Modern Methods in Associative Memory
AM and Broader ML|AI

Parikshit Ram
IBM Research

How to think about Associative Memory Networks as a Machine Learning Model?

How to use them for standard Machine Learning tasks?

Chapter 5

Associative Memory:
A Machine Learning Model

Krotov, et al. "*Modern Methods in Associative Memory.*" 2025.

Associative Memory Networks

$$f_{\boldsymbol{\Xi}} : \mathbb{R}^D \to \mathbb{R}^D \qquad \boldsymbol{\Xi} = [\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \ldots, \boldsymbol{\xi}^K], \boldsymbol{\xi}^\mu \in \mathbb{R}^D$$

Model is parameterized with **stored patterns**

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} F\left(\beta S\left[\mathbf{v}, \boldsymbol{\xi}^\mu\right]\right)\right]$$

Energy function defines the **"model architecture"**

Monotonic function	Separation function	Similarity b/w state & memory

Krotov & Hopfield. "*Dense Associative Memories for Pattern Recognition.*" NeurIPS 2016.

Associative Memory Networks

$$\mathbf{\Xi} = [\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \ldots, \boldsymbol{\xi}^K], \boldsymbol{\xi}^\mu \in \mathbb{R}^D$$

$$E_\beta(\mathbf{v}; \mathbf{\Xi}) = -Q \left[\sum_{\mu=1}^{K} F\left(\beta S\left[\mathbf{v}, \boldsymbol{\xi}^\mu\right]\right) \right]$$

$$f_{\mathbf{\Xi}} : \mathbb{R}^D \to \mathbb{R}^D$$

$$\mathbf{v}^{(0)} \leftarrow \mathbf{x},$$

$$\mathbf{v}^{(t)} \leftarrow \mathbf{v}^{(t-1)} - \eta \nabla_{\mathbf{v}} E_\beta(\mathbf{v}^{(t-1)}; \mathbf{\Xi}), \quad t \in [\![T]\!],$$

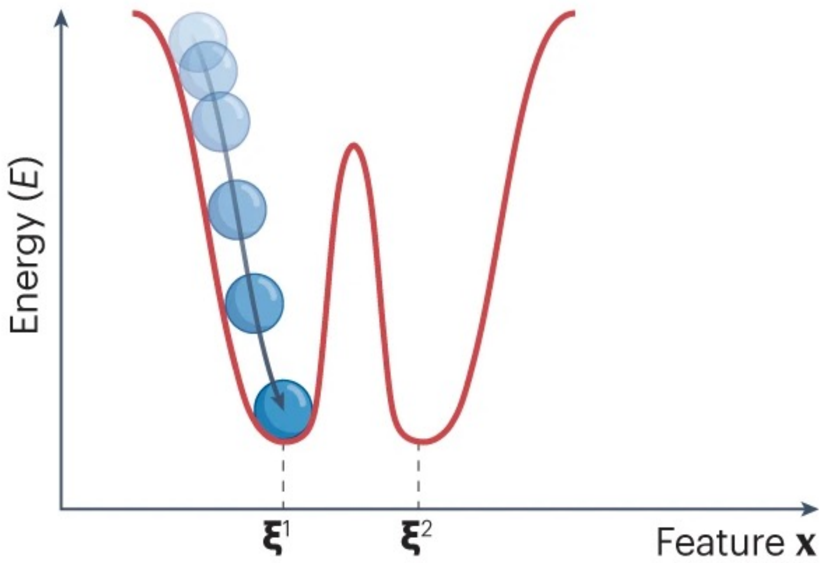
$$f_{\mathbf{\Xi}}(\mathbf{x}) \triangleq \mathbf{v}^{(T)}.$$

Inference (or the forward pass) with a *T*-layer AM network is equivalent to *T* **energy descent steps** using the energy gradient.

Krotov & Hopfield. "*Dense Associative Memories for Pattern Recognition.*" NeurIPS 2016.

Associative Memory Networks

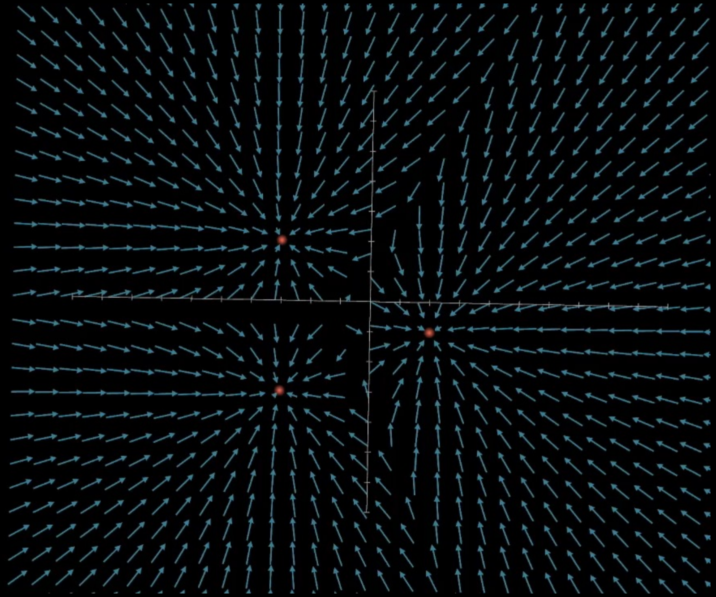
- The **energy of any state** is inversely related to the **probability** of the state
- Inference via **energy minimization** – or **likelihood maximization**



Krotov. "*A New Frontier for Hopfield networks.*" Nature Reviews Physics (2023).

Associative Memory Networks

- More memories lead to more local minima of the energy *up to a point*
- More memories – more parameters – potentially more expressive model



Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Memory Capacity

The number of memories that can form distinct local minima of the energy

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} F\left(\beta S\left[\mathbf{v}, \boldsymbol{\xi}^\mu\right]\right)\right]$$

Classical Associative Memory

$$F(z) = z^2 \Rightarrow K^{\max} \sim O(D)$$

Dense Associative Memory

$$F(z) = z^N \Rightarrow K^{\max} \sim O(D^{N-1})$$

$$F(z) = \exp(z) \Rightarrow K^{\max} \sim O(\exp(D))$$

More memory capacity -- more model expressivity

But increased computational overhead!

Krotov. "*A New Frontier for Hopfield networks.*" Nature Reviews Physics (2023).

Energy is a Kernel Sum

$$f_{\boldsymbol{\Xi}} : \mathbb{R}^D \to \mathbb{R}^D \qquad \boldsymbol{\Xi} = [\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \ldots, \boldsymbol{\xi}^K], \boldsymbol{\xi}^\mu \in \mathbb{R}^D$$

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} F\left(\beta S\left[\mathbf{v}, \boldsymbol{\xi}^\mu\right]\right)\right] = -Q\left[\sum_{\mu=1}^{K} \kappa\left(\mathbf{v}, \boldsymbol{\xi}^\mu\right)\right]$$

Kernel sum

$$F(\beta S[\mathbf{x}, \mathbf{x}']) \triangleq \kappa(\mathbf{x}, \mathbf{x}')$$

AMs and Kernel Machines

- Both **nonparametric**
- Kernel machines
  - Inference with a **single kernel sum** (usually)
- Associative Memory networks
  - AM can be **parametric**
  - Kernel sum computes energy & inference via energy descent
  - Single inference (usually) needs **multiple kernel sums**
  - Need ability to **differentiate through the kernel sums**

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} \kappa\left(\mathbf{v}, \boldsymbol{\xi}^{\mu}\right)\right]$$

Kernel sum

AMs and Kernel Machines

Kernel machines

$$\boldsymbol{x} \longrightarrow \sum_\mu \kappa(\boldsymbol{x}, \boldsymbol{\xi}^\mu) \longrightarrow \text{output}$$

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} \kappa\left(\mathbf{v}, \boldsymbol{\xi}^\mu\right)\right]$$

Kernel sum

Associative Memory networks

$$\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} - \eta \nabla_{\mathbf{v}} Q[s]$$

$$\boldsymbol{x} \longrightarrow \mathbf{v}^{(0)} \longrightarrow \circlearrowright \longrightarrow \text{output}$$

$$s = \sum_\mu \kappa(\mathbf{v}^{(t)}, \boldsymbol{\xi}^\mu)$$

AMs and Kernel Machines

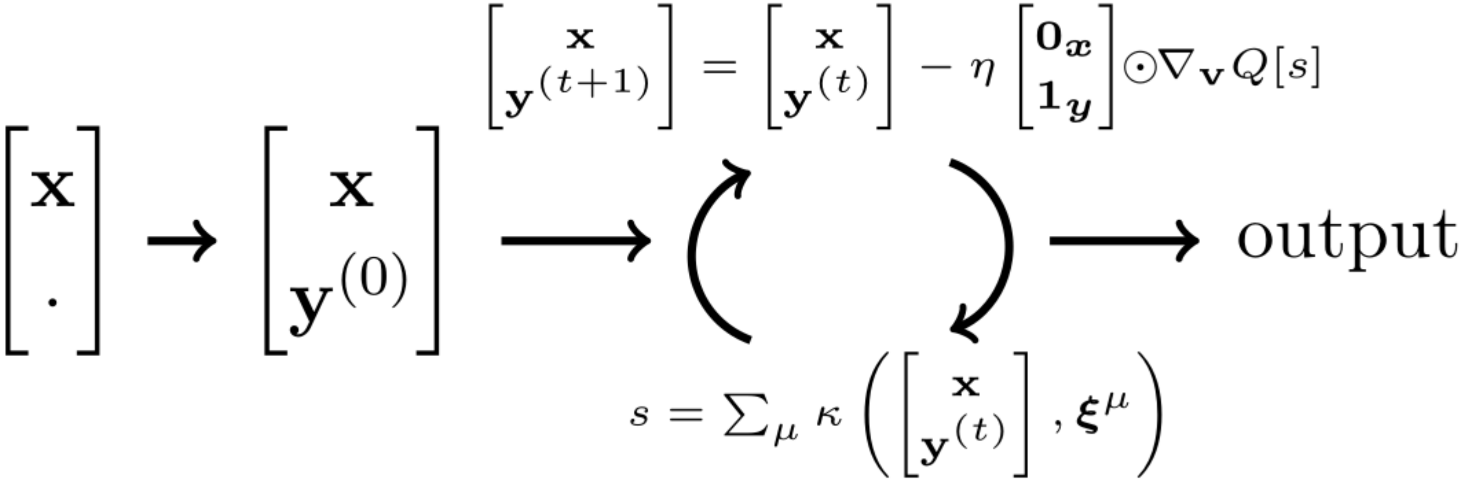
Kernel machines

$$\underbrace{\sum_{\mu} \kappa(\mathbf{v}, \boldsymbol{\xi}^{\mu})}_{\text{Density } p_{\boldsymbol{v}}(\mathbf{v})} \;\rightarrow\; \underbrace{\sum_{\mu} \mathbf{w}_{\mu} \kappa(\mathbf{v}, \boldsymbol{\xi}^{\mu})}_{\int_{\mathbf{y}} \mathbf{y}\, p_{\boldsymbol{xy}}(\mathbf{x},\mathbf{y})\, d\mathbf{y}} \;\rightarrow\; \underbrace{\frac{\sum_{\mu} \mathbf{w}_{\mu} \kappa(\mathbf{v}, \boldsymbol{\xi}^{\mu})}{\sum_{\mu} \kappa(\mathbf{v}, \boldsymbol{\xi}^{\mu})}}_{\mathbb{E}[p_{\boldsymbol{y}}(\mathbf{y}|\boldsymbol{x})] = \frac{\int_{\mathbf{y}} \mathbf{y}\, p_{\boldsymbol{xy}}(\mathbf{x},\mathbf{y})\, d\mathbf{y}}{p_{\boldsymbol{x}}(\mathbf{x})}}$$

Kernel machines can compute the **expectation of the conditional distribution.**

AMs and Kernel Machines

Associative Memory networks



AM networks can find the **modes of the conditional distribution.**

AMs and Kernel Machines

- Both **nonparametric**
- Kernel machines
  - ***O(KD)*** storage and ***O(KD)*** time per inference
- Associative Memory networks
  - ***O(KD)*** storage and ***O(KDT)*** time per inference for a ***T***-layer AM network

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} \kappa\left(\mathbf{v}, \boldsymbol{\xi}^\mu\right)\right]$$

Kernel sum

Opportunities

Can we draw inspiration from the rich literature on kernel machines?

- How can we adapt techniques for **efficient kernel machines** to AM?
- How can we utilize the various **domain-specific kernels** with **unique properties** to design **novel energy functions,** and how do these properties translate to AMs?
- What can we do with this **"mode-finding" capability** of AMs?

Are there other such connections?

How can AMs be used for other ML problems?

Opportunities

Inspiration for Kernel Machines

- Efficiency
- Mode-finding Capabilities
- Novel Energy Functions

Opportunities

Inspiration for Kernel Machines

- **Efficiency**
- Mode-finding Capabilities
- Novel Energy Functions

Memory Capacity & Storage

$$E(\mathbf{v}; \boldsymbol{\Xi}) = -\sum_{\mu=1}^{K} (\langle \mathbf{v}, \boldsymbol{\xi}^{\mu} \rangle)^2 = -\mathbf{v}\boldsymbol{T}\mathbf{v}$$

$$\boldsymbol{T} = \sum_{\mu=1}^{K} \boldsymbol{\xi}^{\mu} (\boldsymbol{\xi}^{\mu})^{\top}$$

$$O(KD) \rightarrow O(D^2)$$

Classical Hopfield energy

Disentangles number of memories from the number of model parameters needed to store them

(sub)Linear capacity

Hopfield. "*Neural Networks and Physical Systems with Emergent Collective Computational Abilities.*" PNAS 1982.

Memory Capacity & Storage

$$E_\beta(\mathbf{v};\boldsymbol{\Xi}) = -\frac{1}{\beta}\log\sum_{\mu=1}^{K}\exp\left(-{}^{\beta}\!/_{2}\,\|\mathbf{v}-\boldsymbol{\xi}^\mu\|^2\right)$$

$$\kappa(\mathbf{v},\boldsymbol{\xi}^\mu) = \exp\left(-{}^{\beta}\!/_{2}\,\|\mathbf{v}-\boldsymbol{\xi}^\mu\|^2\right)$$

Log-sum-exp energy

Need all memories to compute the energy

Exponential capacity

Demircigil et al. "*On a Model of Associative Memory with Huge Storage Capacity.*" J. Statistical Physics. 2017.

Kernel Feature Maps

MrDAM Memory representation DenseAM

$$E_\beta(\mathbf{v};\boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K}\kappa\left(\mathbf{v},\boldsymbol{\xi}^\mu\right)\right] \approx -Q\left[\sum_{\mu=1}^{K}\langle\phi(\mathbf{v}),\phi(\boldsymbol{\xi}^\mu)\rangle\right]$$

Explicit feature map

$$\phi:\mathbb{R}^D\to\mathbb{R}^Y$$

$$\kappa(\mathbf{x},\mathbf{x}')\approx\langle\phi(\mathbf{x}),\phi(\mathbf{x}')\rangle$$

$$= -Q\left[\left\langle\phi(\mathbf{v}),\sum_{\mu=1}^{K}\phi(\boldsymbol{\xi}^\mu)\right\rangle\right]$$

$$= -Q\left[\langle\phi(\mathbf{v}),\boldsymbol{T}\rangle\right]$$

DrDAM Distributed representation DenseAM

Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Distributed Memories

MrDAM energy: explicit memories

$$E_\beta(\mathbf{v};\boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K}\kappa\left(\mathbf{v},\boldsymbol{\xi}^\mu\right)\right]$$

$$\phi : \mathbb{R}^D \rightarrow \mathbb{R}^Y$$

$$\kappa(\mathbf{x},\mathbf{x}') \approx \langle\phi(\mathbf{x}),\phi(\mathbf{x}')\rangle$$

DrDAM energy: distributed memories

$$\tilde{E}_\beta(\mathbf{v};\boldsymbol{T}) = -Q\left[\langle\phi(\mathbf{v}),\boldsymbol{T}\rangle\right]$$

$$\boldsymbol{T} = \sum_{\mu=1}^{K}\phi(\boldsymbol{\xi}^\mu) \in \mathbb{R}^Y$$

$$\kappa(\mathbf{v},\boldsymbol{\xi}^\mu) = \exp\left(-\beta/2\,\|\mathbf{v}-\boldsymbol{\xi}^\mu\|^2\right)$$

Infeasible infinite dimensional feature map

Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Random Approximate Feature Maps

Bochner's Theorem:
A shift-invariant kernel is positive definite if and only it is a Fourier transform of a positive measure

Fourier transform of a positive measure

$$\kappa(\mathbf{z}, \mathbf{z}') = \kappa(\mathbf{z} - \mathbf{z}') = \int_{\mathbb{R}^D} p(\boldsymbol{\omega}) \, \exp(j \langle \boldsymbol{\omega}, \mathbf{z} - \mathbf{z}' \rangle) \, d\omega$$

Shift-invariant

Positive measure

For a positive definite shift-invariant kernel, there exists a such a positive measure

Rahimi & Recht. "*Random Features for Kernel Machines.*" NeurIPS 2007.

Random Features for Kernel Machines

$$\kappa(\mathbf{z}, \mathbf{z}') = \kappa(\mathbf{z} - \mathbf{z}') = \int_{\mathbb{R}^D} p(\boldsymbol{\omega})\, e^{j\langle \boldsymbol{\omega}, \mathbf{z} - \mathbf{z}' \rangle}\, d\omega$$

$$= \mathbb{E}_{\boldsymbol{\omega} \sim p} e^{j\langle \boldsymbol{\omega}, \mathbf{z} \rangle} \overline{e^{j\langle \boldsymbol{\omega}, \mathbf{z}' \rangle}}$$

$$= \mathbb{E}_{\boldsymbol{\omega} \sim p} \left\langle \begin{bmatrix} \cos\langle \boldsymbol{\omega}, \mathbf{z} \rangle \\ \sin\langle \boldsymbol{\omega}, \mathbf{z} \rangle \end{bmatrix}, \begin{bmatrix} \cos\langle \boldsymbol{\omega}, \mathbf{z}' \rangle \\ \sin\langle \boldsymbol{\omega}, \mathbf{z}' \rangle \end{bmatrix} \right\rangle$$

$$\approx \frac{1}{Y} \sum_{i=1}^{Y} \left\langle \begin{bmatrix} \cos\langle \boldsymbol{\omega}^i, \mathbf{z} \rangle \\ \sin\langle \boldsymbol{\omega}^i, \mathbf{z} \rangle \end{bmatrix}, \begin{bmatrix} \cos\langle \boldsymbol{\omega}^i, \mathbf{z}' \rangle \\ \sin\langle \boldsymbol{\omega}^i, \mathbf{z}' \rangle \end{bmatrix} \right\rangle$$

$$= \langle \Phi(\mathbf{z}), \Phi(\mathbf{z}') \rangle$$

$$\Phi(\mathbf{x}) = \frac{1}{\sqrt{Y}} \begin{bmatrix} \cos(\langle \boldsymbol{\omega}^1, \mathbf{x} \rangle) \\ \sin(\langle \boldsymbol{\omega}^1, \mathbf{x} \rangle) \\ \cos(\langle \boldsymbol{\omega}^2, \mathbf{x} \rangle) \\ \sin(\langle \boldsymbol{\omega}^2, \mathbf{x} \rangle) \\ \cdots \\ \cos(\langle \boldsymbol{\omega}^Y, \mathbf{x} \rangle) \\ \sin(\langle \boldsymbol{\omega}^Y, \mathbf{x} \rangle) \end{bmatrix}$$

$$\boldsymbol{\omega}^i \sim p$$

Random features if we know the positive measure

Rahimi & Recht. "*Random Features for Kernel Machines.*" NeurIPS 2007.

Random Features for Associative Memories

$$E = -\log\left(\sum_{\mu} \exp(-\frac{1}{2}||\boldsymbol{\xi}^{\mu} - \mathbf{x}||_2^2)\right)$$

Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Distributed Memories with Random Features

MrDAM energy: explicit memories

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q \left[\sum_{\mu=1}^{K} \kappa\left(\mathbf{v}, \boldsymbol{\xi}^\mu\right) \right]$$

$$O(KD) \to O(Y) = O(D/\epsilon^2)$$

Disentangles number of memories from the number of model parameters needed to store them

DrDAM energy: distributed memories

$$\tilde{E}_\beta(\mathbf{v}; \boldsymbol{T}) = -Q\left[\langle \Phi(\mathbf{v}), \boldsymbol{T} \rangle\right]$$

$$\boldsymbol{T} = \sum_{\mu=1}^{K} \Phi(\boldsymbol{\xi}^\mu) \in \mathbb{R}^Y$$

$$Y \sim O(D/\epsilon^2)$$

$$\left|\kappa(\mathbf{z}, \mathbf{z}') - \left\langle \Phi(\mathbf{z}), \Phi(\mathbf{z}') \right\rangle\right| \leq \epsilon$$

Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Approximation in Energy Descent

MrDAM energy: exact dynamics

$$E_\beta(\mathbf{v};\boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K}\kappa\left(\mathbf{v},\boldsymbol{\xi}^\mu\right)\right]$$

$$\mathbf{v}^{(t)} \leftarrow \mathbf{v}^{(t-1)} - \eta\nabla_{\mathbf{v}}E_\beta(\mathbf{v}^{(t-1)};\boldsymbol{\Xi})$$

DrDAM energy: approx dynamics

$$\tilde{E}_\beta(\mathbf{v};\boldsymbol{T}) = -Q\left[\langle\Phi(\mathbf{v}),\boldsymbol{T}\rangle\right]$$

$$\boldsymbol{T} = \sum_{\mu=1}^{K}\Phi(\boldsymbol{\xi}^\mu) \in \mathbb{R}^Y$$

$$\tilde{\mathbf{v}}^{(t)} \leftarrow \tilde{\mathbf{v}}^{(t-1)} - \eta\nabla_{\mathbf{v}}\tilde{E}_\beta(\mathbf{v}^{(t-1)};\boldsymbol{T})$$

What is the approximation in the DenseAM output?

$$\left\|\mathbf{v}^{(T)} - \tilde{\mathbf{v}}^{(T)}\right\| \leq ?$$

Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Approximation in Energy Descent

MrDAM energy: exact dynamics

$$\mathbf{v}^{(t)} \leftarrow \mathbf{v}^{(t-1)} - \eta \nabla_{\mathbf{v}} E_\beta(\mathbf{v}^{(t-1)}; \boldsymbol{\Xi})$$

DrDAM energy: approx dynamics

$$\tilde{\mathbf{v}}^{(t)} \leftarrow \tilde{\mathbf{v}}^{(t-1)} - \eta \nabla_{\mathbf{v}} \tilde{E}_\beta(\mathbf{v}^{(t-1)}; \boldsymbol{T})$$

Random feature approx bound

$$\left|\kappa(\mathbf{z}, \mathbf{z}') - \langle \Phi(\mathbf{z}), \Phi(\mathbf{z}') \rangle\right| \leq C_1 \sqrt{D/Y}$$

Sufficiently small step-size

$$\eta \leq \frac{C_2}{T(1 + 2K\beta \exp(\beta/2))}$$

Initial energy

$$\left\|\mathbf{v}^{(T)} - \tilde{\mathbf{v}}^{(T)}\right\| \leq \frac{C_1 C_2 \exp(\beta(E_\beta(\mathbf{v}; \boldsymbol{\Xi}) - 1/2))}{\beta(1 - C_2)}$$

Hoover, et al. "*Dense Associative Memories through the Lens of Random Features.*" NeurIPS 2024.

Dense Associative Memory Through the Lens of Random Features

Benjamin Hoover
IBM Research; Georgia Tech
benjamin.hoover@ibm.com

Duen Horng Chau
Georgia Tech
polo@gatech.edu

Hendrik Strobelt
IBM Research; MIT-IBM
hendrik.strobelt@ibm.com

Parikshit Ram
IBM Research
parikshit.ram@ibm.com

Dmitry Krotov
IBM Research
krotov@ibm.com

Associative Memory Tutorial
Getting started
lib
- Pokemon Sprites

tutorial
- Binary Dense Storage
- Energy Transformer
- Memory and Diffusion
- Distributed Memory

Distributed Memory

Random Features enable Dense Associative Memory to store patterns in a distributed manner across a large number of neurons.

Open in Colab

In this notebook, we demonstrate how we utilize random features to disentangle the size of the Dense Associative Memory network from the number of memories to be stored. Given the standard log-sum-exp energy $E_\beta(\cdot; \boldsymbol{\Xi})$, corresponding to a model $f_{\boldsymbol{\Xi}}$ of size $O(DK)$, we demonstrate how we can use the trigonometric random features to develop an approximate energy $\tilde{E}_\beta(\cdot; \mathbf{T})$ using a distributed representation $\mathbf{T}$ of the memories $\boldsymbol{\Xi} = \{\boldsymbol{\xi}^\mu, \mu \in [\![K]\!]\}$, thus giving us a model $f_{\mathbf{T}}$ of size $O(Y)$.

On this page
- Exact Energy Function
  - Visualizing the Energy in 2D
  - Minimizing the Energy via Gradient Descent
- Viewing Energy as a Kernel Sum
- Approximating the Energy with Random Features

Report an issue

Other Formats
- CommonMark

Opportunities

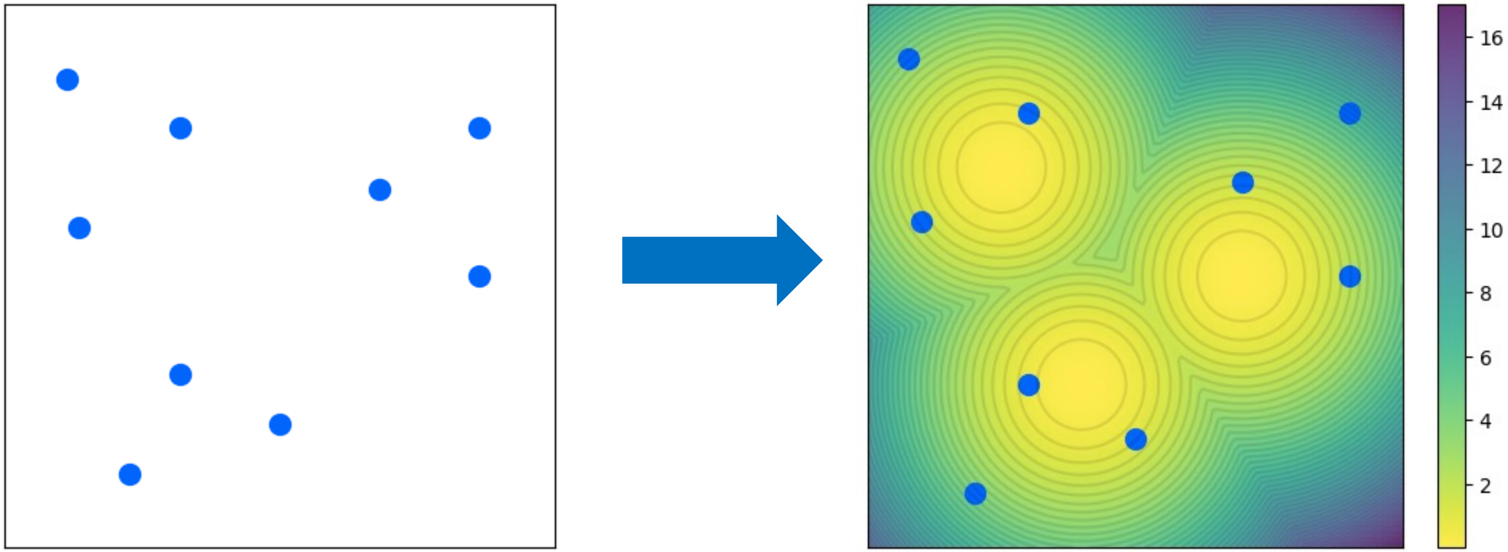
Inspiration for Kernel Machines

- Efficiency
- **Mode-finding Capabilities**
- Novel Energy Functions

Contractive Layer

- Layer operates on inputs independently
- But can exhibit collective contraction

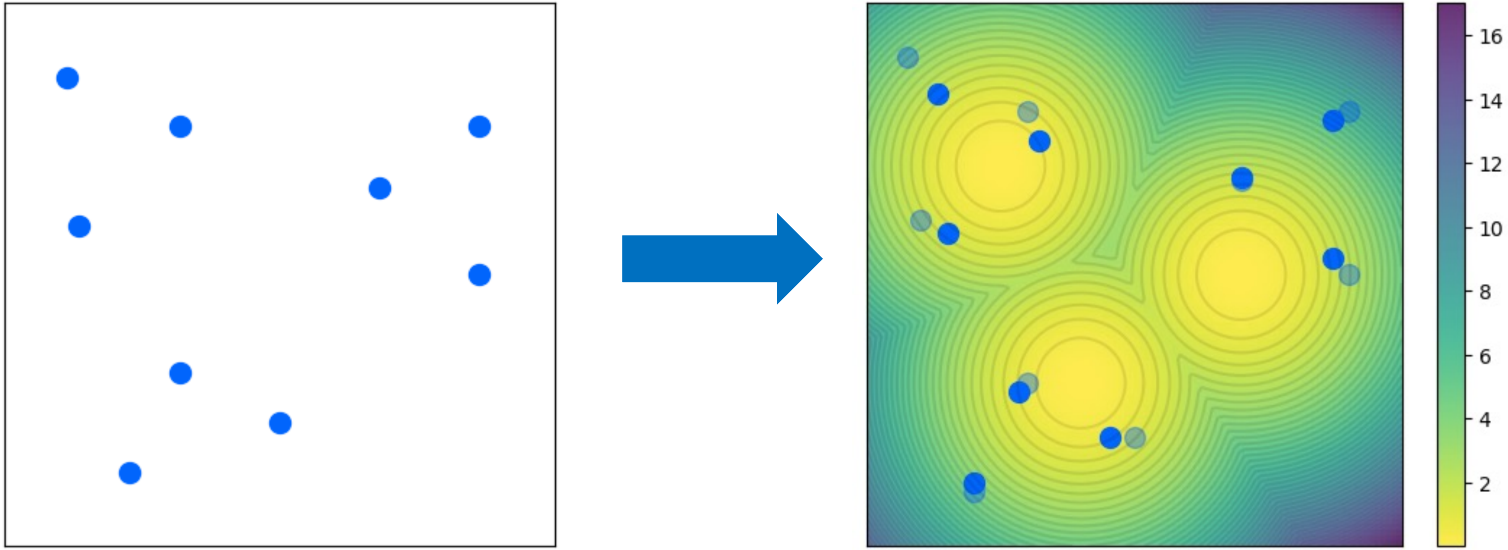
$$f_{\Xi} : \mathbb{R}^D \to \mathbb{R}^D$$



Contractive Layer

- Layer operates on inputs independently
- But can exhibit collective contraction

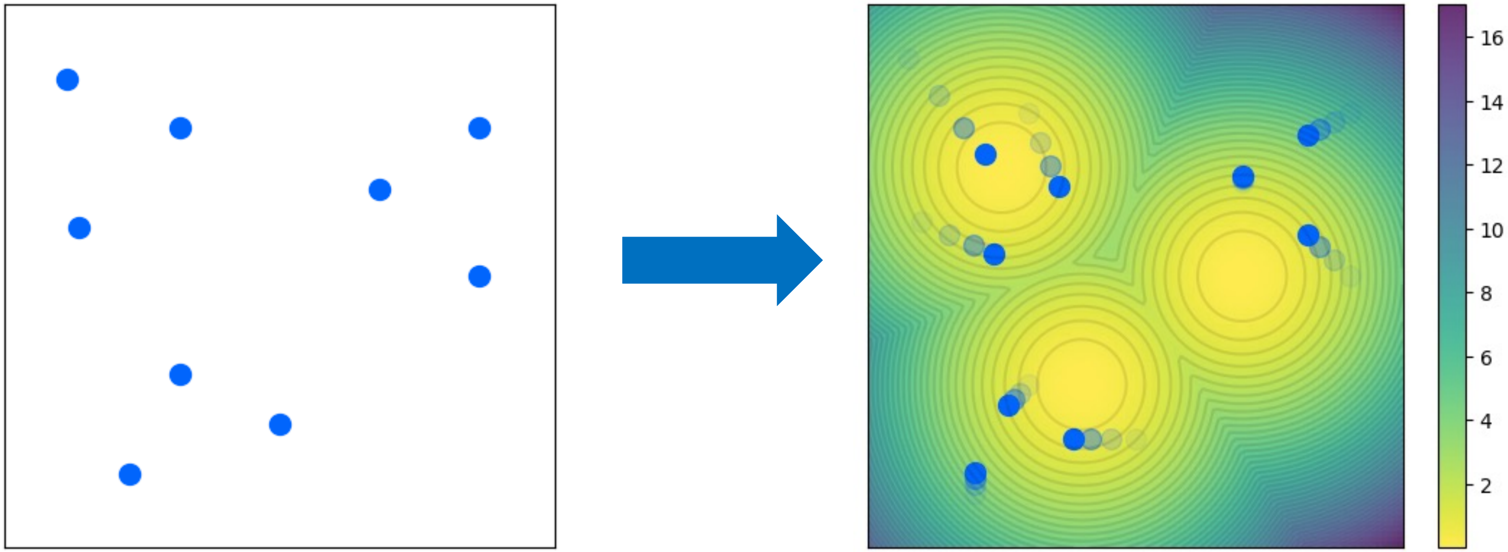
$$f_{\boldsymbol{\Xi}} : \mathbb{R}^D \rightarrow \mathbb{R}^D$$



Contractive Layer

- Layer operates on inputs independently
- But can exhibit collective contraction by **contracting towards modes**

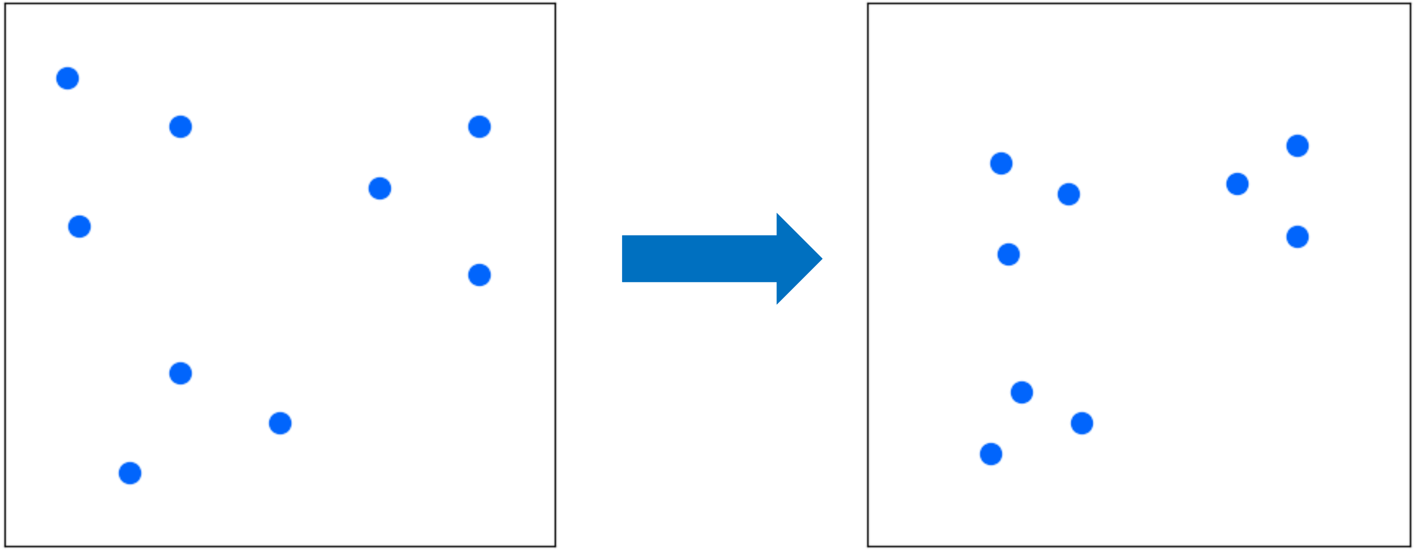
$$f_{\boldsymbol{\Xi}} : \mathbb{R}^D \to \mathbb{R}^D$$



Contractive Layer

- Layer operates on inputs independently
- But can exhibit collective contraction by **contracting towards modes**

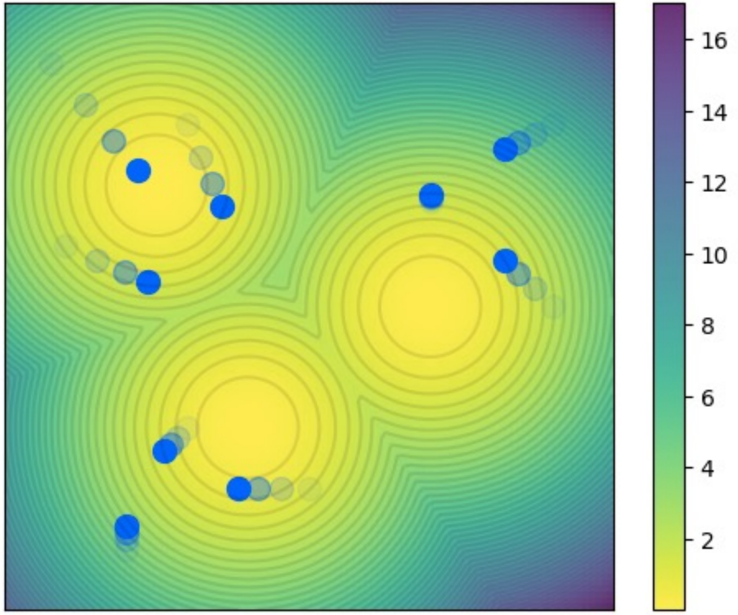
$$f_{\Xi} : \mathbb{R}^D \to \mathbb{R}^D$$



Contractive Layer

- Parameters of the AM control locations of the minima (modes)
- Thus, where the inputs contract towards

$$f_{\mathbf{\Xi}} : \mathbb{R}^D \rightarrow \mathbb{R}^D$$



Clustering

Discrete *k*-means clustering

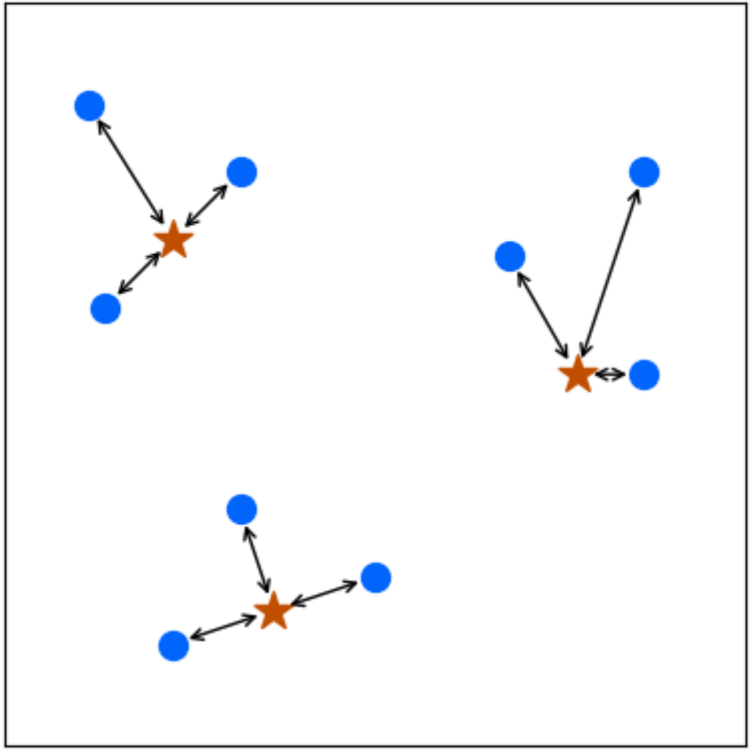
Points to cluster

$$\min_{\mathbf{c}^1,\ldots,\mathbf{c}^k \in \mathbb{R}^D} \sum_{i=1}^{m} \min_{j \in [\![k]\!]} \left\| \mathbf{x}^i - \mathbf{c}^j \right\|^2$$

Centers

Distance to closest center

Discrete optimization – we need discrete assignments to clusters



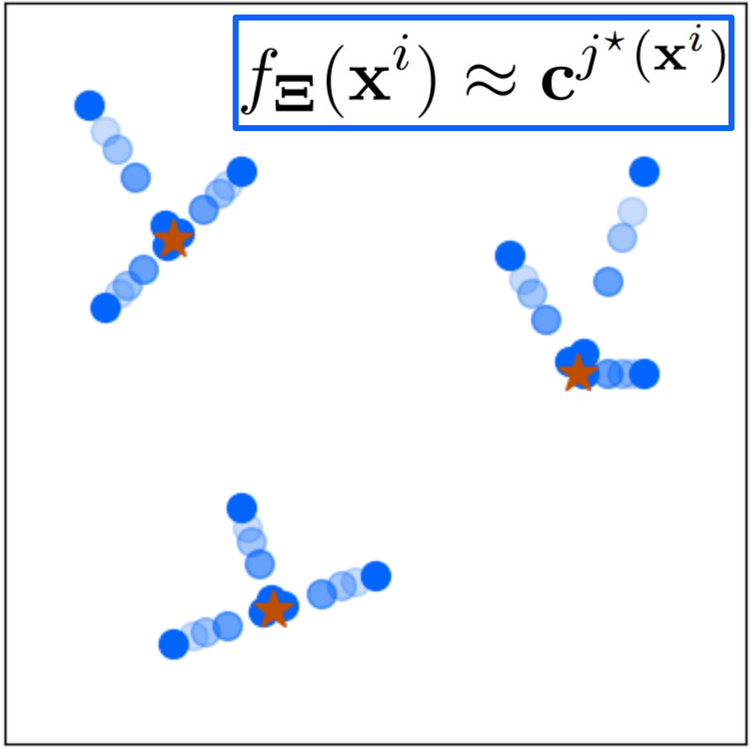
Clustering with Associative Memories

$$\min_{\mathbf{c}^1,\ldots,\mathbf{c}^k \in \mathbb{R}^D} \sum_{i=1}^{m} \min_{j \in [\![k]\!]} \left\| \mathbf{x}^i - \mathbf{c}^j \right\|^2$$

Main idea: Use contraction to emulate discrete assignment

$$\min_{\mathbf{\Xi}} \sum_{i=1}^{m} \left\| \mathbf{x}^i - f_{\mathbf{\Xi}}(\mathbf{x}^i) \right\|^2$$

Differentiable discrete clustering objective



Saha, et al. "*End-to-End Differentiable Clustering with Associative Memories.*" ICML 2023.

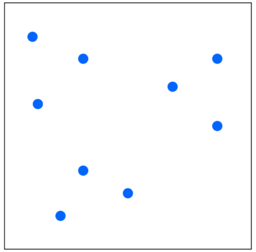
End-to-end Differentiable Clustering with Associative Memories

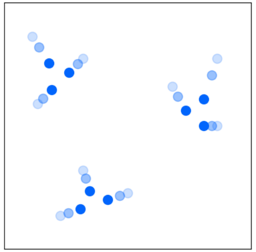
Bishwajit Saha [1] Dmitry Krotov [2] Mohammed J. Zaki [1] Parikshit Ram [2 3]

Deep Clustering with Associative Memories

Main idea: Use contraction to learn a clustered latent space

$$\boldsymbol{x} \xrightarrow{\text{encode}} e_\varphi(\boldsymbol{x}) \xrightarrow{\text{contract}} f_\Xi \circ e_\varphi(\boldsymbol{x}) \xrightarrow{\text{decode}} d_\vartheta \circ f_\Xi \circ e_\varphi(\boldsymbol{x})$$





$$\min_{\varphi, \Xi, \vartheta} \sum_{\boldsymbol{x}} \left\| \boldsymbol{x} - d_\vartheta \circ f_\Xi \circ e_\varphi(\boldsymbol{x}) \right\|^2$$

Single differentiable objective handling both **fidelity of learned representations** and the **collective clustered structure**

Saha, et al. "*Deep Clustering with Associative Memories.*" NFAM@ICLR 2025.

← Go to **ICLR 2025 Workshop NFAM** homepage

Deep Clustering with Associative Memories

Bishwajit Saha, Dmitry Krotov, Mohammed J Zaki, Parikshit Ram

Opportunities

Inspiration for Kernel Machines

- Efficiency
- Mode-finding Capabilities
- **Novel Energy Functions**

Energy Functions

$$f_{\boldsymbol{\Xi}} : \mathbb{R}^D \to \mathbb{R}^D \qquad \boldsymbol{\Xi} = [\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \dots, \boldsymbol{\xi}^K], \boldsymbol{\xi}^\mu \in \mathbb{R}^D$$

Model is parameterized with **stored patterns**

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{K} \kappa\left(\mathbf{v}, \boldsymbol{\xi}^\mu\right)\right]$$

Energy function defined by the kernel

- Gaussian kernel -- the **log-sum-exp or LSE energy**
- Kernel uses exponential separation with exponential capacity
- Is that enough to make it a "good" kernel function?

Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Insights from Density Estimation

$$\boldsymbol{\Xi} = [\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \ldots, \boldsymbol{\xi}^K], \boldsymbol{\xi}^\mu \in \mathbb{R}^D \qquad \boldsymbol{\xi}^\mu \sim p_{\text{data}}, \mu \in [\![K]\!]$$

Kernel density estimate or KDE given samples from a distribution

$$\hat{p}_h(\mathbf{v}; \boldsymbol{\Xi}) = \frac{1}{Kh} \sum_{\mu=1}^{K} \kappa\left(\frac{\mathbf{v} - \boldsymbol{\xi}^\mu}{h}\right)$$

Mean Integrated Squared Error

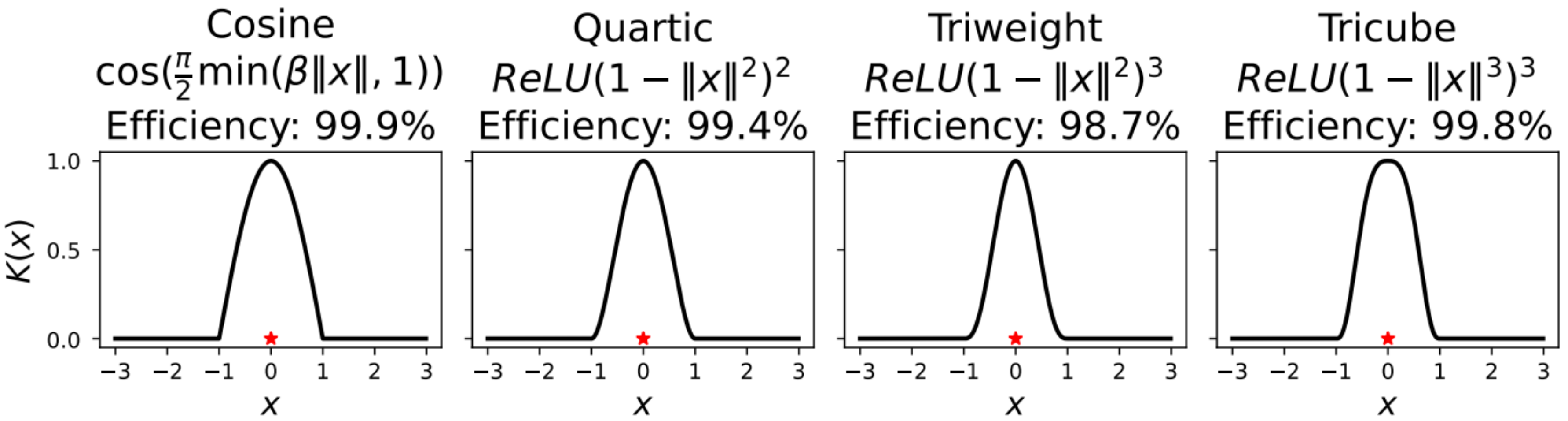
$$\text{MISE}(h) = \mathbb{E}\left[\int_v (\hat{p}_h(v; \boldsymbol{\Xi}) - p_{\text{data}}(v))^2 dv\right] \sim O\left(\left(\sqrt{\int_{\mathrm{z}} \mathrm{z}^2\, \kappa(\mathrm{z})\, d\mathrm{z}} \int_{\mathrm{z}} \kappa(\mathrm{z})^2\, d\mathrm{z}\right)^{\frac{4}{5}}\right)$$

Bias Variance

Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Kernels and their Efficiencies

$$\sqrt{\int_z z^2\,\kappa(z)\,dz \int_z \kappa(z)^2\,dz}$$



For density estimation

- Gaussian kernel is not the best; many other better
- Epanechnikov kernel known to be optimal

Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Log-Sum-Shifted-ReLU or LSR Energy

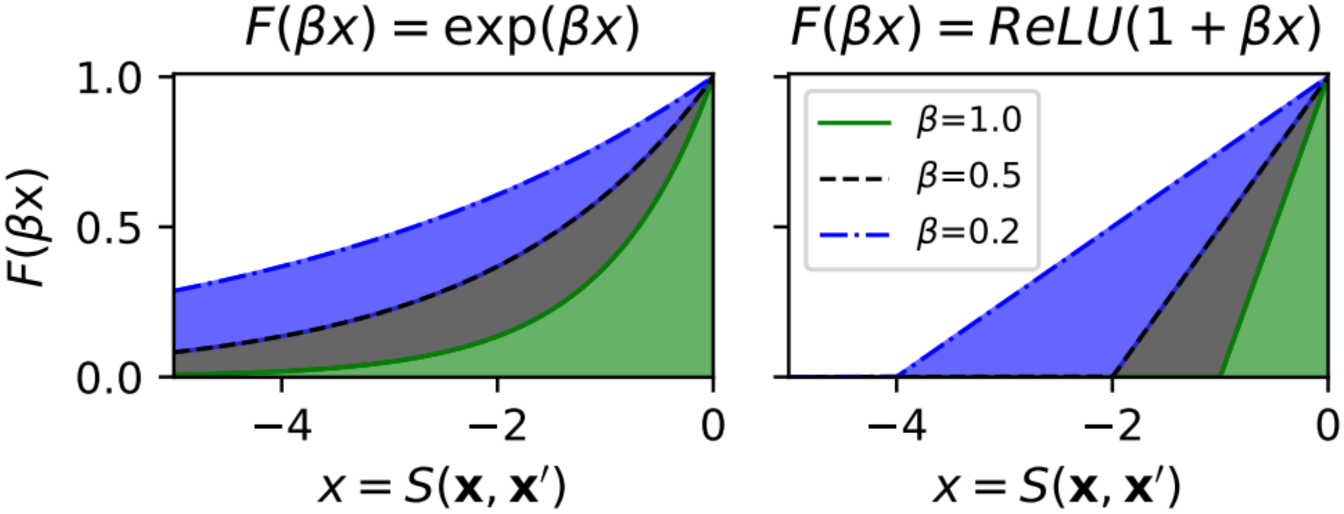
$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -\frac{1}{\beta} \log \sum_{\mu=1}^{K} \exp\left(-{}^{\beta}\!/_{2} \left\|\mathbf{v} - \boldsymbol{\xi}^\mu\right\|^2\right)$$

$$E_\beta(\mathbf{v}; \boldsymbol{\Xi}) = -\frac{1}{\beta} \log \sum_{\mu=1}^{K} \mathrm{ReLU}\left(1 - {}^{\beta}\!/_{2} \left\|\mathbf{v} - \boldsymbol{\xi}^\mu\right\|^2\right)$$

- Exponential capacity *without* exponential separation function
- Simultaneously retrieves memories **and** generates many new local minima

Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Log-Sum-Shifted-ReLU or LSR Energy



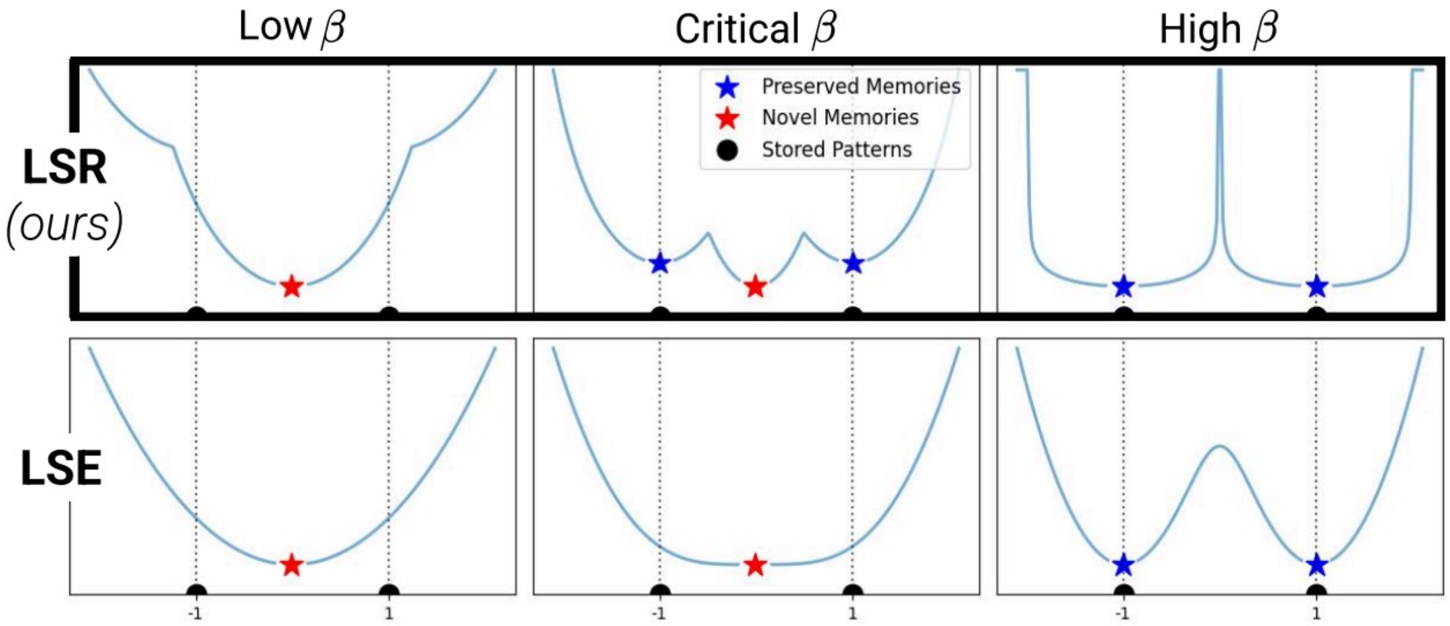
- Exponential capacity *without* exponential separation function
- Simultaneously retrieves memories **and** generates many new local minima

Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Epanechnikov Energy and Emergent Memories

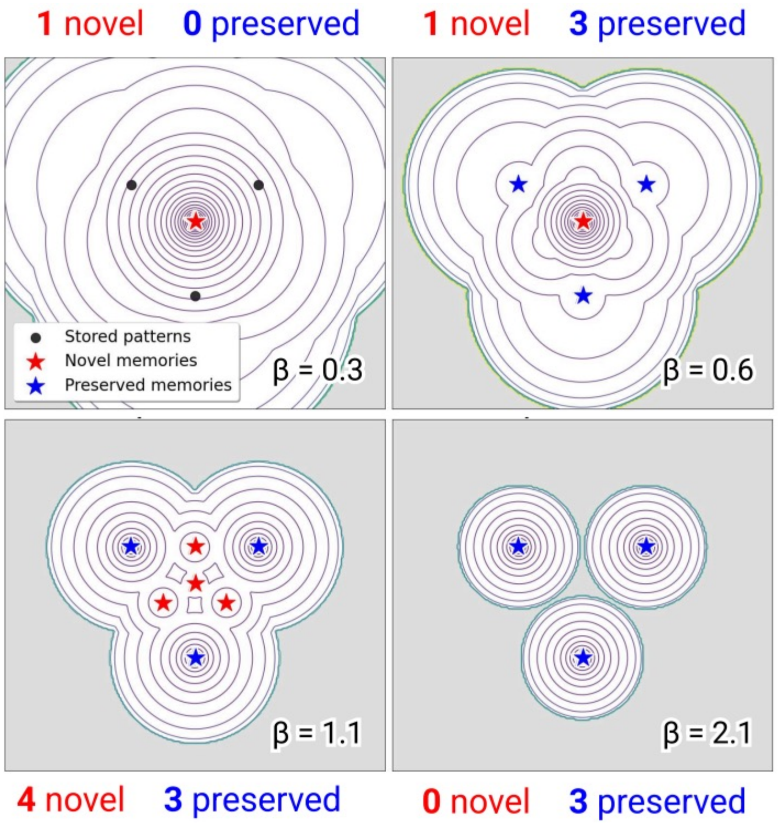
LSR **preserves** memories while creating **novel** ones.

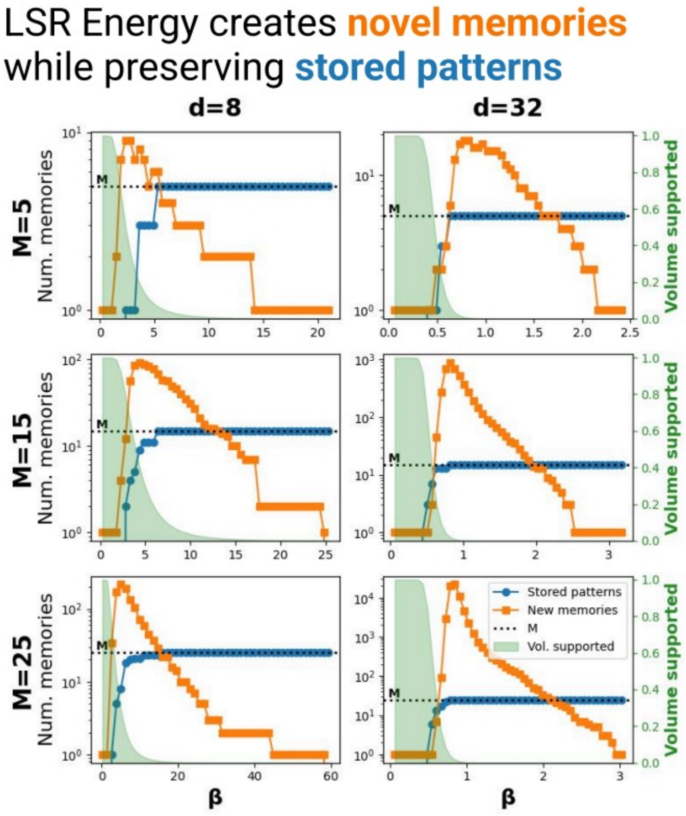
LSE can do only one or the other.



Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Epanechnikov Energy and Emergent Memories

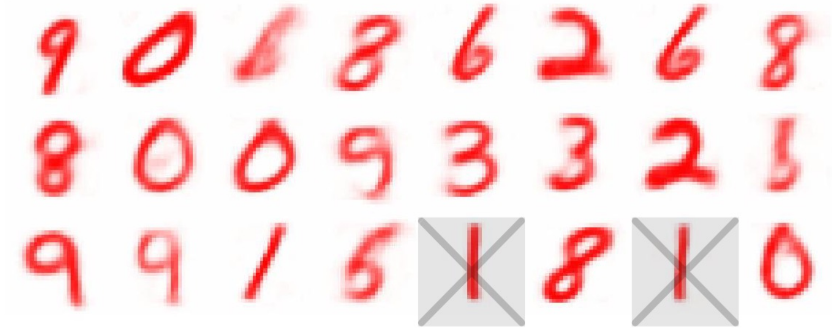




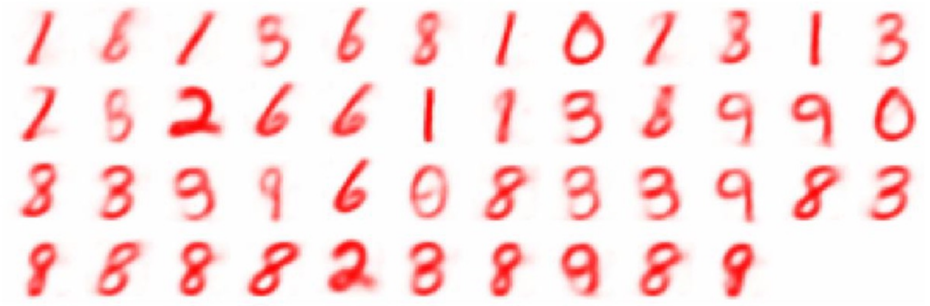
Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Epanechnikov Energy and Emergent Memories

22 / 24 **preserved** memories



46 **emergent** memories



Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

Epanechnikov Energy and Emergent Memories

40/ 40 **preserved** memories

38 **emergent** memories

Hoover, et al. "*Dense Associative Memory with Epanechnikov Energy.*" NFAM@ICLR 2025.

← Go to **ICLR 2025 Workshop NFAM** homepage

Dense Associative Memory with Epanechnikov energy

Benjamin Hoover, Krishna Balasubramanian, Dmitry Krotov, Parikshit Ram

How to think about Associative Memory Networks as a Machine Learning Model?

How to use them for standard Machine Learning tasks?

Chapter 5

Associative Memory:
A Machine Learning Model

Krotov, et al. "*Modern Methods in Associative Memory.*" 2025.