HIERARCHICAL HOPFIELD NETWORK DECOMPOSITION: A SPIKED COVARIANCE FRAMEWORK FOR LATENT PROTOTYPE DISCOVERY

Saleh Sargolzaei
School of Computer Science
University of Windsor
sargolz@uwindsor.ca

Luis Rueda
School of Computer Science
University of Windsor
lrueda@uwindsor.ca

ABSTRACT

We revisit the classical Hopfield network from a spiked covariance perspective, showing how the Hebbian coupling matrix forms a low-rank perturbation of the identity. This viewpoint links outlier eigenvalues in the sample covariance matrix to latent signal vectors, explaining how multiple signals can fuse into a single spurious state. We propose a hierarchical algorithm that uses Hopfield updates to iteratively partition the data, isolating more granular spiked subspaces until no further mergers remain. Unlike classical approaches focusing on capacity alone, our method reveals latent signals even when they are strongly correlated. Experiments on MNIST and LFW confirm that these signals serve as interpretable "prototypes" and improve clustering initialization.

1 INTRODUCTION

The classical Hopfield network Hopfield (1982) is renowned for its ability to store and retrieve binary memory patterns, though it suffers from well-known capacity limitations and the emergence of undesired spurious states McEliece et al. (1987). Although recent advances (*e.g.*, Krotov & Hopfield (2016); Demircigil et al. (2017); Ramsauer et al. (2021)) have pushed the network beyond these historical bounds, the classical model provides valuable analytic tractability—making it an ideal platform for understanding fundamental phenomena such as spurious memory formation.

In parallel, the theory of *spiked covariance models* has revealed how outlier eigenvalues in large random matrices encode meaningful low-rank signals within high-dimensional noise Bloemendal et al. (2016); Ding & Yang (2021). We show that *Hebbian* learning in classical Hopfield networks induces precisely such a spiked structure on the coupling matrix, interpreted as a perturbation from the identity. This observation both explains the network's preference for dominant eigenvector directions and highlights how multiple signals can become fused into a single spurious state when sample-based interactions lead to overlap among outlier eigenvectors.

Although related work has leveraged spectral properties for improving Hopfield training Benedetti et al. (2024); Agliari et al. (2024), to the best of our knowledge, we present the first explicit connection to the spiked covariance model. We leverage this viewpoint to propose a *hierarchical* procedure that iteratively extracts latent signal vectors. Whenever multiple signals collapse into a single Hopfield memory, it indicates an inconsistency between the number of outlier eigenvalues (spikes) and the converged states, suggesting the need for finer decomposition. We then isolate the spiked subspace via Hopfield updates and remove *high-entropy* (uncertain) samples that can cause the dynamics to become "stuck" in one merged state. While our ultimate goal is to reduce each sub-dataset to a single spike, practical limits (e.g., maximum recursion depth) may result in sub-datasets containing more than one. Nevertheless, this hierarchical approach produces a set of interpretable prototypes, capturing salient latent signals even under finite-sample noise and correlated data.

2 BACKGROUND

2.1 CLASSICAL HOPFIELD NETWORK (BRIEF REVIEW)

Consider K memory patterns $\{\boldsymbol{\xi}_j\}_{j=1}^K \subset \{-1,1\}^M$, each of dimension M. A classical Hopfield network stores these patterns by constructing a symmetric coupling matrix $\boldsymbol{W} \in \mathbb{R}^{M\times M}$, whose element W_{ij} measures the strength of interaction between neurons i and j. A common choice is Hebb's rule (Hopfield, 1982; Hebb, 2005):

$$\boldsymbol{W} \;=\; \frac{1}{K}\,\boldsymbol{\Xi}\,\boldsymbol{\Xi}^T \;-\; \boldsymbol{I}_M, \quad \boldsymbol{\Xi} = [\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_K]. \tag{1}$$

Here, $\boldsymbol{I}_M$ is the $M \times M$ identity matrix. The state of the network at time t is $\boldsymbol{a}^{(t)} \in \{-1,1\}^M$, updated iteratively:

$$\boldsymbol{a}^{(t+1)} \;=\; \operatorname{sign}\Big[\boldsymbol{W}\,\boldsymbol{a}^{(t)}\Big]. \tag{2}$$

Under ideal conditions, each stored pattern $\boldsymbol{\xi}_j$ (or small variations around it) is a stable attractor. However, in practice, *spurious* stable states and capacity issues often arise.

2.2 SPIKED COVARIANCE MODEL

We next recall the *spiked covariance* framework (Bloemendal et al., 2016), which describes data vectors as a low-rank signal plus noise. Suppose we observe N samples $\{\boldsymbol{a}_i\}_{i=1}^N \subset \mathbb{R}^M$, stacked into $\boldsymbol{A} \in \mathbb{R}^{M\times N}$. Their empirical (sample) covariance is

$$\boldsymbol{Q} \;=\; \frac{1}{N}\,\boldsymbol{A}\,\boldsymbol{A}^T. \tag{3}$$

In the **population** model, each sample

$$\boldsymbol{a} \;=\; \boldsymbol{z} \;+\; \sum_{l=1}^{r} y_l\,\boldsymbol{u}_l, \tag{4}$$

with $\boldsymbol{z} \in \mathbb{R}^M$ an i.i.d. noise vector (mean zero, unit variance), scalars y_l (mean zero, unit variance), and r deterministic *signal vectors* $\{\boldsymbol{u}_l\}$. The *population covariance* is:

$$\boldsymbol{\Sigma} \;=\; \boldsymbol{I}_M \;+\; \boldsymbol{U}\,\boldsymbol{U}^T, \boldsymbol{U} = [\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_r] \in \mathbb{R}^{M\times r}. \tag{5}$$

When M is comparable to N, eigenvalues of $\boldsymbol{\Sigma}$ exceeding $1+\sqrt{M/N}$ produce outlier sample eigenvalues of $\boldsymbol{Q}$, with corresponding eigenvectors aligning to the population latent eigenvector. These outlier sample eigenvalues are saparated from bulk of the spectrum with edge boundaries:

$$\gamma_\pm := \sqrt{\frac{M}{N}} + (\sqrt{\frac{M}{N}})^{-1} \pm 2 \tag{6}$$

3 HEBB RULE AND SPIKED POPULATION COVARIANCE

3.1 FROM HEBBIAN WEIGHTS TO A SPIKED COVARIANCE MATRIX

Observe that the Hebbian coupling equation 1 is a rank-K perturbation of the identity:

$$\boldsymbol{W} \;=\; \tfrac{1}{K}\,\boldsymbol{\Xi}\,\boldsymbol{\Xi}^T \;-\; \boldsymbol{I}_M. \tag{7}$$

Meanwhile, a *spiked* covariance can be written

$$\boldsymbol{\Sigma} \;=\; \boldsymbol{I}_M \;+\; \tfrac{1}{K}\,\boldsymbol{\Xi}\,\boldsymbol{\Xi}^T. \tag{8}$$

Hence:

Proposition 3.1 (Hebbian $\leftrightarrow$ Spiked Covariance) *If we drop the $\{-1,1\}$ constraint on patterns, then*

$$\boldsymbol{W} \;=\; \boldsymbol{\Sigma} - 2\,\boldsymbol{I}_M \quad \textit{where} \quad \boldsymbol{\Sigma} \;=\; \boldsymbol{I}_M \;+\; \tfrac{1}{K}\,\boldsymbol{\Xi}\,\boldsymbol{\Xi}^T.$$

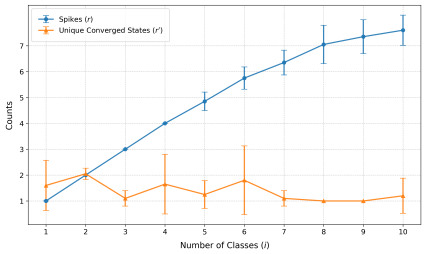


Figure 1: **Number of spikes vs. number of converged Hopfield states.**

Proof. Set $\boldsymbol{U} = \frac{1}{\sqrt{K}}\,\boldsymbol{\Xi}$. Then $\boldsymbol{U}\boldsymbol{U}^T = \frac{1}{K}\,\boldsymbol{\Xi}\boldsymbol{\Xi}^T$. Clearly, $\boldsymbol{W} = \boldsymbol{\Sigma} - 2\,\boldsymbol{I}_M$. □

3.2 Dynamics Toward Principal Components

Proposition 3.2 (Linearized Hopfield Dynamics) *Relax the sign function in eq.* (2) *and consider*

$$\boldsymbol{a}^{(t+1)} = \boldsymbol{W}\,\boldsymbol{a}^{(t)}, \quad \boldsymbol{W} = \boldsymbol{\Sigma} - 2\,\boldsymbol{I}_M. \tag{9}$$

If $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ has eigenvalues $\{\lambda_i\}$, then each coordinate along $\boldsymbol{v}_i$ evolves by a factor $(\lambda_i - 2)$. Hence, components with $\lambda_i > 3$ grow exponentially.

Proof. Since $\boldsymbol{W} = \boldsymbol{V}(\boldsymbol{\Lambda} - 2\boldsymbol{I}_M)\boldsymbol{V}^T$, writing $\boldsymbol{a}^{(t)} = \sum_i c_i^{(t)}\boldsymbol{v}_i$ gives $c_i^{(t+1)} = (\lambda_i - 2)\,c_i^{(t)}$. If $\lambda_i > 3$, $\boldsymbol{v}_i$ dominates as $t \to \infty$. □

In other words, the top eigenvalues of $\boldsymbol{\Sigma}$ (spikes) attract the system state.

3.3 Sample Covariance and Merged Spurious States

We do *not* observe $\boldsymbol{\Sigma}$ directly, but only $\boldsymbol{Q}$. If $\boldsymbol{Q}$ has r outlier eigenvalues, we might expect r stable Hopfield attractors. In practice, multiple signals can *merge* into a single attractor, or extra spurious attractors may appear.

Empirical Example on MNIST. Figure 1 shows how many outlier eigenvalues are detected in $\boldsymbol{Q}$ (i.e. spikes) vs. the number of unique Hopfield converged states, for subsets of MNIST digits. Mergers reduce the final count of stable states. A concrete four-digit illustration (0,4,6,8) in Figure 2 shows how two separate states each blend two digit classes. Recursively splitting each merged state using newly constructed networks eventually recovers distinct directions.

4 Proposed Method

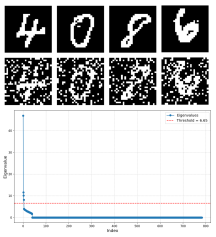
In this section, we present a hierarchical algorithm that uncovers latent signal vectors in a spiked covariance setting by recursively applying Hopfield updates. The procedure alleviates the problem of multiple signals merging into a single memory, ultimately extracting one prototype (i.e., one spiked direction) per latent subspace.

4.1 Problem Formulation

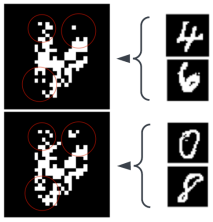
We restate the central problem under the spiked covariance framework. Let $\{\boldsymbol{a}_i\}_{i=1}^N \subset \mathbb{R}^M$ be a dataset of N samples, each modeled as

$$\boldsymbol{a}_i \;=\; \boldsymbol{z}_i + \sum_{l=1}^{r} y_{il}\,\boldsymbol{u}_l,$$

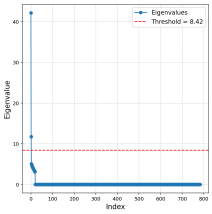
where $\boldsymbol{z}_i$ has i.i.d. zero-mean components (unit variance), and $\{\boldsymbol{u}_l\}$ are underlying latent signals of interest. Our goal is to identify these latent vectors despite finite-sample noise and spurious correlations, ultimately yielding a collection of prototype vectors that capture the data's salient structures.



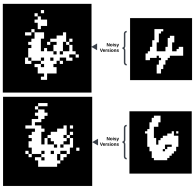
(a) Initial data points and spectrum.



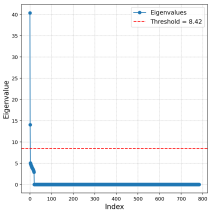
(b) Unique converged states (all samples).



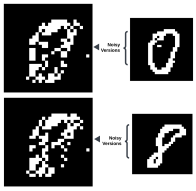
(c) Spectrum after decomposing the first state (digits 4,6).



(d) Converged states after decomposing the first state.



(e) Spectrum after decomposing the second state (digits 0,8).



(f) Converged states after decomposing the second state.

Figure 2: **MNIST example with digits 0,4,6,8**. Hopfield merges them into two states, but rebuilding networks on each subset recovers four directions.

4.2 Hierarchical Hopfield Decomposition (HHD)

We propose **Hierarchical Hopfield Decomposition (HHD)**, which recursively splits data whenever multiple latent signals remain *merged* in a single Hopfield memory. Each recursion level checks how many outlier (spiked) eigenvalues the current sub-dataset has; if more than one spike is found, we apply Hopfield updates to partition the sub-dataset into distinct stable states. The process continues until at most one spike per sub-dataset remains or until depth/size constraints are met.

If the spike count r remains unchanged from the previous recursion, we remove high-entropy samples that align nearly equally with all spiked directions. Formally, for a sample $\boldsymbol{a}_i$, define

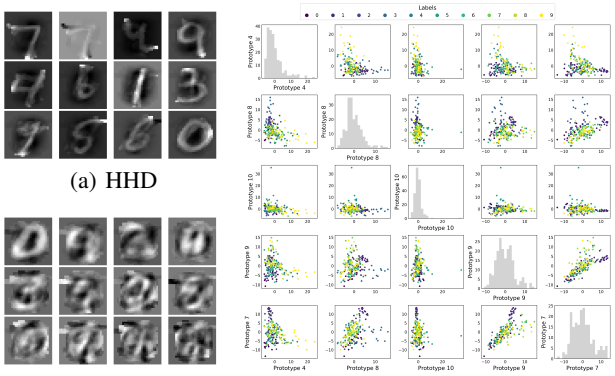
$$p_{ij} = \frac{\exp(\boldsymbol{a}_i^\top \boldsymbol{v}_j)}{\sum_{k=1}^{r} \exp(\boldsymbol{a}_i^\top \boldsymbol{v}_k)}, \quad H_i = -\sum_{j=1}^{r} p_{ij} \log p_{ij},$$

where $\{\boldsymbol{v}_j\}$ are the spiked eigenvectors. Samples with H_i above a threshold are removed, preventing near-degenerate "tie" states in the Hopfield update and allowing further subspace splitting.

Prototype Extraction. Once the recursion terminates in a leaf node, we take the corresponding eigenvector(s) as the final prototype(s). A depth-first traversal gathers these leaf prototypes. Samples can then be assigned to the nearest prototype via cosine similarity or other distance measures.

5 Experiments on Prototype Extraction

We test HHD on MNIST LeCun et al. (1998) and LFW Huang et al. (2008). Figure 3 contrasts HHD prototypes with PCA components on MNIST. Unlike PCA, which enforces orthogonality, HHD can isolate both *correlated* and uncorrelated directions, yielding sharper digit-specific features. Figure 4 shows a similar effect on faces: at higher recursion, prototypes split into more specialized or individual-specific components, revealing a hierarchy from broad facial angles down to unique expressions.



(a) HHD (b) PCA (c) Projection scatter

Figure 3: (a) Prototypes from HHD at various recursion depths. (b) Top principal components from PCA. (c) Scatter plots of data projections onto five HHD prototypes. Certain prototypes share correlated features, while others discriminate different digits.

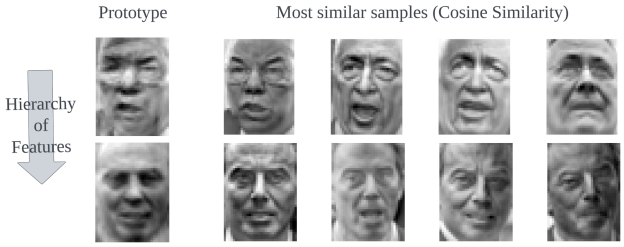


Figure 4: Example of two prototypes extracted from LFW at different recursion levels. Higher-level prototype (top row) captures a broad expression, while deeper recursion (bottom row) yields features of a specific individual.

Clustering. We compare k-means initializations on MNIST subsets: random, k-means++ Arthur & Vassilvitskii (2006), PCA-based, and HHD-based (our prototypes). Table 1 reports V-measure Rosenberg & Hirschberg (2007), ARI Steinley (2004), and AMI Vinh et al. (2010). HHD-based achieves the highest averages, indicating that prototypes aligned with true latent directions help cluster formation.

6 Conclusion

We showed that classical Hopfield networks, via Hebbian coupling, inherently implement a *spiked covariance* model whose outlier eigenvalues correspond to meaningful latent directions. Under finite samples, these directions sometimes merge, yielding spurious states. Proposed Hierarchical Hopfield Decomposition splits merged states, moving to finer spiked subspaces. Experiments elucidate how this approach can be leveraged to isolate correlated signals in high-dimensional data.

Table 1: Mean clustering metrics on 25 random MNIST subsets (150 samples each). Standard deviations in subscripts. The best mean in each column is **bold**.

Method	V-measure	ARI	AMI
K-means++	$0.518_{\pm 0.028}$	$0.232_{\pm 0.046}$	$0.368_{\pm 0.035}$
Random	$0.514_{\pm 0.041}$	$0.219_{\pm 0.049}$	$0.354_{\pm 0.051}$
PCA-based	$0.523_{\pm 0.035}$	$0.249_{\pm 0.043}$	$0.368_{\pm 0.046}$
HHD-based	$\mathbf{0.539}_{\pm 0.034}$	$\mathbf{0.259}_{\pm 0.055}$	$\mathbf{0.386}_{\pm 0.044}$

REFERENCES

E. Agliari, F. Alemanno, M. Aquaro, and A. Fachechi. Regularization, early-stopping and dreaming: A hopfield-like setup to address generalization and overfitting. *Neural Networks*, 177:106389, 2024. ISSN 0893-6080. URL https://doi.org/10.1016/j.neunet.2024.106389.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical Report 2006-13, Stanford InfoLab, June 2006. URL http://ilpubs.stanford.edu:8090/778/.

Marco Benedetti, Louis Carillo, Enzo Marinari, and Marc Mézard. Eigenvector dreaming. *Journal of Statistical Mechanics: Theory and Experiment*, 2024(1):013302, 2024. URL https://doi.org/10.1088/1742-5468/ad138e.

Alex Bloemendal, Antti Knowles, Horng-Tzer Yau, and Jun Yin. On the principal components of sample covariance matrices. *Probability theory and related fields*, 164(1):459–552, 2016. URL https://doi.org/10.1007/s00440-015-0616-x.

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(2):288–299, May 2017. ISSN 1572-9613. URL http://dx.doi.org/10.1007/s10955-017-1806-y.

Xiucai Ding and Fan Yang. Spiked separable covariance matrices and principal components. *The Annals of Statistics*, 49(2):1113 – 1138, 2021. doi: 10.1214/20-AOS1995. URL https://doi.org/10.1214/20-AOS1995.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005. URL https://doi.org/10.4324/9781410612403.

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. URL https://doi.org/10.1073/pnas.79.8.2554.

Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition, 2016. URL https://doi.org/10.48550/arXiv.1606.01164.

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

ROBERTJ McEliece, Edwardc Posner, EUGENER Rodemich, and SANTOSHS Venkatesh. The capacity of the hopfield associative memory. *IEEE transactions on Information Theory*, 33(4):461–482, 1987. URL https://doi.org/10.1109/TIT.1987.1057328.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need, 2021. URL https://doi.org/10.48550/arXiv.2008.02217.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 410–420, 2007.

Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010. URL http://jmlr.org/papers/v11/vinh10a.html.

A Algorithm Pseudocode

We present the pseudocode for Hierarchical Hopfield Decomposition (HHD) here.

Algorithm 1 Hierarchical Hopfield Decomposition

Input: data matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ (N samples as columns), minimum cluster size $C_{\min}$, maximum depth $D_{\max}$.
Output: hierarchical tree of sub-datasets.
Initialize depth $\leftarrow 0,\ r_{\text{prev}} \leftarrow \infty$.
function RecursiveDecompose($\boldsymbol{A}$, depth, r_{prev}):
  $\boldsymbol{Q} = \sqrt{\frac{M}{N}}^{-1} \cdot \frac{\boldsymbol{A}\boldsymbol{A}^\top}{N}$ // scaled sample covariance
  Compute eigenvalues $\lambda_1 \geq \cdots \geq \lambda_M$ of $\boldsymbol{Q}$, and let spikedIdx $\subset \{1, \ldots, M\}$ collect indices of $\lambda_i > \gamma_+$ (section 2.2).
  $r \leftarrow$ |spikedIdx| // number of spiked eigenvalues
  if $(r \leq 1)$ **or** (depth $\geq D_{\max}$) **or** $(N < C_{\min})$ **then**
    return *leaf node* containing {col. indices of $\boldsymbol{A}$} and r.
  if $(r \geq r_{\text{prev}})$ **then**:
    Remove the columns of $\boldsymbol{A}$ with high entropies w.r.t. the spiked eigenvectors
    return RecursiveDecompose($\boldsymbol{A}$, depth, r_{prev})
  // Construct Hopfield coupling matrix
  $\boldsymbol{W} = N^{-1}\boldsymbol{A}\boldsymbol{A}^\top - \boldsymbol{I}_M$.
  // Perform Hopfield updates until convergence
  $\boldsymbol{S}^{(0)} \leftarrow \text{sign}(\boldsymbol{A}^\top)$
  repeat:
    $\boldsymbol{S}^{(\text{new})} = \text{sign}\big(\boldsymbol{W}\,\boldsymbol{S}\big)$
    if $(\boldsymbol{S}^{(\text{new})} == \boldsymbol{S})$ **then break**
    else $\boldsymbol{S} \leftarrow \boldsymbol{S}^{(\text{new})}$
  // Partition data columns by unique stable states:
  Let $\mathcal{U}$ be the set of distinct columns of $\boldsymbol{S}^\top$.
  Initialize $\mathcal{C} \leftarrow \varnothing$.
  for each state $\boldsymbol{u} \in \mathcal{U}$ **do**
    Let $\boldsymbol{A}_u$ be the submatrix of N_u columns of $\boldsymbol{A}$ that satisfy $\boldsymbol{S}^\top[:, i] = \boldsymbol{u}$.
    if $\big(N_u \geq C_{\min}\big)$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\text{RecursiveDecompose}(\boldsymbol{A}_u, \text{depth} + 1, r)\}$
  return $\mathcal{C}$
end function
return RecursiveDecompose$\big(\boldsymbol{A}, 0, \infty\big)$

B Ablation Study on Entropy Removal

We investigated how removing high-entropy samples influences the hierarchical Hopfield decomposition. We compare two conditions:

1. **No Removal:** The algorithm never prunes samples based on entropy.
2. **With Removal:** The algorithm discards columns whose entropy is within 0.1 of the maximum entropy at each recursion step.

Both conditions use the same maximum recursion depth $D_{\max} = 50$ and minimum cluster size $C_{\min} = 1$. In Table 2, we report several tree-level metrics (§4.2) that summarize the final decomposition.

Discussion: Without removing high-entropy samples, the decomposition quickly settles into a small number of leaf nodes at a shallow depth, each leaf containing multiple spikes on average. In contrast, enabling entropy removal forces additional splits, yielding a deeper decomposition (average

Table 2: **Ablation study on entropy removal.** We compare a run with no entropy removal to one that removes any sample whose entropy is within 0.1 of the maximum. Both experiments used $D_{\max} = 50$ and $C_{\min} = 1$.

	#Leaves	Avg. Depth	Avg. Spikes	#Multi-Spike Leaves	Avg. Leaf Size
No Removal	**4**	**1.00**	**6.50**	**2**	**70.00**
With Removal	**27**	**13.41**	**1.30**	**7**	**5.22**

leaf depth of 13.41) and smaller leaf sizes of 5.22. The average number of spikes per leaf is also reduced to about 1.30, indicating that most leaves are nearly single-spike subspaces. Although this process creates more total leaves and sometimes yields leaves with more than one spike (7 multi-spike leaves), the overall decomposition is more fine-grained. Hence, high-entropy sample removal helps avoid early entanglements of signals in a single state, allowing the algorithm to proceed further toward single-spike sub-datasets.

C Discussion and Future Directions

A central advantage of HHD is its hierarchical splitting of data. At higher recursion levels, broad patterns may merge multiple classes; at deeper levels, specialized directions emerge. Depending on the application, one can stop at intermediate depths for semi-granular prototypes or proceed until each sub-dataset is pure. Averaging sub-datasets offers an alternative "mean representative" approach if a single spiked factor is too strict. Future extensions include partial supervision, embedding methods, or alternate definitions of "entropy" to refine the sample-removal step.