DENSE ASSOCIATIVE MEMORY WITH EPANECHNIKOV ENERGY

Benjamin Hoover
IBM Research
Georgia Tech

Krishnakumar Balasubramanian
UC Davis

Dmitry Krotov
IBM Research

Parikshit Ram
IBM Research

ABSTRACT

We propose a novel energy function for Dense Associative Memory (DenseAM) networks, the log-sum-ReLU (LSR), inspired by optimal kernel density estimation. Unlike the common log-sum-exponential (LSE) function, LSR is based on the Epanechnikov kernel and enables exact memory retrieval with exponential capacity without requiring exponential separation functions. Uniquely, it introduces abundant additional *emergent* local minima while preserving perfect pattern recovery–a characteristic previously unseen in DenseAM literature. Empirical results show LSR generates significantly more local minima and produces samples with higher log-likelihood than LSE-based models, making it promising for both memory storage and generative tasks.

1 Associative Memories and Energy Functions

Energy-based Associative Memory networks or AMs are models parameterized with M "memories" in d dimensions, $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_\mu \in \mathbb{R}^d, \mu \in [\![M]\!]\}$. A popular class of models from this family can be described by an energy function defined on the state vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$:

$$E(\mathbf{x}; \boldsymbol{\Xi}) = -Q\left[\sum_{\mu=1}^{M} F\left(\beta S\left(g(\mathbf{x}), \boldsymbol{\xi}_\mu\right)\right)\right], \tag{1}$$

where $g : \mathbb{R}^d \to \mathbb{R}^d$ is a vector operation (such as binarization, (layer) normalization), $S : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is similarity function (e.g., dot-product, negative Euclidean distance), $\beta > 0$ denotes the inverse temperature, $F : \mathbb{R} \to \mathbb{R}$ is a rapidly growing separation function (power, exponential) and Q is a monotonic scaling function (logarithm, linear) (Hoover et al., 2024). With g as the sign-function, $\boldsymbol{\xi}_\mu \in \{-1, +1\}^d$, $S(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$ and F as the quadratic function, and Q as a linear function, we recover the classical Hopfield model (Hopfield, 1982).

The output of the AM corresponds to one of the local minima of this energy function. A memory $\boldsymbol{\xi}_\mu$ is said to be retrieved if $\mathbf{x} \approx \boldsymbol{\xi}_\mu$ corresponds to one of the local minima, and the memory capacity of the AM corresponds to the largest number $M^\star$ of correctly retrieved memories. For the classical AM, $M^\star \sim O(d)$. With the introduction of rapidly increasing power separation function – that is, $F(x) = x^p, p > 2$ – the modern Dense Associative Memory (DenseAM) have a memory capacity of $M^\star \sim O(d^p)$ (Krotov & Hopfield, 2016). An exponential separation function and a logarithmic scaling function – $F(x) = \exp(x), Q(x) = \log x$ – gives us the widely considered log-sum-exp or LSE energy function (Demircigil et al., 2017; Ramsauer et al., 2021; Krotov & Hopfield, 2021) along with exponential memory capacity $M^\star \sim \exp(d)$ (Lucibello & Mézard, 2024). Additionally, hierarchical organizations of memories have been studied in Krotov (2021); Hoover et al. (2022).

In this work, we consider the following motivating question – *can we achieve simultaneous memorization and generalization?* While the exp separation function leads to large memory capacity in DenseAMs, memory capacity is not the only desiderata. The ability to create meaningful new patterns and handle complex data distributions is equally important, which has led researchers to explore alternative separation functions. Given that the gradient of LSE results in a softmax over all the memories, Hu et al. (2023) and dos Santos et al. (2024) consider sparsified versions of the softmax, resulting in new gradients for the LSE energy.[1] Wu et al. (2024) instead learn new representations for the memories to increase memory capacity, but still focus on the LSE energy (now in the learned representation space).

[1] Sparsified softmax based gradients can be viewed as specific projections of the original gradient.

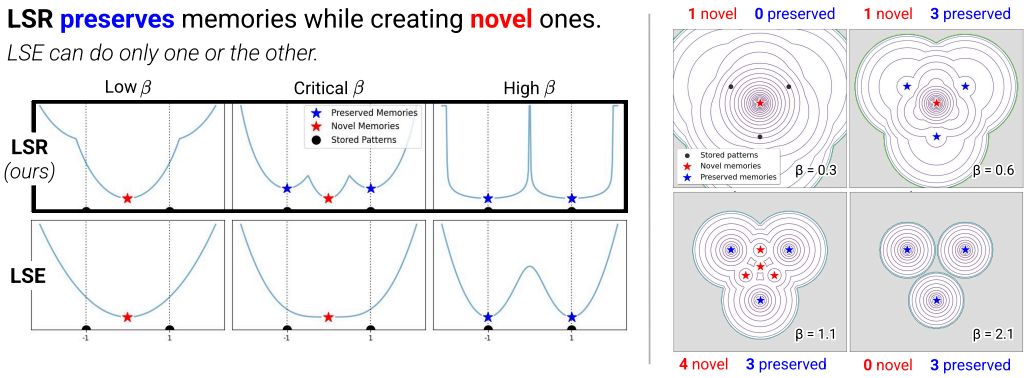

Figure 1: LSR energy can create more memories than there are stored patterns under critical regimes of β. Left: 1D LSR vs LSE energy landscape. Note that LSE is never capable of having more local minima than the number of stored patterns. Right: 2D LSR energy landscape, where increasing β creates novel local minima where basins intersect. Unsupported regions are shaded gray.

We instead consider the well-established connection between the energy and probability density function. An energy function $\mathsf{E} : \mathbb{R}^d \to \mathbb{R}$ induces a probability density function $p : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ with $p(\mathbf{x}) = \exp(-\mathsf{E}(\mathbf{x}))/\int_{\mathbf{z}} \exp(-\mathsf{E}(\mathbf{z}))d\mathbf{z}$. Conversely, given a density p, we have an energy $\mathsf{E}(\mathbf{x}) \propto -\log p(\mathbf{x})$, the negative log-likelihood, and minimizing the energy corresponds to maximizing the log-likelihood (with respect to the corresponding density). Based on this connection, with $Q(\cdot) = \log(\cdot)$, the $\exp(-E(\mathbf{x}; \boldsymbol{\Xi})) = \sum_\mu F(S(\mathbf{x}, \boldsymbol{\xi}_\mu))$ in eq. (1) (assuming g is identity) is the corresponding (unnormalized) density at $\mathbf{x}$. Assuming that the memories $\boldsymbol{\xi}_\mu \sim p$ are sampled from an unknown ground truth density p, the $\exp(-E(\mathbf{x}; \boldsymbol{\Xi}))$ is an unnormalized **kernel density estimate** or KDE of p at $\mathbf{x}$ with the *kernel* F and bandwidth $1/\beta$ (Wand & Jones, 1994). Thus, the LSE energy with $F(x) = \exp(x)$ and $S(\mathbf{x}, \mathbf{x}') = -1/2\|\mathbf{x} - \mathbf{x}'\|^2$ corresponds to the KDE of p with the Gaussian kernel.

KDE is well studied in nonparametric statistics (Wand & Jones, 1994; Devroye & Lugosi, 2001), and various forms of kernels have been explored. The quality of the estimates are well characterized in terms of properties on the kernels; we will elaborate on this in the sequel. While the Gaussian kernel is extremely popular for KDE (much like LSE in AM literature), there are various other kernels which have better estimation abilities than the Gaussian kernel. Among the commonly used kernels, the Epanechnikov kernel has the most favorable estimation quality (see section 2). In our notation, this corresponds to a kernel $F(x) = \max(1 + x, 0) = \mathrm{ReLU}(1 + x)$, a shifted ReLU operation (again with $S(\mathbf{x}, \mathbf{x}') = -1/2\|\mathbf{x} - \mathbf{x}'\|^2$). This results in a novel energy function we name log-sum-ReLU or LSR (see eq. (3)).

While the Epanechnikov kernel has the most favorable guarantees in KDE, it is not clear what such guarantees in KDE mean for DenseAM. This is the main topic of this paper – what does the LSR energy bring to the table? To this end, we make the following contributions:

- **Novel ReLU-based energy function with exponential memory capacity.** We propose a LSR energy function for DenseAM utilizing the popular ReLU activation, built upon the connection between energy functions and densities. We demonstrate exact retrieval and exponential memory capacity of LSR energy, without the use of exp as the separation function.
- **Simultaneous storage and emergence.** We show that this LSR energy has a unique property of *simultaneously* being able to retrieve all original memories (training data points) while also creating many additional *emergent*[2] local minima – the total number of local energy minima of LSR can exceed the number of stored patterns in the DenseAM, a property absent with LSE.
- **Memory-preserving generation.** We explore using these abundant new local minima to sample from a distribution, creating a DenseAM that simultaneously *memorizes* original memories and *generates* new ones. See fig. 1 for an illustration.

[2]AM literature typically refers to unexpected minima as "spurious" local minima. We use the term "emergent" to emphasize that these minima can be meaningful samples of an underlying distribution (see section 4.2)

2 Kernel Density Estimation and the Choice of Kernels

We now provide brief overview of Kernel Density Estimation (KDE) considering the univariate setting for simplicity; similar conclusion hold also in higher dimensions. Given a sample $\boldsymbol{\Xi} = \{\xi_\mu \in \mathbb{R}, \mu \in [\![M]\!]\}$ drawn from an unknown density f, the KDE is defined as $\hat{f}_h(\xi) = (Mh)^{-1} \sum_{\mu=1}^{M} K\left(\frac{\xi - \xi_\mu}{h}\right)$, where $K(\cdot)$ is the kernel function and $h > 0$ is the bandwidth parameter. The kernel function is assumed to satisfy: (i) symmetry (i.e., $K(-\xi) = K(\xi)$, for all $\xi \in \mathbb{R}$), (ii) positivity (i.e., $K(\xi) \geq 0$, for all $\xi \in \mathbb{R}$) and (iii) normalization (i.e., $\int K(\xi)\, d\xi = 1$). Note immediately that for the purpose of KDE, the scale of the kernel function is not unique. That is, for a given $K(\cdot)$, we can define $\tilde{K}(\cdot) = b^{-1}K(\cdot/b)$, for some $b > 0$. Then, one obtains the same KDE by rescaling the choice of h. Hence, the shape of the kernel function plays a more important in determining the choice of the kernel. We now introduce two parameters associated with the kernel, $\mu_K \coloneqq \int \xi^2 K(\xi)\, d\xi$ and $\sigma_K \coloneqq \int K^2(\xi)\, d\xi$ that correspond to the *scale* and *regularity* of the kernel. We will discuss below how the **generalization error** of KDE depends on the aforementioned parameters.

The **generalization error** of $\hat{f}_h(x)$ is measured by the Mean Integrated Squared Error (MISE), given by $\text{MISE}(h) = \mathbb{E}[\int (\hat{f}_h(\xi) - f(\xi))^2 d\xi]$. Assuming $f(\xi)$ is twice continuously differentiable, a second-order Taylor expansion gives the leading terms of the $\text{MISE}(h)$, which decomposes into squared bias and variance terms: $\mathsf{MISE}(h) \approx \frac{\mu_K^2}{4} h^4 \int |f''(\xi)|^2 d\xi + \frac{\sigma_K}{nh}$; see Wand & Jones (1994, Section 2.5) for details. This result shows that reducing h decreases bias but increases variance, while increasing h smooths the estimate but introduces bias, highlighting the bias-variance tradeoff. The optimal mean-square is obtained by minimizing $\text{MISE}(h)$ with respect to h. Doing so, we obtain the optimal choice of h and the optimal generalization accuracy as

$$h_* \coloneqq \left(\frac{\sigma_K^2}{n\mu_K^2} \frac{4}{\int |f''(\xi)|^2 d\xi}\right)^{1/5} \quad \text{and} \quad \mathsf{MISE}(h_*) \approx \frac{5}{4}\left(\frac{\sqrt{\mu_K}\sigma_K \int |f''(\xi)|^2 d\xi}{n}\right)^{4/5}, \tag{2}$$

respectively. From this, we see that the choice of the kernel K in the KDE, controls the generalization error via the term $\sqrt{\mu_K}\sigma_K$.

Thus, a natural question is to find the choice of kernel $K(\cdot)$ that results in the minimum $\mathsf{MISE}(h_*)$. As discussed above, the scale of the kernel function is non-unique. Hence, the problem boils down to minimizing σ_K (which is regularity parameter of the kernel, determining the shape), subjected to $\mu_K = 1$ (without loss of generality), over the class of normalized, symmetric, and positive kernels. This problem is well-studied (see, for example, Epanechnikov (1969), Müller (1984), Wand & Jones (1994, Section 2.7)), and as it turns out, the Epanechnikov kernel achieves the optimal **generalization error**. The quantity, $\mathsf{Eff(K)} \coloneqq \sigma_K / \sigma_{K_{\text{epan}}}$ is hence referred to as the efficiency of any kernel with respect to the Epanechnikov kernel. A description of choices for kernel functions and their efficiency relative to the Epanechnikov kernel is provided in Section A.

3 A New Energy Function

Given the motivation for using the Epanechnikov kernel in KDE, we will explore the use of the corresponding shifted-ReLU separation function $\text{ReLU}(1 + x)$ in the energy function instead of the widely used exponentiation. Before we state the precise energy functions, we compare and contrast the shapes of these separation functions $F(\beta x)$ in fig. 2 for varying values of the inverse temperature β. Note that, as the β increases, both these separation functions decay faster. However, as expected, the shifted-ReLU separation linearly decays and then zeroes out.

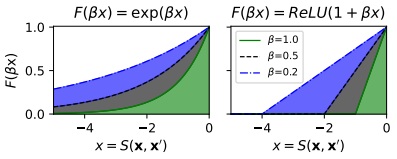

Figure 2: Visualizing the separation functions $F(\beta x) = \exp(\beta x)$ (LSE) and $F(\beta x) = \text{ReLU}(1 + \beta x)$ (LSR) with $x = S(\mathbf{x}, \mathbf{x}')$ for varying values of β. We focus on $S(\mathbf{x}, \mathbf{x}') = -1/2\|\mathbf{x} - \mathbf{x}'\|^2$.

Recall that the energy of the LSE ENERGY is given by $E_{\text{LSE}}(\mathbf{x}) = -\frac{1}{\beta} \log \sum_{\mu=1}^{M} \exp(-\frac{\beta}{2} \|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2)$. Based on the discussion on separation functions, our proposed LSR ENERGY (which we also refer to as Epanechnikov energy) is given by

$$E_{\text{LSR}}(\mathbf{x}) = -\frac{1}{\beta} \log \left(\epsilon + \sum_{\mu=1}^{M} \text{ReLU}\left(1 - \frac{\beta}{2} \|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2\right)\right), \tag{3}$$

where $\|\cdot\|$ describes the Euclidean norm and β is an inverse temperature. The factor $\epsilon \geq 0$ in the LSR energy is a small nonnegative constant, and $\epsilon > 0$ ensures that every point in the space has finite (albeit extremely large $O(\log(1/\epsilon))$) energy for all values of β. Indeed, with $\epsilon = 0$, defining $\mathsf{S}_\mu \triangleq \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \boldsymbol{\xi}_\mu\| \leq \sqrt{2/\beta}\}$, it is easy to see that $\forall \mathbf{x} \in \mathcal{X} \backslash \cup_{\mu=1}^{M} \mathsf{S}_\mu$, $E_{\text{LSR}}(\mathbf{x}) = \infty$. This is a result of the finite-ness of the ReLU based separation function. Regions of infinite energy implies zero probability density, which matches the finite support of the density estimate with the Epanechnikov kernel. Based on the introduced LSR energy, we next highlight the following favourable properties; see section B.3 for the proofs and technical details.

Theorem 1. *Let $r = \min_{\mu,\nu \in [\![M]\!], \mu \neq \nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$ be the minimum Euclidean distance between any two memories. Let $S_\mu(\Delta) = \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \boldsymbol{\xi}_\mu\| \leq \Delta\}$ be a basin around the μ^{th} memory for some basin radius $\Delta \in (0, r)$. Then, with $\beta = 2/(r - \Delta)^2$, for any $\mu \in [\![M]\!]$ and any input $\mathbf{x} \in S_\mu(\Delta)$, the output of the DenseAM via energy gradient descent is exactly $\boldsymbol{\xi}_\mu$, implying that all memories $\boldsymbol{\xi}_\mu, \mu \in [\![M]\!]$ are retrievable. Furthermore, if the learning rate of the energy gradient descent is set appropriately, then for any $\mu \in [\![M]\!]$ and any $\mathbf{x} \in S_\mu(\Delta)$, the memory is exactly retrieved with a single energy gradient descent step (single step retrieval).*

This above result states that, given a set of memories, and an appropriately selected β, there is a distinct basin of attraction $S_\mu(\Delta)$ around each memory $\boldsymbol{\xi}_\mu$, and any input $\mathbf{x}$ from within that basin exactly retrieves the memory as the output of the DenseAM.

Theorem 2. *Consider a DenseAM parameterized with M memories $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M$ sampled uniformly from $\{-1, +1\}^D$. Then, with probability at least $\delta \in (0, 1)$, and $M \sim O(\sqrt{\log(1/\delta)} \exp(\alpha^2 D/2))$ for a positive $\alpha \in (0, 1)$, all memories are retrievable as per theorem 1 with the value of the minimum pairwise distance $r = \min_{\mu,\nu \in [\![M]\!], \mu \neq \nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\| \geq \sqrt{2d(1-\alpha)}$ and per-memory basin radius $\Delta \in (0, \sqrt{2d(1-\alpha)})$ with a $\beta \geq 2/(\sqrt{2d(1-\alpha)} - \Delta)^2$.*

This result shows that the DenseAM equipped with this novel LSR energy has exponential memory capacity similar to that of LSE energy (Ramsauer et al., 2021; Lucibello & Mézard, 2024). Finally, we show that, for the DenseAM configured as per in Theorem 2 with exponential memory capacity can also create potentially many new *emergent* local minima:

Theorem 3. *Consider the configuration of the DenseAM in Theorem 2. For any input $\mathbf{x} \in \mathcal{X}$, let $B(\mathbf{x}) \triangleq \{\mu \in [\![M]\!] : \|\mathbf{x} - \boldsymbol{\xi}_\mu\| \leq \sqrt{2/\beta}\}$. For any $\mathbf{x}$ such that $|B(\mathbf{x})| > 1$, the energy descent with the LSR energy will return a new emergent minima given by $\frac{1}{|B(\mathbf{x})|} \sum_{\mu \in B(\mathbf{x})} \boldsymbol{\xi}_\mu$.*

Note that, with $|B(\mathbf{x})| > 1$, the output of the DenseAM is not equal to any of the original memories $\{\boldsymbol{\xi}_\mu, \mu \in [\![M]\!]\}$. The region $\{\mathbf{x} \in \mathcal{X} : |B(\mathbf{x})| > 1\} \subset \mathcal{X}$ is precisely characterized as $\left(\cup_{\mu \in [\![M]\!]} \mathsf{S}_\mu\right) \backslash \left(\cup_{\mu \in [\![M]\!]} S_\mu(\Delta)\right)$ where S_μ is the region of finite energy around the μ^{th} memory and $S_\mu(\Delta)$ (defined in Theorem 1) is the distinct attracting basin for the μ^{th} memory. This implies that this DenseAM is capable of simultaneously retrieving all (up to exponentially many) memories while also creating many emergent local minima. While we have not precisely characterized the total number of these emergent local minima, this number is naively bounded from above by 2^M.

4 Experiments

4.1 Quantifying novel minima

To quantify the number of local minima induced by the LSR energy, we uniformly sample M patterns from the d-dimensional unit hypercube to serve as memories $\boldsymbol{\Xi}$. We can enumerate all possible local minima of LSR energy by computing the centroid $\bar{\boldsymbol{\xi}}_\mathcal{K} := \frac{1}{|\mathcal{K}|} \sum_{\mu \in \mathcal{K}} \boldsymbol{\xi}_\mu$ for all possible subsets of memories $\mathcal{K} \subseteq [\![M]\!]$ (including singleton sets, there are 2^M possible subsets). For each centroid, we first check that the centroid is suppported (i.e., that $E_{\text{LSR}}(\bar{\boldsymbol{\xi}}_\mathcal{K}) < \infty$ at $\epsilon = 0$), and then declare that $\bar{\boldsymbol{\xi}}_\mathcal{K}$ is a local minimum of the LSR energy if $\left\|\nabla E_{\text{LSR}}(\bar{\boldsymbol{\xi}}_\mathcal{K})\right\| < \delta$ for small $\delta > 0$. The results for this analysis at different choices for M, β, and d are shown in fig. 3 (left), where β values are varied across the interesting regime between fully overlapping support regions (a single local minimum in the unit hypercube) to fully disjoint support regions around each memory.See also section B.1.

Under certain ranges of β, we observe that LSR can preserve the stored patterns while simultaneously creating orders of magnitude more "novel" memories (i.e., memories that do not appear in $\boldsymbol{\Xi}$).

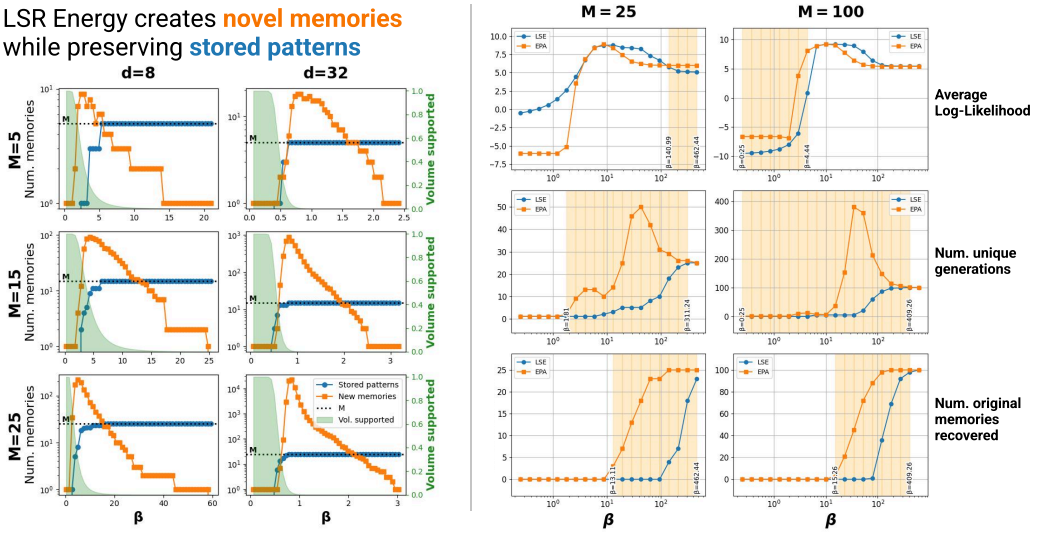

Figure 3: (Left) Analyzing local minima in LSR energy reveals a number of **novel memories** several orders of magnitude larger than M, the number of stored patterns, at critical values of β (note that the y-axes are logscale). The novel memories emerge at overlapping support regions of the stored patterns while still ensuring the **stored patterns** live at local minima. Smaller values of β have a larger **region of support** on the unit hypercube. (Right) In regimes where LSR samples many distinct memories, we also observe a log-likelihood comparable to, and even slightly higher than, LSE under the true density function (mixture of $k = 5$ Gaussians, $\sigma = 0.1$ and $d = 8$). Specific regions of β where LSR outperforms LSE on particular metrics are specified by the orange regions.

The peak number of novel memories consistently occurs when β is tuned such that approximately 60% of the original stored patterns remain recoverable and 20% of the unit hypercube is supported, regardless of the choice of M and d.

4.2 Quality of Log-likelihood vs. Gaussian Kernel

Let $p(\mathbf{x})$ be a mixture of k Gaussians whose means $\boldsymbol{\mu}_i \sim \mathcal{U}([0,1]^d)$ for $i \in [\![k]\!]$ are uniformly sampled from the d-dimensional unit hypercube with scalar covariances such that $p(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k}\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_i, \sigma^2\mathbf{I}_d)$. We sample M points $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M \sim p(x)$ to serve as the stored patterns $\boldsymbol{\Xi}$ used to parameterize both the LSE and LSR energies from eq. (3). Define the *support boundary* induced by pattern $\boldsymbol{\xi}_\mu$ to be $\text{supp}[\boldsymbol{\xi}_\mu] = \{\mathbf{x} \mid \|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2 = 2\beta^{-1}\}$. Then, for $E \in \{E_{\text{LSE}}, E_{\text{LSR}}\}$ and points $\mathbf{x}_n^{(0)}$, $n \in [\![N]\!]$ sampled near the support boundary of each stored pattern,[3] samples can be generated using gradient descent

$$\mathbf{x}_n^{(t)} = \mathbf{x}_n^{(t-1)} - \alpha\nabla E(\mathbf{x}_n^{(t-1)}), \tag{4}$$

for some small step size $\alpha > 0$ until convergence to a *memory* $\mathbf{x}_n^\star$. Thus we have N samples $\mathbf{x}_1^\star, \ldots, \mathbf{x}_N^\star$ on which we evaluate three metrics of interest in fig. 3 (right):

1. **Average Log-Likelihood**. Does LSR generate samples with higher log-likelihood under $p(\mathbf{x})$ than LSE?
2. **Number of Unique Samples**. How many more local minima can we empirically retrieve from LSR compared to LSE?
3. **Number of Original Memories Recoverable**. Can LSR recover more of the stored patterns in regimes where it also generates novel memories?

The results tell a consistent story. Despite LSE energy being a more natural choice to model the underlying $p(\mathbf{x})$ (given its inherent Gaussian mixture structure), LSR can outperform LSE in log-likelihood while simultaneously generating more diverse samples and preserving the recoverability of the stored patterns. See section B.2 for experimental details and extended discussion.

[3] We use the same initial points to seed the dynamics of both E_{LSR} and E_{LSE}. See section B.2 for details.

References

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

Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2001.

Saul José Rodrigues dos Santos, Vlad Niculae, Daniel C McNamee, and Andre Martins. Sparse and structured hopfield networks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=OdPlFWExX1.

Vassiliy A Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

Benjamin Hoover, Duen Horng Chau, Hendrik Strobelt, and Dmitry Krotov. A universal abstraction for hierarchical hopfield networks. In *The Symbiosis of Deep Learning and Differential Equations II*, 2022.

Benjamin Hoover, Duen Horng Chau, Hendrik Strobelt, Parikshit Ram, and Dmitry Krotov. Dense associative memory through the lens of random features. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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

Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/57bc0a850255e2041341bf74c7e2b9fa-Paper-Conference.pdf.

Dmitry Krotov. Hierarchical associative memory. *arXiv preprint arXiv:2107.06446*, 2021.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in Neural Information Processing Systems*, 29, 2016.

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

Carlo Lucibello and Marc Mézard. Exponential capacity of dense associative memories. *Physical Review Letters*, 132(7):077301, 2024.

Hans-Georg Müller. Smooth optimum kernel estimators of densities, regression curves and modes. *The Annals of Statistics*, pp. 766–774, 1984.

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

Matt P Wand and M Chris Jones. *Kernel smoothing*. CRC press, 1994.

Dennis Wu, Jerry Yao-Chieh Hu, Teng-Yun Hsiao, and Han Liu. Uniform memory retrieval with larger capacity for modern hopfield models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53471–53514. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24i.html.

A On Kernels

We show different kernels that are typically used for KDE and their efficiency relative to the Epanechnikov kernel in fig. 4. See the explanation on optimal kernel density estimation in section 2 for more details.

B Experimental Details

B.1 Details: Quantifying Novel Minima

In this experiment we tested across a geometrically spaced range of $\beta \in [2d^{-1}, 2r_{\text{min}}^{-2}]$, where $r_{\text{min}} := \min_{\mu \neq \nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\|$ is the minimum pairwise distance between any two stored patterns in the current subset $\mathcal{K} \subseteq [\![M]\!]$. At the largest β, the support regions of the stored patterns are disjoint and the only memories are the M stored patterns themselves; this configuration has a very small support region (shown as the shaded green curve in fig. 3, which is computed by monte carlo sampling 1e6 points on the unit hypercube and computing the fraction of energies that are finite at $\epsilon = 0$) as a fraction of the unit hypercube. At the smallest tested β, only a single energy minimum is induced at the centroid of all stored patterns with a region of support covering the whole unit hypercube. At the largest tested β, all original memories are recoverable and there are no spurious memories.

B.2 Details: Quality of Log-Likelihood

B.2.1 Determining the uniqueness of minima

500 initial points are uniformly sampled from the support boundary around each memory before the gradient descent process in eq. (4) is performed for 13000 steps at a cosine-decayed learning rate α from $0.01 \to 0.0001$. However, even after this descent process, special care must be taken to ensure that unique minima are correctly computed. When sampling from the energy function via discrete gradient descent, there is necessarily some error introduced by the discrete step size α and floating point precision. Generally, memory retrieval is said to converge when $\|\nabla_{\mathbf{x}} E(\mathbf{x})\| < \epsilon$ for some small $\epsilon > 0$, or when the number of iterations T exceeds some threshold at small α. When counting the number of unique samples from the LSE energy, we perform spectral clustering on the graph created by the generated samples, where two samples $\mathbf{x}_A^\star$ and $\mathbf{x}_B^\star$ are *adjacent* if $\|\mathbf{x}_A^\star - \mathbf{x}_B^\star\| \leq \frac{2}{\beta}$. We choose a threshold of $1\text{e}{-5}$ on the eigenvalues of the Laplacian.

When counting the uniqueness of the samples from the LSR energy, we perform the following trick to exactly compute the fixed points of the dynamics. We first compute our "best guess" for the fixed point by performing standard gradient descent according to eq. (4) for T steps, at which point $\mathbf{z} := \mathbf{x}^{(T)}$ is close (but not exactly equal) to the fixed point $\mathbf{x}^\star$. We then pass $\mathbf{z}$ to algorithm 1 to compute the fixed point exactly. With a good initial guess $\mathbf{z}$, algorithm 1 converges after a single iteration.

Finally, we choose to sample points near the support boundary of each stored pattern because this maximizes the probability that we will end up in a spurious minimum. The size of spurious basins in high dimension can be very small, and the probability of landing in them decreases rapidly with increasing β (see the region of support plot in fig. 3). This, in addition to being computationally limited to a small number of total samples on which we perform the memory retrieval in eq. (4)

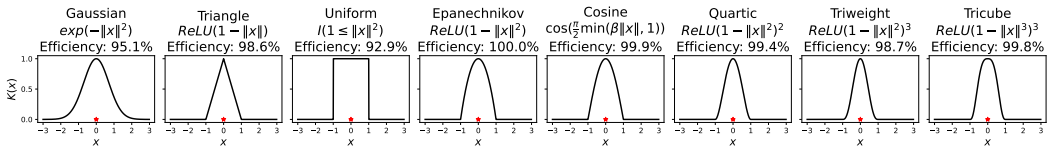

Figure 4: Different kernels used in KDE with their expression and KDE efficiency relative to the Epanechnikov kernel (*higher is better*, see text for details). The center of each kernel is marked with a red $\star$. To highlight the shape of the kernel, we have removed any scaling in the kernel expression. Note that all above kernels except Gaussian have finite support. The Epanechnikov kernel has the highest efficiency (100%). While the Gaussian kernel is extremely popular, and it is more efficient (95.1%) than the Uniform kernel (92.9%), there are various other kernels with better efficiency.

Algorithm 1: Fixed Point Computation for the LSR Memory

Input: Initial guess $\mathbf{z}$, stored patterns $\{\boldsymbol{\xi}_m\}_{m=1}^M$, inverse temperature β
Output: Fixed point $\mathbf{z}^\star$
$\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z} + \infty$ `// Initialize previous point`
while $\mathbf{z}_{prev} \neq \mathbf{z}$ **do**
  $\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}$
  $S(\mathbf{z}) \leftarrow \{\boldsymbol{\xi}_m \; ; \; \|\mathbf{z} - \boldsymbol{\xi}_m\|^2 \leq \frac{2}{\beta}\}$ `// Compute support centroids`
  $\mathbf{z} \leftarrow \frac{1}{|S(\mathbf{z})|} \sum_{\boldsymbol{\xi}_m \in S(\mathbf{z})} \boldsymbol{\xi}_m$ `// Update to mean of support centroids`
end
return $\mathbf{z}$

B.3 Theoretical Results

B.3.1 Proof of Theorem 1

For any $\mathbf{x} \in \mathcal{X}$, let $B(\mathbf{x}) = \{\mu \in [\![M]\!] : \|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2 \leq 2/\beta\}$. Then the gradient of the LSR energy in equation 3 is given by

$$\nabla_{\mathbf{x}} E_{\text{LSR}}(\mathbf{x}) = \sum_{\mu=1}^{M} \frac{(\mathbf{x} - \boldsymbol{\xi}_\mu)\, \mathbb{1}\left[\|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2 \leq \frac{2}{\beta}\right]}{\epsilon + \left[\sum_{\nu=1}^{M} \text{ReLU}\left(1 - \frac{\beta}{2}\|\mathbf{x} - \boldsymbol{\xi}_\nu\|^2\right)\right]} \tag{5}$$

$$= \frac{\sum_{\mu \in B(\mathbf{x})} (\mathbf{x} - \boldsymbol{\xi}_\mu)}{\epsilon + \left[\sum_{\nu \in B(\mathbf{x})} \text{ReLU}\left(1 - \frac{\beta}{2}\|\mathbf{x} - \boldsymbol{\xi}_\nu\|^2\right)\right]} \tag{6}$$

With $\beta = 2/(r - \Delta)^2$, $B(\mathbf{x}) = \{\mu \in [\![M]\!] : \|\mathbf{x} - \boldsymbol{\xi}_\mu\| \leq (r - \Delta)\}$. For any $\mathbf{x} \in S_\mu(\Delta)$, $B(\mathbf{x}) = \{\mu\}$. Thus the LSR energy gradient simplifies to

$$\forall \mathbf{x} \in S_\mu(\Delta), \quad \nabla_{\mathbf{x}} E_{\text{LSR}}(\mathbf{x}) = \frac{(\mathbf{x} - \boldsymbol{\xi}_\mu)}{\epsilon + \text{ReLU}\left(1 - \frac{\beta}{2}\|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2\right)}, \tag{7}$$

which is exactly zero at $\mathbf{x} = \boldsymbol{\xi}_\mu$, thus giving us the retrieval of the μ^{th} memory via energy gradient descent.

Furthermore, again for $\mathbf{x} \in S_\mu(\Delta)$ with a energy gradient descent learning rate set to $\eta \leftarrow \epsilon + \text{ReLU}\left(1 - \beta/2\|\mathbf{x} - \boldsymbol{\xi}_\mu\|^2\right)$, the update is exactly $\eta \nabla_{\mathbf{x}} E_{\text{LSR}}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\xi}_\mu)$. Thus a single step gradient descent update to $\mathbf{x}$ with $\mathbf{x} - \eta \nabla_{\mathbf{x}} E_{\text{LSR}}(\mathbf{x}) = \mathbf{x} - (\mathbf{x} - \boldsymbol{\xi}_\mu) = \boldsymbol{\xi}_\mu$ results in the retrieval of the μ^{th} memory.

B.3.2 Proof of theorem 2

Lemma 1 (See Corollary 10 here)**.** *If $\mathbf{x}, \mathbf{x}'$ are chosen randomly from $\{-1, +1\}^D$, then*

$$\Pr\left[\left|\frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\|\mathbf{x}\|\,\|\mathbf{x}'\|}\right| > \sqrt{\frac{\log c}{D}}\right] < \frac{1}{c}. \tag{8}$$

Lemma 2. *For any $x > -1$, we have*

$$\frac{x}{1+x} \leq \log(1+x) \leq x. \tag{9}$$

Proof. (Theorem 2.)

Given memories $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M$ sampled randomly from $\mathcal{X} = \{-1, +1\}^D$, Lemma 1 states that for any pair of memories $\boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu$, with a scalar $c > 1$, their cosine similarity is bounded from above as:

$$\Pr\left[\frac{\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle}{\|\boldsymbol{\xi}_\mu\| \|\boldsymbol{\xi}_\nu\|} \leq \sqrt{\frac{\log c}{D}}\right] \geq \Pr\left[\left|\frac{\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle}{\|\boldsymbol{\xi}_\mu\| \|\boldsymbol{\xi}_\nu\|}\right| \leq \sqrt{\frac{\log c}{D}}\right] \tag{10}$$

$$= 1 - \Pr\left[\left|\frac{\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle}{\|\boldsymbol{\xi}_\mu\| \|\boldsymbol{\xi}_\nu\|}\right| > \sqrt{\frac{\log c}{D}}\right] \tag{11}$$

$$\geq \left(1 - \frac{1}{c}\right). \tag{12}$$

Thus, the maximum cosine similarity between any pairs of memories can be bounded as:

$$\Pr\left[\left(\max_{\mu,\nu \in [\![M]\!], \mu \neq \nu} \frac{\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle}{\|\boldsymbol{\xi}_\mu\| \|\boldsymbol{\xi}_\nu\|}\right) \leq \sqrt{\frac{\log c}{D}}\right] = \Pr\left[\bigwedge_{\mu,\nu \in [\![M]\!], \mu \neq \nu} \left(\frac{\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle}{\|\boldsymbol{\xi}_\mu\| \|\boldsymbol{\xi}_\nu\|} \leq \sqrt{\frac{\log c}{D}}\right)\right] \tag{13}$$

$$\geq \prod_{\mu,\nu \in [\![M]\!], \mu \neq \nu} \Pr\left[\frac{\langle \boldsymbol{\xi}_\mu, \boldsymbol{\xi}_\nu \rangle}{\|\boldsymbol{\xi}_\mu\| \|\boldsymbol{\xi}_\nu\|} \leq \sqrt{\frac{\log c}{D}}\right] \tag{14}$$

$$\geq \left(1 - \frac{1}{c}\right)^{M^2}. \tag{15}$$

If we set $\log c = \alpha^2 D$ for some $\alpha \in (0, 1)$, then the maximum cosine similarity between any pair of memories is at most α, and the thus, the minimum pairwise Euclidean distance $r = \min_{\mu,\nu \in [\![M]\!], \mu \neq \nu} \|\boldsymbol{\xi}_\mu - \boldsymbol{\xi}_\nu\| \geq \sqrt{2d(1-\alpha)}$. Then with $\beta = 2/(r - \Delta) \geq 2/(\sqrt{2d(1-\alpha)} - \Delta)$, we are able to retrieve any memory $\boldsymbol{\xi}_\mu$ with an $\mathbf{x} \in S_\mu(\Delta)$ with $\Delta < \sqrt{2d(1-\alpha)} \leq r$.

Now what remains to be seen is the relationship between the selected value of $\log c = \alpha^2 D$ and the success probability $\delta = (1 - 1/c)^{1/M^2}$. Plugging in the value of $c = \exp(\alpha^2 D)$ in the success probability, we have

$$\exp(\alpha^2 D) = \frac{1}{1 - \delta^{1/M^2}} \quad \Rightarrow 1 - \delta^{1/M^2} = \exp(-\alpha^2 D) \tag{16}$$

$$\Rightarrow M = \sqrt{\frac{\log(1/\delta)}{\log\left(\frac{1}{1 - \exp(-\alpha^2 D)}\right)}}. \tag{17}$$

Using Lemma 2, and noting that $\frac{1}{1-\exp(-\alpha^2 D)} = 1 + \frac{\exp(-\alpha^2 D)}{1-\exp(-\alpha^2 D)}$, we see that

$$\exp(-\alpha^2 D) \leq \log\left(\frac{1}{1 - \exp(-\alpha^2 D)}\right) \leq \frac{\exp(-\alpha^2 D)}{1 - \exp(-\alpha^2 D)} = \frac{1}{\exp(\alpha^2 D) - 1}. \tag{18}$$

Plugging this in eq. (17), we have

$$\sqrt{\log(1/\delta)(\exp(\alpha^2 D) - 1)} \leq M \leq \sqrt{\log(1/\delta)(\exp(\alpha^2 D))}, \tag{19}$$

giving us $M \sim O(\sqrt{\log(1/\delta)} \exp(\alpha^2 D/2))$. □

B.3.3 Proof of Theorem 3

Proof. (Theorem 3.)

For any $\mathbf{x} \in \mathcal{X}$, given the definition of $B(\mathbf{x})$, recall that the gradient of the LSR energy is given in eq. 5. When $|B(\mathbf{x})| > 1$, this gradient is zero when

$$\mathbf{x} = \frac{1}{|B(\mathbf{x})|} \sum_{\mu \in B(\mathbf{x})} \boldsymbol{\xi}_\mu,$$

the geometric mean of the memories corresponding to the set $B(\mathbf{x})$. □