their Theorem A.3—however their proof has a mistake, which we correct here.[9] Given a separation angle $\alpha_{\min}$, we lower bound the number of patterns N we can place in the sphere separated by at least this angle. Estimating this quantity is an important open problem in combinatorics, related to determining the size of spherical codes (of which kissing numbers are a particular case; Conway and Sloane 2013). We invoke a lower bound due to Chabauty (1953), Shannon (1959), and Wyner (1965) (see also Jenssen et al. (2018) for a tighter bound), who show that $N \geq (1 + o(1))\sqrt{2\pi D} \frac{\cos \alpha_{\min}}{(\sin \alpha_{\min})^{D-1}}$. For $\alpha_{\min} = \frac{\pi}{3}$, which corresponds to the kissing number problem, we obtain the bound $N \geq (1 + o(1))\sqrt{3\pi D/8} (2/\sqrt{3})^D = \mathcal{O}\left((2/\sqrt{3})^D\right)$. In this scenario, we have $\Delta_i = M^2(1 - \cos \alpha_{\min})$ by the definition of Δ_i. From Proposition 9, we have exact retrieval under ϵ-perturbations if $\Delta_i \geq m\beta^{-1} + 2M\epsilon$. Combining the two expressions, we obtain $\epsilon \leq \frac{M}{2}(1 - \cos \alpha_{\min}) - \frac{m}{2\beta M}$. Setting $\alpha_{\min} = \frac{\pi}{3}$, we obtain $\epsilon \leq \frac{M}{2}\left(1 - \frac{1}{2}\right) - \frac{m}{2\beta M} = \frac{M}{4} - \frac{m}{2\beta M}$. For the right hand side to be positive, we must have $M^2 > 2m/\beta$.

Assume now patterns are placed uniformly at random in the sphere. From Brauchart et al. (2018) we have, for any $\delta > 0$:

$$P(N^{\frac{2}{D-1}}\alpha_{\min} \geq \delta) \geq 1 - \frac{\kappa_{D-1}}{2}\delta^{D-1}, \quad \text{with} \quad \kappa_D := \frac{1}{D\sqrt{\pi}}\frac{\Gamma((D+1)/2)}{\Gamma(D/2)}. \tag{40}$$

---

8. In fact, the result still holds if patterns are not normalized but have their norm upper bounded by M, i.e., if they lie within a ball of radius M and not necessarily on the sphere.

9. Concretely, Ramsauer et al. (2021) claim that given a separation angle $\alpha_{\min}$, we can place $N = (2\pi/\alpha_{\min})^{D-1}$ patterns equidistant on the sphere, but this is not correct.

Given our failure probability p, we need to have $P(M^2(1 - \cos\alpha_{\min}) \geq m\beta^{-1} + 2M\epsilon) \geq 1 - p$, which is equivalent to

$$P\left\{N^{\frac{2}{D-1}}\alpha_{\min} \geq N^{\frac{2}{D-1}}\arccos\left(1 - \frac{m}{\beta M^2} - \frac{2\epsilon}{M}\right)\right\} \geq 1 - p. \tag{41}$$

Therefore, we set $p = \frac{\kappa_{D-1}}{2}N^2\left[\arccos\left(1 - \frac{m}{\beta M^2} - \frac{2\epsilon}{M}\right)\right]^{D-1}$. Choosing $N = \sqrt{\frac{2p}{\kappa_{D-1}}}\zeta^{\frac{D-1}{2}}$ patterns for some $\zeta > 1$, we obtain $1 = \left[\zeta\arccos\left(1 - \frac{m}{\beta M^2} - \frac{2\epsilon}{M}\right)\right]^{D-1}$. Therefore the failure rate p is attainable provided the perturbation error is

$$\epsilon \leq \frac{M}{2}\left(1 - \cos\frac{1}{\zeta}\right) - \frac{m}{2\beta M}. \tag{42}$$

For the right hand side to be positive, we must have $\cos\frac{1}{\zeta} < 1 - \frac{m}{\beta M^2}$, i.e., $\zeta < \frac{1}{\arccos\left(1 - \frac{m}{\beta M^2}\right)}$.

B.6 Proof of Proposition 13

The first statement in the proposition is stated and proved by Blondel et al. (2020) as a corollary of their Proposition 8. We prove here a more general version, which includes the second statement as a novel result. Using Blondel et al. (2020, Proposition 8), we have that the structured margin of L_Ω is given by the following expression,

$$m = \sup_{\boldsymbol{y}\in\mathcal{Y},\,\boldsymbol{\mu}\in\text{conv}(\mathcal{Y})} \frac{\Omega(\boldsymbol{y}) - \Omega(\boldsymbol{\mu})}{r^2 - \boldsymbol{\mu}^\top\boldsymbol{y}},$$

if the supremum exists. For SparseMAP, using $\Omega(\boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{\mu}_V\|^2 = \frac{1}{2}\|\boldsymbol{\mu}\|^2 - \frac{1}{2}\|\boldsymbol{\mu}_F\|^2$ for any $\boldsymbol{\mu} \in \text{conv}(\mathcal{Y})$, and using the fact that $\|\boldsymbol{y}\| = r$ for any $\boldsymbol{y} \in \mathcal{Y}$, we obtain:

$$m = \sup_{\boldsymbol{y}\in\mathcal{Y},\,\boldsymbol{\mu}\in\text{conv}(\mathcal{Y})} \frac{\frac{1}{2}r^2 - \frac{1}{2}\|\boldsymbol{\mu}\|^2 + \frac{1}{2}\|\boldsymbol{\mu}_F\|^2 - \frac{1}{2}r_F^2}{\boldsymbol{y}^\top(\boldsymbol{y} - \boldsymbol{\mu})} \leq^{(\dagger)} \sup_{\boldsymbol{y}\in\mathcal{Y},\,\boldsymbol{\mu}\in\text{conv}(\mathcal{Y})} \frac{\frac{1}{2}r^2 - \frac{1}{2}\|\boldsymbol{\mu}\|^2}{\boldsymbol{y}^\top(\boldsymbol{y} - \boldsymbol{\mu})}$$

$$= 1 - \inf_{\boldsymbol{y}\in\mathcal{Y},\,\boldsymbol{\mu}\in\text{conv}(\mathcal{Y})} \frac{\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}\|^2}{\boldsymbol{y}^\top(\boldsymbol{y} - \boldsymbol{\mu})} \leq^{(\ddagger)} 1,$$

where the inequality $(\dagger)$ follows from the convexity of $\frac{1}{2}\|\cdot\|^2$, which implies that $\frac{1}{2}\|\boldsymbol{\mu}_F\|^2 \leq \frac{1}{2}\|\boldsymbol{y}_F\|^2 = \frac{1}{2}r_F^2$; and the inequality $(\ddagger)$ follows from the fact that both the numerator and denominator in the second term are non-negative, the latter due to the Cauchy-Schwartz inequality and the fact that $\|\boldsymbol{\mu}\| \leq r$. This proves the second part of Proposition 13.

To prove the first part, note first that, if there are no higher order interactions, then $r_F = 0$ and $\boldsymbol{\mu}_F$ is an "empty vector", which implies that $(\dagger)$ is an equality. We prove now that, in this case, $(\ddagger)$ is also an equality, which implies that $m = 1$. We do that by showing that, for any $\boldsymbol{y} \in \mathcal{Y}$, we have $\inf_{\boldsymbol{\mu}\in\text{conv}(\mathcal{Y})} \frac{\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}\|^2}{\boldsymbol{y}^\top(\boldsymbol{y} - \boldsymbol{\mu})} = 0$. Indeed, choosing $\boldsymbol{\mu} = t\boldsymbol{y}' + (1 - t)\boldsymbol{y}$ for an arbitrary $\boldsymbol{y}' \in \mathcal{Y} \setminus \{\boldsymbol{y}\}$, and letting $t \to 0^+$, we obtain $\frac{\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}\|^2}{\boldsymbol{y}^\top(\boldsymbol{y} - \boldsymbol{\mu})} = \frac{\frac{t}{2}\|\boldsymbol{y} - \boldsymbol{y}'\|^2}{\boldsymbol{y}^\top(\boldsymbol{y} - \boldsymbol{y}')} \to 0$.

B.7 Proof of Proposition 14

A point $\boldsymbol{q}$ is stationary iff it satisfies $\boldsymbol{q} = \boldsymbol{X}^\top \hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q})$. Therefore, $\boldsymbol{X}^\top \boldsymbol{y}_i$ is guaranteed to be a stationary point if $\hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{y}_i) = \boldsymbol{y}_i$.[10] By assumption, we have $\beta \Delta_i \geq \frac{1}{2} D_i^2 \geq \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$ for all j. Since $\Delta_i \leq \boldsymbol{y}_i^\top \boldsymbol{X} \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j)$ by definition, this implies $\beta \boldsymbol{y}_i^\top \boldsymbol{X} \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \geq \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$. Since SparseMAP has a margin $m \leq 1$, we recognize that the latter inequality is a margin condition (Def. 12), which implies zero loss, i.e., $\hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{y}_i) = \boldsymbol{y}_i$, as desired.

If the initial query satisfies $\boldsymbol{q}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \geq \frac{D_i^2}{2\beta}$ for all $j \neq i$, we have again from the margin property that $\hat{\boldsymbol{y}}_\Omega(\beta \boldsymbol{X} \boldsymbol{q}) = \boldsymbol{y}_i$, which ensures convergence in one step to $\boldsymbol{X}^\top \boldsymbol{y}_i$.

If $\boldsymbol{q}$ is ϵ-close to $\boldsymbol{X}^\top \boldsymbol{y}_i$, then $\boldsymbol{q} = \boldsymbol{X}^\top \boldsymbol{y}_i + \epsilon \boldsymbol{r}$ for some vector $\boldsymbol{r}$ with $\|\boldsymbol{r}\| = 1$. Therefore,

$$\boldsymbol{q}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) = (\boldsymbol{X}^\top \boldsymbol{y}_i + \epsilon \boldsymbol{r})^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \geq \Delta_i + \epsilon \boldsymbol{r}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j). \tag{43}$$

We now bound $-\boldsymbol{r}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j)$ in two possible ways. Using the Cauchy-Schwarz inequality, we have $-\boldsymbol{r}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \leq \|\boldsymbol{X} \boldsymbol{r}\|\|\boldsymbol{y}_i - \boldsymbol{y}_j\| \leq \sigma_{\max}(\boldsymbol{X}) D_i$, where $\sigma_{\max}(\boldsymbol{X})$ is the largest singular value of $\boldsymbol{X}$ (its spectral norm). On the other hand, denoting $R_i := \max_j \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_1$, we can also use Hölder's inequality to obtain $-\boldsymbol{r}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \leq \|\boldsymbol{X} \boldsymbol{r}\|_\infty \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_1 \leq M R_i$, where we used the fact that $\|\boldsymbol{X} \boldsymbol{r}\|_\infty = \max_k |\boldsymbol{x}_k^\top \boldsymbol{r}| \leq \|\boldsymbol{x}_k\|\|\boldsymbol{r}\| = M$. Combining the two inequalities, we obtain $\boldsymbol{q}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \geq \Delta_i - \epsilon \min\{\sigma_{\max}(\boldsymbol{X}) D_i, M R_i\}$. Using the assumption that $\Delta_i \geq \frac{D_i^2}{2\beta} + \epsilon \min\{\sigma_{\max}(\boldsymbol{X}) D_i, M R_i\}$, we obtain $\boldsymbol{q}^\top \boldsymbol{X}^\top (\boldsymbol{y}_i - \boldsymbol{y}_j) \geq \frac{D_i^2}{2\beta}$, which from the previous points ensures convergence to $\boldsymbol{X}^\top \boldsymbol{y}_i$ in one iteration. The result follows by noting that, since $\mathcal{Y} \subseteq \{0, 1\}^D$, we have $R_i = D_i^2$.

Appendix C. Additional Experiments and Experimental Details

C.1 Memory Retrieval

Further insights into Hopfield-Fenchel-Young network variants are provided in Figure 14. The figure reveals that normalization excels across a variable number of memories.

C.2 MNIST K-MIL

For K-MIL, we created 4 datasets by grouping the MNIST examples into bags, for $K \in \{1, 2, 3, 5\}$. A bag is positive if it contains at least K targets, where the target is the number "9" (we chose "9" as it can be easily misunderstood with "7" or "4"). The embedding architecture is the same as Ilse et al. (2018), but instead of attention-based pooling, we use our α-entmax pooling, with $\alpha = 1$ mirroring the pooling method in Ramsauer et al. (2021), and $\alpha = 2$ corresponding to the pooling in Hu et al. (2023). Additionally, we incorporate α-normmax pooling and SparseMAP pooling with k-subsets. Further details of the K-MIL datasets are shown in Table 7.

We train the models for 5 different random seeds, where the first one is used for tuning the hyperparameters. The reported test accuracies represent the average across these seeds. We use 500 bags for testing and 500 bags for validation. The hyperparameters are tuned via grid search, where the grid space is shown in Table 8. We consider a dropout hyperparameter, commonly referred to as bag dropout, to the Hopfield matrix due to the risk of overfitting

---

10. But not necessarily "only if"—we could have $\boldsymbol{X}^\top \boldsymbol{y}_i$ in the convex hull of the other pattern associations.
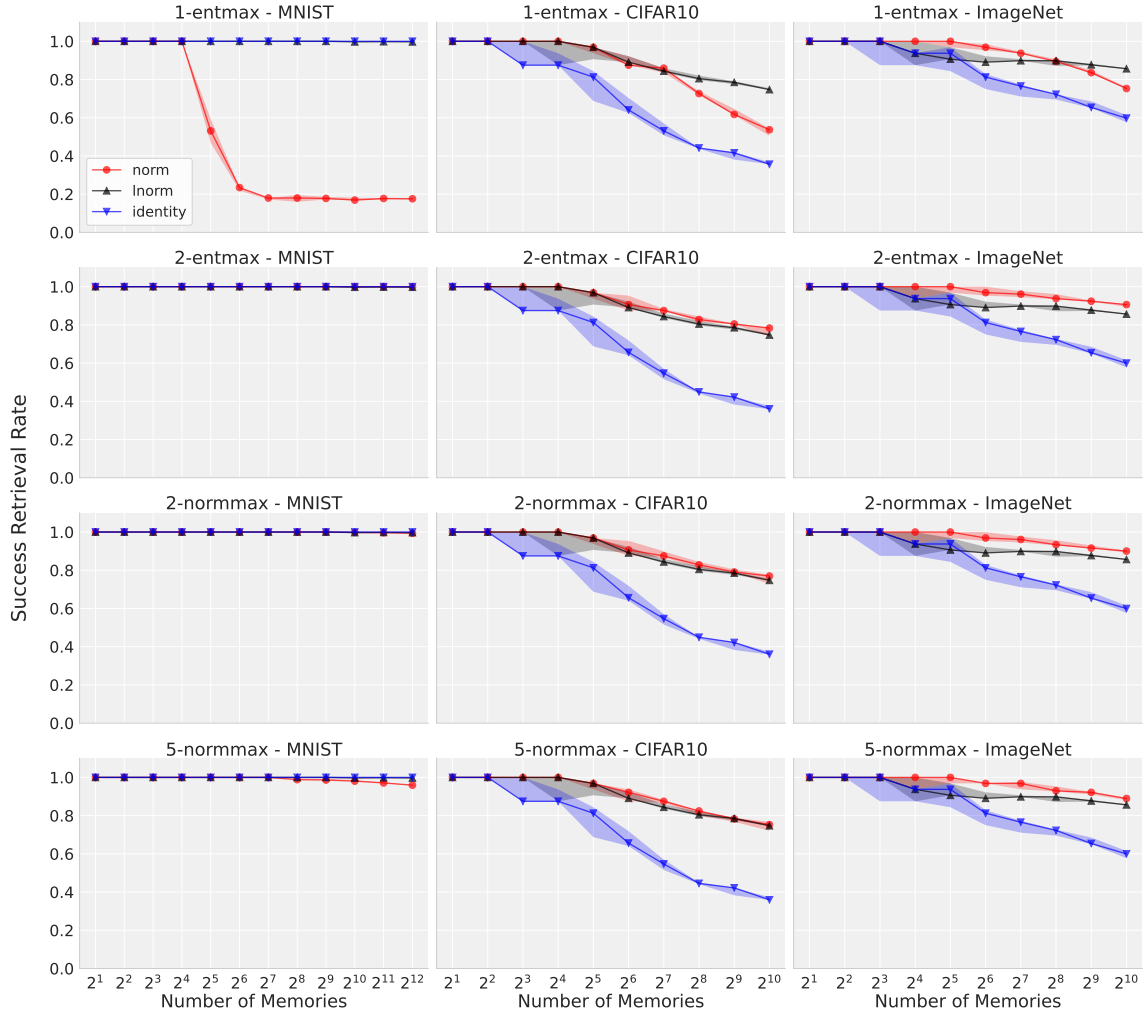
Figure 14: Memory capacity for different numbers of stored memories for $\beta = 1$ and for different $\hat{\boldsymbol{y}}_\Omega$ and $\hat{\boldsymbol{y}}_\Psi$.

(as done by Ramsauer et al. (2021)). All models were trained for 50 epochs. We incorporated an early-stopping mechanism, with patience 5, that selects the optimal checkpoint based on performance on the validation set.

C.3 MIL benchmarks

The MIL benchmark datasets (Fox, Tiger and Elephant) comprise preprocessed and segmented color images sourced from the Corel dataset Ilse et al. (2018). Each image is composed of distinct segments or blobs, each defined by descriptors such as color, texture, and shape. The datasets include 100 positive and 100 negative example images, with the negative ones randomly selected from a pool of photos featuring various other animals.

Table 7: Dataset sample details for the MNIST K-MIL experiment. The size L_i of the ith bag is determined through $L_i = \max\{K, L'_i\}$ where $L'_i \sim \mathcal{N}(\mu, \sigma^2)$. The number of positive instances in a bag is uniformly sampled between K and L_i for positive bags and between 0 and $K - 1$ for negative bags.

Dataset	μ	σ	Features	Pos. training bags	Neg. training bags
MNIST, $K = 1$	10	1	28×28	1000	1000
MNIST, $K = 2$	11	2	28×28	1000	1000
MNIST, $K = 3$	12	3	28×28	1000	1000
MNIST, $K = 5$	14	5	28×28	1000	1000

Table 8: Hyperparameter space for the MNIST MIL experiment. Hidden size is the dimension of keys and queries and γ is a parameter of the exponential learning rate scheduler (Li and Arora, 2020).

Parameter	Range
learning rate	$\{10^{-5}, 10^{-6}\}$
γ	$\{0.98 , 0.96\}$
hidden size	$\{16, 64\}$
number of heads	$\{8, 16\}$
β	$\{0.25, 0.5, 1.0, 2.0, 4.0, 8.0\}$
bag dropout	$\{0.0, 0.75\}$

The HopfieldPooling layers (α-entmax; α-normmax; SparseMAP, k-subsets) take as input a collection of embedded instances, along with a trainable yet constant query. This query pattern is used for the purpose of averaging class-indicative instances, thereby facilitating the compression of bags of variable sizes into a consistent representation. This compression is important for effectively discriminating between different bags. To fine-tune the model, a manual hyperparameter search was conducted on a validation set.

In our approach to tasks involving Elephant, Fox and Tiger, we followed a similar architecture as (Ramsauer et al., 2021):

1. The first two layers are fully connected linear embedding layers with ReLU activation.

2. The output of the second layer serves as the input for the HopfieldPooling layer, where the pooling operation is executed.

3. Subsequently, we employ a single layer as the final linear output layer for classification with a sigmoid as the classifier.

During the hyperparameter search, various configurations were tested, including different hidden layer widths and learning rates. Particular attention was given to the hyperparameters of the HopfieldPooling layers, such as the number of heads, head dimension, and the inverse temperature β. To avoid overfitting, bag dropout (dropout at the attention weights) was

Table 9: Hyperparameter space for the MIL benchmark experiments. Hidden size is the space in which keys and queries are associated and γ is a parameter of the exponential learning rate scheduler.

Parameter	Range
learning rate	$\{10^{-3}, 10^{-5}\}$
γ	$\{0.98, 0.96\}$
embedding dimensions	$\{32, 128\}$
embedding layers	$\{2\}$
hidden size	$\{32, 64\}$
number of heads	$\{12\}$
β	$\{0.1, 1, 10\}$
bag dropout	$\{0.0, 0.75\}$

implemented as the chosen regularization technique. All hyperparameters tested are shown in Table 9.

We trained for 50 epochs with early stopping with patience 5, using the Adam optimizer Loshchilov and Hutter (2017) with exponential learning rate decay. Model validation was conducted through a 10-fold nested cross-validation, repeated five times with different data splits where the first seed is used for hyperparameter tuning. The reported test ROC AUC scores represent the average across these repetitions.

References

S.-i. Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, 100(11):1197–1206, 1972.

S.-I. Amari and K. Maginu. Statistical neurodynamics of associative memory. *Neural Networks*, 1(1):63–73, 1988.

J. M. Amigó, S. G. Balogh, and S. Hernández. A brief review of generalized entropies. *Entropy*, 20(11):813, 2018. doi: 10.3390/e20110813.

D. J. Amit, H. Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Physical Review A*, 32(2):1007, 1985a.

D. J. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55(14):1530, 1985b.

B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

J. R. Anderson and G. H. Bower. Recognition and retrieval processes in free recall. *Psychological Review*, 79:97–123, 1972.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

J. Bastings, W. Aziz, and I. Titov. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 2963–2977, 2019.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

M. Blondel, A. F. Martins, and V. Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(1):1314–1382, 2020.

V. Boboeva, A. Pezzotta, and C. Clopath. Free recall scaling laws and short-term memory effects in a latching attractor network. *Proceedings of the National Academy of Sciences of the United States of America*, 2021. doi: 10.1073/pnas.2026092118.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. ISBN 978-0-521-83378-3.

J. S. Brauchart, A. B. Reznikov, E. B. Saff, I. H. Sloan, Y. G. Wang, and R. S. Womersley. Random point sets on the sphere—hole radii, covering, and separation. *Experimental Mathematics*, 27(1):62–81, 2018.

C. Chabauty. Resultats sur l'empilement de calottes egales sur une perisphere de $\mathbb{R}^n$ et correction a un travail anterieur. *Comptes Rendus Hebdomadaires des Seances de l'Academie des Sciences*, 236(15):1462–1464, 1953.

J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, 2013.

G. M. Correia, V. Niculae, and A. F. Martins. Adaptively sparse transformers. In *Proceedings of EMNLP-IJCNLP*, 2019.

A. Davydov, S. Jaffe, A. Singh, and F. Bullo. Retrieving k-nearest memories with modern hopfield networks. In *Associative Memory {\&} Hopfield Networks in 2023*, 2023.

M. Demircigil, J. Heusel, M. Löwe, S. Upgang, and F. Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168:288–299, 2017.

H. Eichenbaum. Memory: Organization and control. *Annual Review of Psychology*, 68:19–45, 2017. doi: 10.1146/annurev-psych-010416-044131.

W. Fenchel. On conjugate convex functions. *Canadian Journal of Mathematics*, 1:73–77, 1949.

A. S. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *Annals of Statistics*, pages 1367—-1433, 2004.

N. M. Guerreiro and A. F. T. Martins. Spectra: Sparse structured text rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6534–6550, 2021.

J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the theory of neural computation*, volume 1. Basic Books, 1991.

B. Hoover, Y. Liang, B. Pham, R. Panda, H. Strobelt, D. H. Chau, M. J. Zaki, and D. Krotov. Energy transformer. In *Advances in Neural Information Processing Systems*, 2023.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

R. Horst and N. V. Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

J. Y.-C. Hu, D. Yang, D. Wu, C. Xu, B.-Y. Chen, and H. Liu. On sparse modern hopfield model. In *Advances in Neural Information Processing Systems*, 2023.

J. Y.-C. Hu, B.-Y. Chen, D. Wu, F. Ruan, and H. Liu. Nonparametric modern hopfield models. *arXiv preprint arXiv:2404.03900*, 2024.

M. Ilse, J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, pages 2127–2136. PMLR, 2018.

E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925. doi: 10.1007/BF02980577.

A. Jacovi and Y. Goldberg. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310, 2021.

M. Jenssen, F. Joos, and W. Perkins. On kissing numbers and spherical codes in high dimensions. *Advances in Mathematics*, 335:307–321, 2018.

A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

D. Krotov and J. J. Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems*, 2016.

D. Krotov and J. J. Hopfield. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*, 2021.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

Q. V. Le et al. Tiny imagenet visual recognition challenge. In *CS 231N: Convolutional Neural Networks for Visual Recognition*, 2015.

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

T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, 2016.

J. Leutgeb, S. Leutgeb, M. Moser, and E. Moser. Pattern separation in the dentate gyrus and CA3 of the hippocampus. *Science*, 315(5814):961–966, 2007.

Z. Li and S. Arora. An exponential learning rate schedule for deep learning. In *International Conference on Learning Representations*, 2020.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

C. Malaviya, G. P. Ferreira, and A. F. T. Martins. Sparse and constrained attention for neural machine translation, 2018.

A. Martins and R. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of ICML*, 2016.

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

R. M. May. Patterns of species abundance and distribution. *Ecology and Evolution of Communities*, pages 81–120, 1975.

J. J. McAuley, J. Leskovec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *ICDM*, 2012.

R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh. The capacity of the hopfield associative memory. *IEEE transactions on Information Theory*, 33(4):461–482, 1987.

D. C. McNamee, K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman. Flexible modulation of sequence generation in the entorhinal–hippocampal system. *Nature Neuroscience*, 2021. doi: 10.1038/s41593-021-00831-7.

B. Millidge, T. Salvatori, Y. Song, T. Lukasiewicz, and R. Bogacz. Universal hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning*, pages 15561–15583. PMLR, 2022.

M. Naim, M. Katkov, S. Romani, and M. Tsodyks. Fundamental law of memory recall. *Physical Review Letters*, 124:018101, 2020.

K. Nakano. Associatron – a model of associative memory. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):380–388, 1972.

J. Neunuebel and J. Knierim. CA3 retrieves coherent representations from degraded input: direct evidence for CA3 pattern completion and dentate gyrus pattern separation. *Neuron*, 81(2):416–427, 2014.

T. Q. Nguyen and J. Salazar. Transformers without tears: Improving the normalization of self-attention. In *IWSLT*, 2019.

V. Niculae and M. Blondel. A regularized framework for sparse and structured neural attention. *Advances in Neural Information Processing Systems*, 30, 2017.

V. Niculae, A. F. Martins, M. Blondel, and C. Cardie. Sparsemap: Differentiable sparse structured inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

G. Palm. Neural associative memories and sparse coding. *Neural Netw*, 37:165–171, 2013.

B. Peters, V. Niculae, and A. F. Martins. Sparse sequence-to-sequence models. In *Proceedings of ACL*, 2019.

H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, T. Adler, D. Kreil, M. K. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. Hopfield networks is all you need. In *Proceedings of ICLR*, 2021.

S. Recanatesi, M. Katkov, S. Romani, and M. Tsodyks. Neural network model of memory retrieval. *Frontiers in Computational Neuroscience*, 9:149, 2015.

R. T. Rockafellar. Convex analysis. *Princeton Math. Series*, 28, 1970.

S. Santos, V. Niculae, D. C. McNamee, and A. F. Martins. Sparse and structured hopfield networks. In *International Conference on Machine Learning*, 2024.

W. Severa, O. Parekh, C. James, and J. Aimone. A combinatorial model for dentate gyrus sparse coding. *Neural Computation*, 29(1):94–117, 2017.

C. E. Shannon. Probability of error for optimal codes in a gaussian channel. *Bell System Technical Journal*, 38(3):611–656, 1959.

E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.

B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems*, 2003.

C. Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52:479–487, 1988.

D. Tse, R. Langston, M. Kakeyama, I. Bethus, P. Spooner, E. Wood, M. Witter, and R. Morris. Schemas and memory consolidation. *Science*, 316(5821):76–82, 2007. doi: 10.1126/science.1135935.

I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005.

E. Tulving. Episodic and semantic memory. In *Organization of Memory*, volume 1, pages 381–403. Academic Press, 1972.

E. Tulving. How many memory systems are there? *American Psychologist*, 40(4):385, 1985.

E. Tulving and D. M. Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, 80(5):352–373, 1973.

D. Tyulmankov, C. Fang, A. Vadaparty, and G. R. Yang. Biological learning in key-value memory networks. In *Advances in Neural Information Processing Systems*, 2021.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

J. C. Whittington, J. Warren, and T. E. Behrens. Relating transformers to models and neural representations of the hippocampal formation. In *Proceedings of ICLR*, 2021.

D. Wu, J. Y.-C. Hu, T.-Y. Hsiao, and H. Liu. Uniform memory retrieval with larger capacity for modern hopfield models. In *Forty-first International Conference on Machine Learning (ICML)*, 2024a. URL https://arxiv.org/abs/2404.03827.

D. Wu, J. Y.-C. Hu, W. Li, B.-Y. Chen, and H. Liu. STanhop: Sparse tandem hopfield model for memory-enhanced time series prediction. In *Proceedings of ICLR*, 2024b.

A. D. Wyner. Capabilities of bounded discrepancy decoding. *Bell System Technical Journal*, 44(6):1061–1122, 1965.

M. Yassa and C. Stark. Pattern separation in the hippocampus. *Trends Neurosci*, 34(10): 515–525, 2011.

A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural computation*, 15(4): 915–936, 2003.

Z. Zhang, K. Rudra, and A. Anand. Explain and predict, and then predict again. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021.