Hopfield Boosting for Out-of-Distribution Detection

Claus Hofmann[1 2] **Simon Schmid**[3] **Bernhard Lehner**[1 4] **Daniel Klotz**[5 6] **Sepp Hochreiter**[2 1]

Abstract

Out-of-distribution (OOD) detection is crucial for real-world machine learning. Outlier exposure methods, which use auxiliary outlier data, can significantly enhance OOD detection. We present Hopfield Boosting, a boosting technique employing modern Hopfield energy (MHE) to refine the decision boundary between in-distribution (ID) and OOD data. Our method focuses on challenging outlier examples near the decision boundary, achieving a 40% improvement in FPR95 on CIFAR-10, setting a new state-of-the-art in OOD detection with outlier exposure.

1 Introduction

Effective out-of-distribution (OOD) detection is vital for real-world machine learning systems, as they inevitably encounter data different from their training distribution, which can lead to overconfident and wildly incorrect predictions after deployment. In this paper, we introduce Hopfield Boosting, a novel OOD detection approach that harnesses the energy component of modern Hopfield networks (MHNs; Ramsauer et al., 2021).

In contrast to previous work, which employs modern Hopfield energy (MHE) only at model inference (Zhang et al., 2022), our method uses an auxiliary outlier data set (AUX) to *boost* the model's OOD detection capacity during training. Hopfield Boosting improves OOD detection by allowing to learn a boundary around the in-distribution (ID) data during training.

2 Related Work

Out-of-distribution detection. One common OOD detection method uses statistics from a classifier trained on ID data, like maximum softmax probability (MSP; Hendrycks & Gimpel, 2017) or the classifier's energy score (Liu et al., 2020). Zhang et al. (2022) introduced an OOD detection approach that incorporates MHE, leveraging the entire training data. However, their method doesn't utilize AUX data for additional discrimination.

Various approaches, such as Hendrycks et al. (2019b); Liu et al. (2020); Chen et al. (2021), and Ming et al. (2022), incorporate AUX data to enhance their OOD detection capabilities, leveraging additional information. Usually, only a small subset of AUX samples shares semantic similarity with the ID data, as most are easily distinguishable from it. Some methods, like Chen et al. (2021) and Ming et al. (2022) intentionally select samples near the OOD decision boundary for training, ensuring tighter encapsulation of the ID data.

[1] Silicon Austria Labs, JKU LIT SAL eSPML Lab, Austria

[2] Johannes Kepler University Linz, Institute for Machine Learning, JKU LIT SAL eSPML Lab, Austria

[3] Software Competence Center Hagenberg GmbH, Austria

[4] Johannes Kepler University Linz, JKU LIT SAL eSPML Lab, Austria

[5] ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning, Johannes Kepler University, Linz, Austria

[6] Department of Computational Hydrosystems, Helmholtz Centre for Environmental Research–UFZ, Leipzig, Germany

Correspondence to: Claus Hofmann < hofmann@ml.jku.at >

Associative Memory & Hopfield Networks in 2023. NeurIPS 2023 workshop.

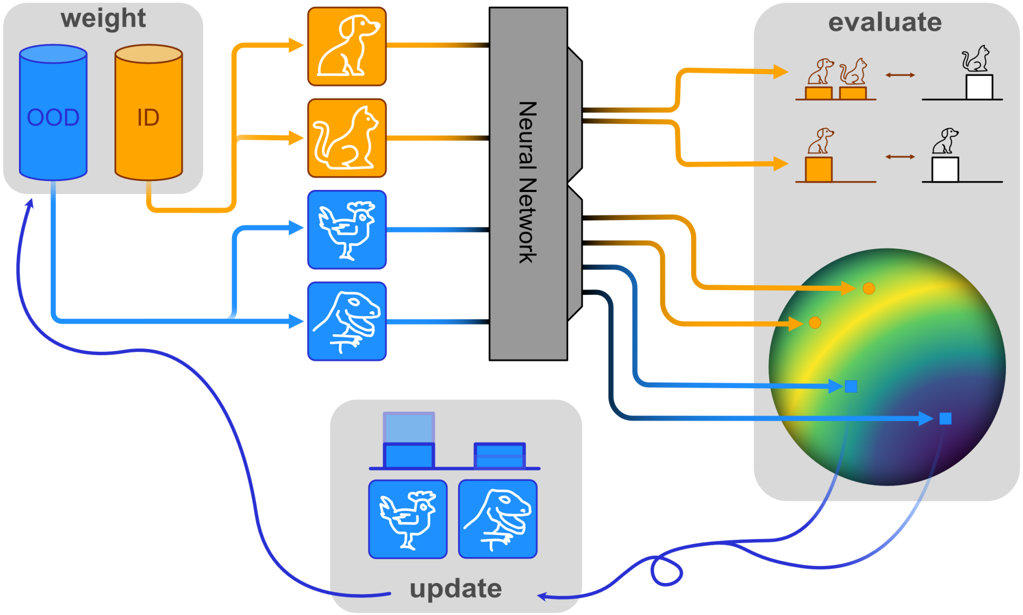


Figure 1: Conceptual depiction of Hopfield Boosting.The first step (weight) creates a minibatch-sample, and thereby weak learners, first by choosing in-distribution samples (ID, orange) without replacement and second by choosing out-of-distribution samples (OOD, blue) with replacement according to their assigned probabilities. The second step (evaluate) computes a composite loss for the resulting predictions (Section 3). And, the third step (update) assigns new probabilities to the OOD samples according to their position on the hypersphere (see Figure 4).

Boosting-based classification. Boosting, in particular, AdaBoost (Freund & Schapire, 1995), directs ensemble learning toward challenging, hard-to-classify data instances. These challenging instances often lie near the maximum margin hyperplane (Rätsch et al., 2001), akin to support vectors in support vector machines (Cortes & Vapnik, 1995).

3 Method

OOD detection framework. The task of OOD detection is usually formulated as a binary classification problem with the classes ID and OOD. It is common practice to use a threshold γ on a score $s(\boldsymbol{\xi})$ to decide whether a given sample $\boldsymbol{\xi}$ is ID or OOD. The estimated class $\hat{B}$ can then be decided by

$$\hat{B}(\boldsymbol{\xi}, \gamma) = \begin{cases} ID & \text{if } s(\boldsymbol{\xi}) \geq \gamma \\ OOD & \text{if } s(\boldsymbol{\xi}) < \gamma \end{cases} \tag{1}$$

It is common to select γ to ensure 95% of new ID samples are correctly classified and to then evaluate the false positive rate on OOD data (FPR95). Metrics like the area under the receiver operating characteristic (AUROC) can be computed directly without the need to specify γ.

Modern Hopfield energy. Given N d-dimensional stored patterns $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N)$ arranged in a matrix $\boldsymbol{X}$, and a d-dimensional query $\boldsymbol{\xi}$, we define the MHE (Ramsauer et al., 2021) as

$$\mathrm{E}(\boldsymbol{\xi}; \boldsymbol{X}) \ = \ - \operatorname{lse}(\beta, \boldsymbol{X}^T \boldsymbol{\xi}) + \ \frac{1}{2} \ \boldsymbol{\xi}^T \boldsymbol{\xi} \ + \beta^{-1} \log N \ + \frac{1}{2} M^2 . \tag{2}$$

where M is the largest norm of a pattern: $M = \max_i ||x_i||$; and lse is the log-sum-exponential:

$$\mathrm{lse}(\beta, \boldsymbol{x}) = \beta^{-1} \log \left(\sum_{i=1}^{N} \exp(\beta x_i) \right) \tag{3}$$

Outlier exposure with MHE. In this section, we will introduce how to shape the energy function to improve the detection of patterns outside the model's training distribution. We train a classifier on the ID data using the standard cross-entropy loss and add an OOD loss that uses the AUX data set (as in Hendrycks et al., 2019b; Liu et al., 2020; Ming et al., 2022):

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{OOD}} \tag{4}$$

We employ a novel OOD loss motivated by MHE. Given an arbitrary sample $\boldsymbol{\xi} \in \mathbb{R}^d$ that does not necessarily have to be contained in $\boldsymbol{I}$ and $\boldsymbol{O}$, the per-sample OOD loss $L_{\text{OOD}}(\boldsymbol{\xi})$ is defined as

$$L_{\text{OOD}}(\boldsymbol{\xi}) = -\,2\ \mathrm{lse}(\beta, (\boldsymbol{I} \parallel \boldsymbol{O})^T \boldsymbol{\xi}) \ + \ \mathrm{lse}(\beta, \boldsymbol{I}^T \boldsymbol{\xi}) \ + \ \mathrm{lse}(\beta, \boldsymbol{O}^T \boldsymbol{\xi}) \tag{5}$$

where $\boldsymbol{I}$ contains ID patterns, $\boldsymbol{O}$ contains AUX patterns, and $(\boldsymbol{I} \parallel \boldsymbol{O})$ denotes the concatenation of the patterns $\boldsymbol{I}$ and $\boldsymbol{O}$ along the axis of the samples. Appendix B.1 displays the energy landscape of $L_{\text{OOD}}(\boldsymbol{\xi})$ using exemplary data on a 3-dimensional sphere. The loss is maximal at the boundary between ID and AUX data and decreases with increasing distance from the decision boundary in both directions. We average the per-sample OOD loss over the samples in $\boldsymbol{I}$ and $\boldsymbol{O}$:

$$\mathcal{L}_{\text{OOD}} = \frac{1}{2N} \sum_{\boldsymbol{\xi} \sim (\boldsymbol{I} \parallel \boldsymbol{O})} L_{\text{OOD}}(\boldsymbol{\xi}) \tag{6}$$

When training a model, we feed batches of N ID samples and N AUX samples to the model and compute the loss using the pairwise similarity matrix. N can be much smaller than the sizes of the full ID or AUX data sets. We normalize the sample vectors in $\boldsymbol{I}$, $\boldsymbol{O}$ and $\boldsymbol{\xi}$ to unit length.

Weight updates. An integral aspect of Hopfield Boosting is the use of weighted samples, akin to AdaBoost (Freund & Schapire, 1995). For an illustrative example of the mechanism, we refer to Appendix B.2. We assign weights, denoted as w, to each training sample from the AUX data set $\boldsymbol{O}$, which collectively form the weight vector $\boldsymbol{w}_t$. Initially, all weight values are set to $\boldsymbol{w}_1 = 1/|\boldsymbol{O}|$. These weights guide the sampling of mini-batches from $\boldsymbol{O}$ during training. We update $\boldsymbol{w}_t$ based on the per-sample loss $L_{\text{OOD}}(\boldsymbol{\xi})$ for all training samples $\boldsymbol{\xi}$, aggregated into the matrix Ξ. The resultant loss vector $L_{\text{OOD}}(\Xi)$ gives the updated weights through normalization by $\mathrm{softmax}$:

$$\boldsymbol{w}_{t+1} = \mathrm{softmax}(\beta L_{\text{OOD}}(\Xi)) \tag{7}$$

Algorithm 1 Hopfield Boosting

```
Require: $T$, $N$, $\boldsymbol{I}$, $\boldsymbol{O}$, $\mathcal{L}_{\text{CE}}$, $L_{\text{OOD}}$, $\beta$
  Set all weights $w_1$ to $1/|\boldsymbol{O}|$
  for $t = 1$ to $T$ do
      1. Weight. Get hypothesis $h_t : \boldsymbol{I}_s \parallel \boldsymbol{O}_s \rightarrow \{ID, OOD\}$ by
          1.a. Minibatch sampling $\boldsymbol{I}_s$ from $\boldsymbol{I}$, and
          1.b. Sub sampling $\boldsymbol{O}_s$ from $\boldsymbol{O}$ according to the weighting $\boldsymbol{w}_t$.
      2. Evaluate. Compute composite loss from Equation (4) on $\boldsymbol{I}_s$ and $\boldsymbol{O}_s$.
      3. Update. Adapt constituents for the next iteration: Update model:
          3.a. At every step, update the full model (backbone, classification head, and MHE).
          3.b. At every $t * N$ step calculate new weights for $\boldsymbol{O}$ with $\boldsymbol{w}_{t+1} = \mathrm{softmax}(\beta L_{\text{OOD}}(\Xi))$.
  end for
```

Summary - Algorithm. Algorithm 1 outlines Hopfield Boosting's process, with each iteration t comprising three steps: Weight, evaluate, and update. We start by randomly sampling a mini-batch

Table 1: Comparison of OOD detection performance on CIFAR-10 of Hopfield Boosting compared to POEM (Ming et al., 2022), EBO (Liu et al., 2020), and MSP-OE (Hendrycks et al., 2019b) on a ResNet-18 encoder. ↓ indicates "lower is better" and ↑ indicates "higher is better". Standard deviations are estimated across five independent training runs.

	Hopfield Boosting (ours)		POEM		EBO		MSP-OE	
OOD Dataset	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
SVHN	$\mathbf{0.23}^{\pm\mathbf{0.08}}$	$\mathbf{99.57}^{\pm\mathbf{0.06}}$	$1.48^{\pm 0.68}$	$99.33^{\pm 0.15}$	$2.67^{\pm 1.01}$	$99.15^{\pm 0.26}$	$4.31^{\pm 1.23}$	$99.20^{\pm 0.17}$
LSUN-Crop	$\mathbf{0.82}^{\pm\mathbf{0.20}}$	$\mathbf{99.40}^{\pm\mathbf{0.05}}$	$4.02^{\pm 0.91}$	$98.89^{\pm 0.15}$	$6.82^{\pm 0.83}$	$98.43^{\pm 0.11}$	$7.01^{\pm 1.28}$	$98.83^{\pm 0.16}$
LSUN-Resize	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$\mathbf{99.98}^{\pm\mathbf{0.02}}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.88^{\pm 0.12}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$\mathbf{99.98}^{\pm\mathbf{0.02}}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.96^{\pm 0.00}$
Textures	$\mathbf{0.16}^{\pm\mathbf{0.02}}$	$\mathbf{99.85}^{\pm\mathbf{0.01}}$	$0.49^{\pm 0.04}$	$99.72^{\pm 0.05}$	$1.11^{\pm 0.18}$	$99.61^{\pm 0.02}$	$2.29^{\pm 0.18}$	$99.57^{\pm 0.02}$
iSUN	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.97^{\pm 0.02}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.87^{\pm 0.12}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$\mathbf{99.98}^{\pm\mathbf{0.02}}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.96^{\pm 0.00}$
Places 365	$\mathbf{4.28}^{\pm\mathbf{0.26}}$	$\mathbf{98.51}^{\pm\mathbf{0.11}}$	$7.7^{\pm 0.68}$	$97.56^{\pm 0.26}$	$11.77^{\pm 0.76}$	$96.39^{\pm 0.33}$	$21.42^{\pm 0.98}$	$95.91^{\pm 0.19}$
Mean	**0.92**	**99.55**	2.28	99.21	3.73	98.92	5.84	98.91

from the ID data set and **weighting** the OOD data set based on $\boldsymbol{w}_t$. Next, we **evaluate** the composite loss on the mini-batch. Lastly, we **update** model parameters and, every Nth step, refresh the AUX data set weights $\boldsymbol{w}_{t+1}$.

Inference. At inference time, the OOD score $s(\boldsymbol{\xi})$ is computed as

$$s(\boldsymbol{\xi}) \;=\; -\,\mathrm{E}(\boldsymbol{\xi};\boldsymbol{I}) \;=\; \mathrm{lse}(\beta, \boldsymbol{I}^T\boldsymbol{\xi}) - \frac{1}{2}\,\boldsymbol{\xi}^T\boldsymbol{\xi} \;-\beta^{-1}\log N \;-\frac{1}{2}M^2 \tag{8}$$

We compute the score $s(\boldsymbol{\xi})$ using the full ID data set $\boldsymbol{I}$. For inference, performance improves when vectors in $\boldsymbol{I}$ and $\boldsymbol{\xi}$ remain unnormalized. We believe that this is because the norms learned by $\mathcal{L}_{\mathrm{CE}}$ contain information that is valuable for OOD detection. This is in contrast to optimizing $\mathcal{L}_{\mathrm{OOD}}$, where missing vector normalization leads to steadily increasing vector norms (see Appendix B.4).

4 Experiments

4.1 Setup

For training and evaluation, following Ming et al. (2022), we train our OOD detection approach using a ResNet-18 (He et al., 2016) encoder on CIFAR-10 (Krizhevsky, 2009) as the ID data set. As the AUX data set, we utilize a downsampled version of ImageNet-RC (Chrabaszcz et al., 2017). To assess OOD detection performance, we employ various OOD data sets with distributions different from both the ID and AUX data sets. For evaluation, we calculate scores $s(\boldsymbol{\xi})$ as defined in Equation (8) and assess the discriminative power of $s(\boldsymbol{\xi})$ between CIFAR-10 and the OOD data sets using the false positive rate at 95% true positives (FPR95) and AUROC.

The network trains for 100 epochs. In each epoch, the model sees the full ID data set and a selection of AUX samples (sampled according to $\boldsymbol{w}_t$). We sample mini-batches of size 128 per data set, resulting in a total batch size of 256. We evaluate the composite loss for each mini-batch and update the model accordingly. After an epoch, we update the sample weights $\boldsymbol{w}_{t+1}$ on 500,000 AUX data instances. During this process, we do not compute gradients or update model parameters. For these weight updates, we fill $\boldsymbol{I}$ and $\boldsymbol{O}$ with the *full* ID data set and a portion of the AUX data set of equal size.

4.2 Results

We compared Hopfield Boosting with three other OOD detection approaches that also use an auxiliary outlier data set: MSP-OE (Hendrycks et al., 2019a), EBO (Liu et al., 2020), and POEM (Ming et al., 2022), which was the previous top-performing method with outlier exposure.

The CIFAR-10 results are summarized in Table 1. Hopfield Boosting achieves equal or superior performance to the other methods in the FPR95 metric on all OOD data sets, outperforming POEM by 40% on the average per-data set FPR95 results. Applying our L_{OOD} only slightly impacts ID classification performance, reducing it from 95.07% without L_{OOD} to 93.38% with L_{OOD}.

5 Conclusion

We introduced Hopfield Boosting, an outlier exposure approach for OOD detection that uses MHE to *boost* a classifier between ID and OOD data, focusing on samples near the decision boundary. Our experiments on an established OOD detection benchmark show Hopfield Boosting as the new state-of-the-art.

We demonstrated how the Hopfield energy-based OOD loss shapes the loss surface, forming a decision boundary between in-distribution and outlier data. Additionally, we showcased how the boosting mechanism sharpens this boundary more effectively than random sampling.

Acknowledgements

The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. We thank the projects AI-MOTION (LIT-2018-6-YOU-212), DeepFlood (LIT-2019-8-YOU-213), Medical Cognitive Computing Center (MC3), INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), EPILEPSIA (FFG-892171), AIRI FG 9-N (FWF-36284, FWF-36235), AI4GreenHeatingGrids(FFG- 899943), INTEGRATE (FFG-892418), ELISE (H2020-ICT-2019-3 ID: 951847), Stars4Waters (HORIZON-CL6-2021-CLIMATE-01-01). We thank Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), University SAL Labs initiative, FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, GLS (Univ. Waterloo) Software Competence Center Hagenberg GmbH, TÜV Austria, Frauscher Sensonic, TRUMPF and the NVIDIA Corporation. This work has been supported by the "University SAL Labs" initiative of Silicon Austria Labs (SAL) and its Austrian partner universities for applied fundamental research for electronic-based systems.

References

Laurence F Abbott and Yair Arian. Storage capacity of generalized networks. *Physical review A*, 36 (10):5091, 1987.

S. Ahn, A. Korattikara, and M. Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1771–1778, Madison, WI, USA, 2012. Omnipress.

Pierre Baldi and Santosh S Venkatesh. Number of stable points for spin-glasses and neural networks of higher orders. *Physical Review Letters*, 58(9):913, 1987.

Barbara Caputo and Heinrich Niemann. Storage capacity of kernel associative memories. In *Artificial Neural Networks—ICANN 2002: International Conference Madrid, Spain, August 28–30, 2002 Proceedings 12*, pp. 51–56. Springer, 2002.

HH Chen, YC Lee, GZ Sun, HY Lee, Tom Maxwell, and C Lee Giles. High order correlation model for associative memory. In *AIP Conference Proceedings*, volume 151, pp. 86–99. American Institute of Physics, 1986.

Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, and Somesh Jha. Atom: Robustifying out-of-distribution detection using outlier mining. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, pp. 430–445. Springer, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pp. 23–37. Springer-Verlag, 1995.

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

E Gardner. Multiconnected neural network models. *Journal of Physics A: Mathematical and General*, 20(11):3453, 1987.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkxzx0NtDB.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019a.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=HyxCxhRcY7.

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.

J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092, 1984. doi: 10.1073/pnas.81.10.3088.

D Horn and M Usher. Capacities of multiconnected memory models. *Journal de Physique*, 49(3): 389–395, 1988.

A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Deptartment of Computer Science, University of Toronto, 2009.

D. Krotov and J. J. Hopfield. Dense associative memory for pattern recognition. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, pp. 1172–1180. Curran Associates, Inc., 2016.

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.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21464–21475. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f5496252609c43eb8a3d147ab9b9c006-Paper.pdf.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Yifei Ming, Ying Fan, and Yixuan Li. POEM: Out-of-distribution detection with posterior sampling. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15650–15665. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ming22a.html.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Demetri Psaltis and Cheol Hoon Park. Nonlinear discriminant functions and associative memories. In *AIP conference Proceedings*, volume 151, pp. 370–375. American Institute of Physics, 1986.

H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, D. Kreil, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. Hopfield networks is all you need. In *9th International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42: 287–320, 2001.

Bernhard Schäfl, Lukas Gruber, Angela Bitto-Nemling, and Sepp Hochreiter. Hopular: Modern hopfield networks for tabular data. *arXiv preprint arXiv:2206.00664*, 2022.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020.

Y. W. Teh, A. H. Thiery, and S. J. Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *J. Mach. Learn. Res.*, 17(1):193–225, 2016.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

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

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688, Madison, WI, USA, 2011. Omnipress.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

P. Xu, J. Chen, D. Zou, and Q. Gu. Global convergence of Langevin dynamics based algorithms for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 3122–3133. Curran Associates, Inc., 2018.

Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, Shi Han, Dongmei Zhang, et al. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2022.

R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

A Details on Continuous Modern Hopfield Networks

The following arguments are adopted from Fürst et al. (2022) and Ramsauer et al. (2021). Associative memory networks have been designed to store and retrieve samples. Hopfield networks are energy-based, binary associative memories, which were popularized as artificial neural network architectures in the 1980s (Hopfield, 1982, 1984). Their storage capacity can be considerably increased by polynomial terms in the energy function (Chen et al., 1986; Psaltis & Park, 1986; Baldi & Venkatesh, 1987; Gardner, 1987; Abbott & Arian, 1987; Horn & Usher, 1988; Caputo & Niemann, 2002; Krotov & Hopfield, 2016). In contrast to these binary memory networks, we use continuous associative memory networks with far higher storage capacity. These networks are continuous and differentiable, retrieve with a single update, and have exponential storage capacity (and are therefore scalable, i.e., able tackle large problems; Ramsauer et al., 2021).

Formally, we denote a set of patterns $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ that are stacked as columns to the matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and a state pattern (query) $\boldsymbol{\xi} \in \mathbb{R}^d$ that represents the current state. The largest norm of a stored pattern is $M = \max_i \|\boldsymbol{x}_i\|$. Then, the energy E of continuous Modern Hopfield Networks with state $\boldsymbol{\xi}$ is defined as (Ramsauer et al., 2021)

$$\mathrm{E} \ = \ - \ \beta^{-1} \ \log \left(\sum_{i=1}^{N} \exp(\beta \boldsymbol{x}_i^T \boldsymbol{\xi}) \right) + \ \frac{1}{2} \ \boldsymbol{\xi}^T \boldsymbol{\xi} \ + \ \mathrm{C}, \tag{9}$$

where $\mathrm{C} = \beta^{-1} \log N \ + \ \frac{1}{2} \ M^2$. For energy E and state $\boldsymbol{\xi}$, Ramsauer et al. (2021) proved that the update rule

$$\boldsymbol{\xi}^{\text{new}} \ = \ \boldsymbol{X} \ \text{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi}) \tag{10}$$

converges globally to stationary points of the energy E and coincides with the attention mechanisms of Transformers (Vaswani et al., 2017; Ramsauer et al., 2021).

The *separation* Δ_i of a pattern $\boldsymbol{x}_i$ is its minimal dot product difference to any of the other patterns:

$$\Delta_i = \min_{j, j \neq i} \left(\boldsymbol{x}_i^T \boldsymbol{x}_i - \boldsymbol{x}_i^T \boldsymbol{x}_j \right) . \tag{11}$$

A pattern is *well-separated* from the data if Δ_i is above a given threshold (specified in Ramsauer et al., 2021). If the patterns $\boldsymbol{x}_i$ are well-separated, the update rule Equation 10 converges to a fixed point close to a stored pattern. If some patterns are similar to one another and, therefore, not well-separated, the update rule converges to a fixed point close to the mean of the similar patterns.

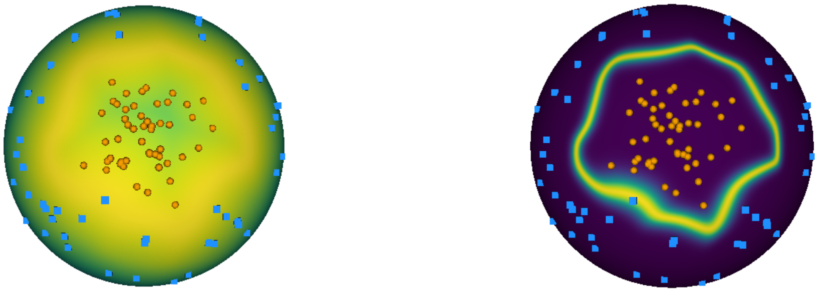
The update rule of a Hopfield network thus identifies sample–sample relations between stored patterns. This enables similarity-based learning methods like nearest neighbor search (see Schäfl et al., 2022), which Hopfield Boosting leverages to detect samples outside the distribution of the training data.

B Toy Examples

This section presents examples to illustrate the main intuitions of Hopfield Boosting. For illustration purposes, the examples do not consider the inlier classification task — i.e., the first term on the right-hand side of equation (4).

B.1 3D visualizations of the OOD Loss

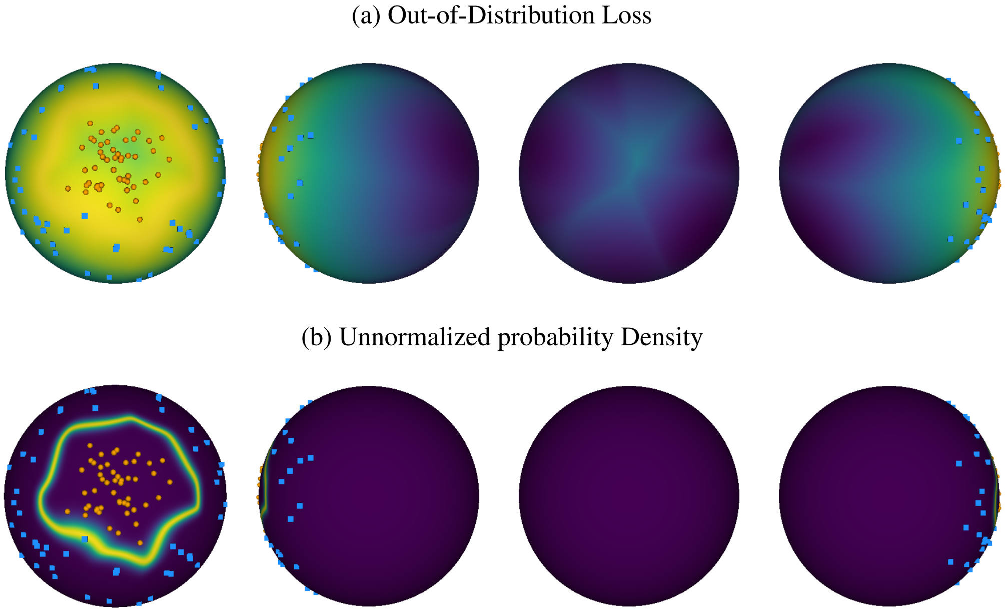
The first example is a 3D sphere that depicts the behavior of the OOD loss (Figure 2). To show how inliers and outliers shape the OOD loss surface, we generated the patterns so that $\boldsymbol{I}$ clusters around the pole while the many outliers populate the large reminding perimeters of the sphere. This is analogous to the idea that one has access to a large AUX data set, where some are more and some less informative for OOD detection (as, for example, conceptualized in Chen et al., 2021; Ming et al., 2022).



(a) Out-of-Distribution Loss

(b) Unnormalized Probability Density

Figure 2: Depiction of the $L_{\text{OOD}}(\boldsymbol{\xi})$ mechanism for separting inliers from outliers. Plot (a) shows $L_{\text{OOD}}(\boldsymbol{\xi})$ on a 3D sphere with exemplary inlier (orange) and outlier (blue) points. Plot (b) the corresponding unnormalized probability density $\exp(\beta L_{\text{OOD}}(\boldsymbol{\xi}))$. In both (a) and (b), β was set to 128.



(a) Out-of-Distribution Loss

(b) Unnormalized probability Density

Figure 3: Depiction of the OOD loss ($L_{\text{OOD}}(\boldsymbol{\xi})$) in Fig. 2 in more orientations. (a) shows $L_{\text{OOD}}(\boldsymbol{\xi})$ with exemplary inlier (orange) and outlier (blue) points; and (b) the corresponding unnormalized probability density $\exp(\beta L_{\text{OOD}}(\boldsymbol{\xi}))$. β was set to 128. Both (a) and (b) rotate the sphere by 0, 90, 180, and 270 degrees around the vertical axis.

B.2 Adaptive resampling

The second example demonstrates how the weighting step in Hopfield Boosting allows good estimations of the decision boundary by sampling a small amount of highly relevant data points (Figure 4). For small, low dimensional data sets, one can always use all the data to compute L_{OOD} (Figure 4, a). For large problems (e.g. Chen et al., 2021; Ming et al., 2022) this strategy becomes difficult. Sampling N data points uniformly at random will then yield many points that are uninformative for fitting a decision boundary (Figure 4, b). In contrast, when using Hopfield Boosting and sampling N data points according to $\boldsymbol{w}_t$, the decision boundary more faithfully represents the decision boundary fitted on the entire data set.

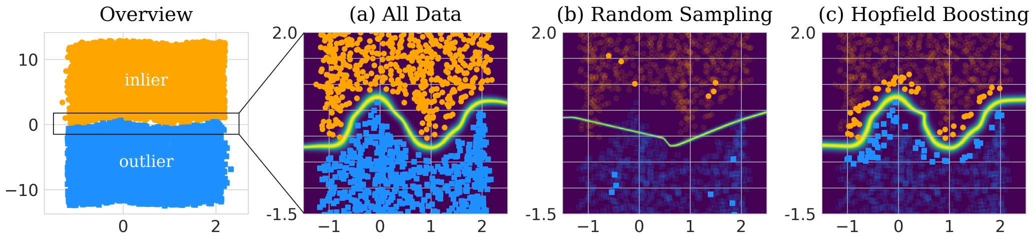


Figure 4: Synthetic example of the adaptive resampling mechanism.

B.3 Effect on Learned Representation

In order to analyze the impact of Hopfield Boosting on learned representations, we utilize the output of our model's embedding layer (see 4.1) as the input for a manifold learning-based visualization. Uniform Manifold Approximation and Projection (UMAP) McInnes et al. (2018) is a non-linear dimensionality reduction technique known for its ability to preserve both global and local structure in high-dimensional data.

First, we train two models – with and without Hopfield Boosting– and extract the embeddings of both ID and OOD data sets from them. This results in a 512-dimensional vector representation for each data point, which we further reduce to two dimensions with UMAP. The training data for UMAP always corresponds to the training data of the respective method. That is, the model trained without Hopfield Boosting is solely trained on CIFAR-10 data, and the model trained with Hopfield Boosting is presented with CIFAR-10 and AUX data during training, respectively. We then compare the learned representations concerning ID and OOD data.

Figure 5 shows the UMAP embeddings of ID (CIFAR-10) and OOD (AUX and SVHN) data based on our model trained without (a) and with Hopfield Boosting (b). Without Hopfield Boosting, OOD data points typically overlap with ID data points, with just a few exceptions, making it difficult to differentiate between them. Conversely, Hopfield Boosting allows to distinctly separate ID and OOD data in the embedding.

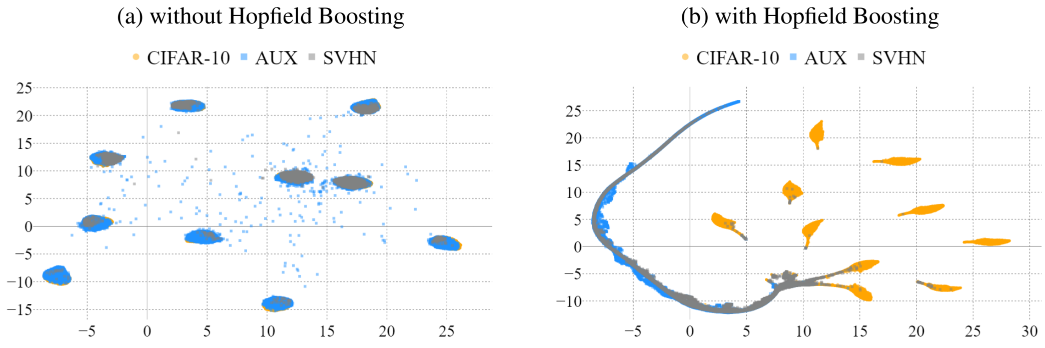
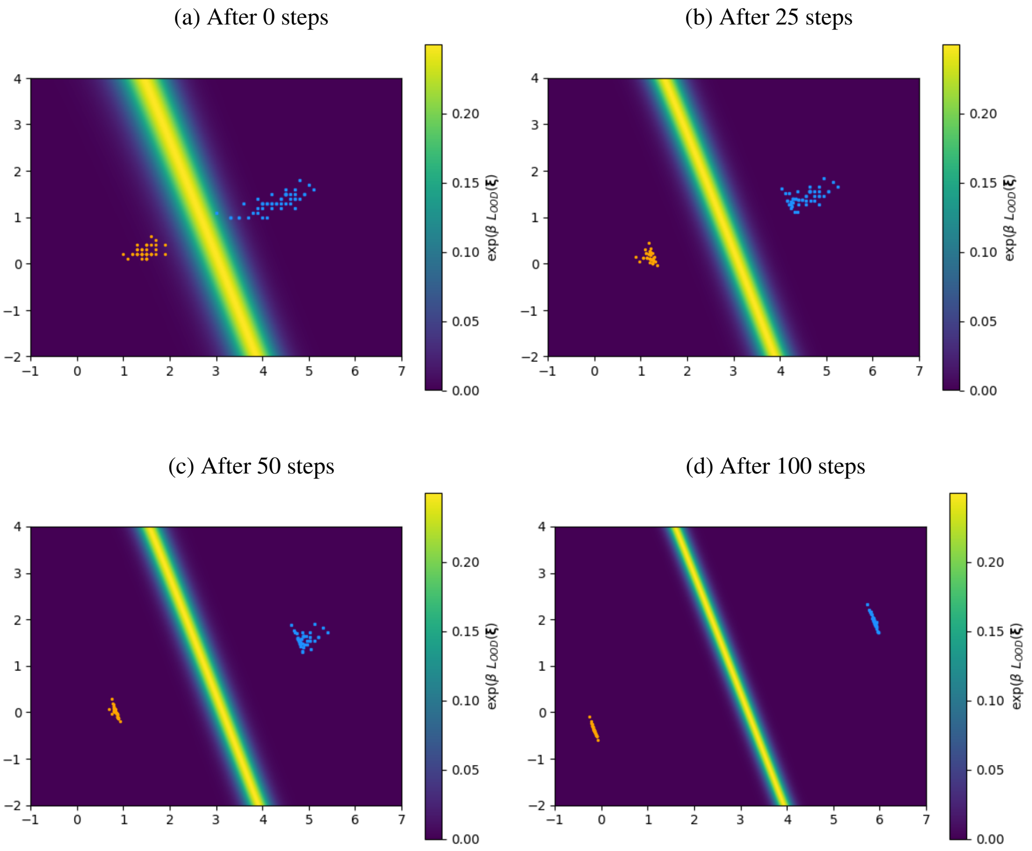


Figure 5: UMAP embeddings of ID (CIFAR-10) and OOD (AUX and SVHN) data based on our model trained without (a) and with Hopfield Boosting (b). Clearly, without Hopfield Boosting, the embedded OOD data points tend to overlap with the ID data points, making it impossible to distinguish between ID and OOD. On the other hand, Hopfield Boosting shows a clear separation of ID and OOD data in the embedding.

B.4 Dynamics of $\mathcal{L}_{\text{OOD}}$ on Patterns in Euclidean Space


(a) After 0 steps

(b) After 25 steps

(c) After 50 steps

(d) After 100 steps

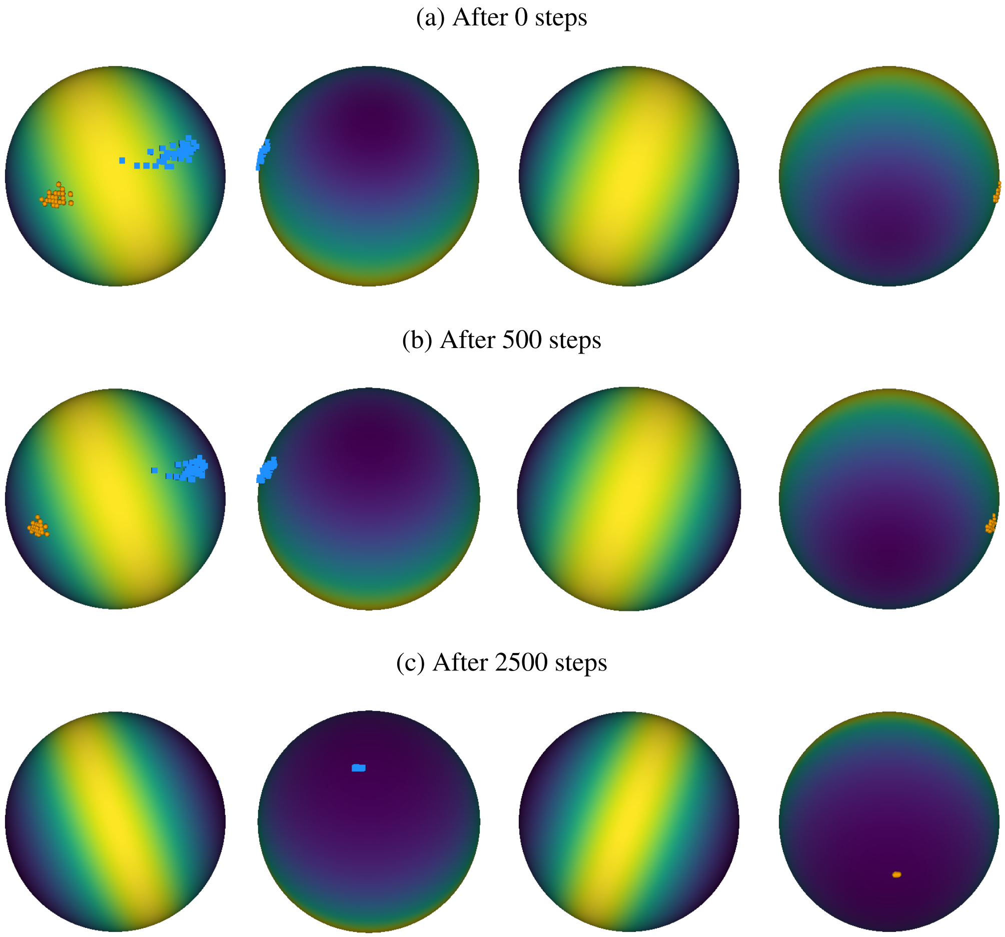
Figure 6: $\mathcal{L}_{\text{OOD}}$ applied to exemplary data points on euclidean space. Gradient updates are applied to the data points directly. We observe that the variance orthogonal to the decision boundary shrinks while the variance parallel to the decision boundary does not change to this extent. β is set to 2.

In this example, we applied our out-of-distribution loss $\mathcal{L}_{\text{OOD}}$ on a simple binary classification problem. As we are working in Euclidean space and not on a sphere, we use a modified version of MHE, which uses the negative Euclidean distance instead of the dot-product-similarity. For the formal relation between Equation (12) and MHE, we refer to Appendix C.1:

$$E(\boldsymbol{\xi};\boldsymbol{X}) = -\beta^{-1}\log\left(\sum_{i=1}^{N}\exp(-\frac{\beta}{2}\,||\boldsymbol{\xi}-\boldsymbol{x}_i||_2^2)\right) \qquad (12)$$

Figure 6a shows the initial state of the patterns and the decision boundary $\exp(\beta L_{\text{OOD}}(\boldsymbol{\xi}))$. We store the samples of the two classes as stored patterns in $\boldsymbol{I}$ and $\boldsymbol{O}$, respectively, and compute the per-sample loss from Equation (5) for all samples. We then set the learning rate to 0.1 and perform gradient descent with $\mathcal{L}_{\text{OOD}}$ on the data points. Figure 6b shows that after 25 steps, the distance between the data points and the decision boundary has increased, especially for samples that had previously been close to the decision boundary. After 100 steps, as shown in Figure 6d, the variability orthogonal to the decision boundary has almost completely vanished, while the variability parallel to the decision boundary is maintained.

B.5 Dynamics of $\mathcal{L}_{\text{OOD}}$ on Patterns on the Sphere



(a) After 0 steps

(b) After 500 steps

(c) After 2500 steps

Figure 7: $\mathcal{L}_{\text{OOD}}$ applied to exemplary data points on a sphere. Gradients are applied to the data points directly. We observe that the geometry of the space forces the patterns to opposing poles of the sphere.

C Hopfield Boosting in Relation

C.1 Relation to Radial Basis Function Networks

This section shows the relation between radial basis function networks (RBF networks) and modern Hopfield energy (following Schäfl et al., 2022). RBF networks have previously been used as hypotheses for boosting (e.g., Rätsch et al., 2001). We define an RBF network as follows:

$$\phi(\boldsymbol{\xi}) = \sum_{i=1}^{N} w_i \exp(-\frac{\beta}{2}||\boldsymbol{\xi} - \boldsymbol{\mu}_i||_2^2) \tag{13}$$

where w_i are normalized weights

$$w_i = \text{softmax}(\beta \boldsymbol{a})_i = \frac{\exp(\beta a_i)}{\sum_{j=1}^{N} \exp(\beta a_j)} \tag{14}$$

An energy can be obtained by taking the negative $\log$ of $\phi(\boldsymbol{\xi})$

$$
\begin{aligned}
\mathrm{E}(\boldsymbol{\xi}) \; &= -\, \beta^{-1} \, \log\left(\phi(\boldsymbol{\xi})\right) && (15)\\
&= -\, \beta^{-1} \, \log\left(\sum_{i=1}^{N} w_i \exp(-\frac{\beta}{2} \, ||\boldsymbol{\xi} - \boldsymbol{\mu}_i||_2^2)) \right) && (16)\\
&= -\, \beta^{-1} \, \log\left(\sum_{i=1}^{N} \exp(\, \beta(-\frac{1}{2}||\boldsymbol{\xi} - \boldsymbol{\mu}_i||_2^2 + \beta^{-1} \log \mathrm{softmax}(\beta \boldsymbol{a})_i) \,) \right) && (17)\\
&= -\, \beta^{-1} \, \log\left(\sum_{i=1}^{N} \exp(\, \beta(-\frac{1}{2}||\boldsymbol{\xi} - \boldsymbol{\mu}_i||_2^2 + a_i - \mathrm{lse}(\beta, \boldsymbol{a}) \,) \right) && (18)\\
&= -\, \beta^{-1} \, \log\left(\sum_{i=1}^{N} \exp(\, \beta(-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi} + \boldsymbol{\mu}_i^T \boldsymbol{\xi} - \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\mu}_i + a_i) \,) \right) + \mathrm{lse}(\beta, \boldsymbol{a}) && (19)
\end{aligned}
$$

Next, we set $a_i = \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\mu}_i$

$$
\mathrm{E}(\boldsymbol{\xi}) \; = \; -\, \beta^{-1} \, \log\left(\sum_{i=1}^{N} \exp(\beta \boldsymbol{\mu}_i^T \boldsymbol{\xi}) \right) + \frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi} + \mathrm{lse}(\beta, \boldsymbol{a}) \tag{20}
$$

Finally, we use the fact that $\mathrm{lse}(\beta, \boldsymbol{a}) \leq \max_i a_i + \beta^{-1} \log N$

$$
\mathrm{E}(\boldsymbol{\xi}) \; = \; -\, \beta^{-1} \, \log\left(\sum_{i=1}^{N} \exp(\beta \boldsymbol{\mu}_i^T \boldsymbol{\xi}) \right) + \frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi} + \beta^{-1} \log N + \frac{1}{2} M^2 \tag{21}
$$

where $M = \max_i ||\boldsymbol{\mu}_i||$

C.2 Contrastive Representation Learning

A commonly used loss function in contrastive representation learning (e.g., Chen et al., 2020; He et al., 2020) is the InfoNCE loss (Oord et al., 2018):

$$
\mathcal{L}_{\text{NCE}} = \underset{\substack{(x,y)\sim p_{\text{pos}} \\ \{x_i^-\}_{i=1}^{M} \sim p_{\text{data}}}}{\mathbb{E}} \left[-\log \frac{e^{f(x)^T f(y)/\tau}}{e^{f(x)^T f(y)/\tau} + \sum_i e^{f(x_i^-)^T f(y)/\tau}} \right] \tag{22}
$$

Wang & Isola (2020) show that $\mathcal{L}_{\text{NCE}}$ optimizes two objectives:

$$
\mathcal{L}_{\text{NCE}} = \underbrace{\underset{(x,y)\sim p_{pos}}{\mathbb{E}} \left[-f(x)^T f(y)/\tau \right]}_{\text{Alignment}} + \underbrace{\underset{\substack{(x,y)\sim p_{pos} \\ \{x_i^-\}_{i=1}^{M} \sim p_{data}}}{\mathbb{E}} \left[\log\left(e^{f(x)^T f(y)/\tau} + \sum_i e^{f(x_i^-)^T f(x)/\tau} \right) \right]}_{\text{Uniformity}} \tag{23}
$$

Alignment enforces that features from positive pairs are similar, while uniformity encourages a uniform distribution of the samples over the hypersphere.

In comparison, our proposed loss, $\mathcal{L}_{\text{OOD}}$, does not visibly enforce alignment between samples within the same class. Instead, we can observe that it promotes uniformity to the instances of the *foreign* class. Due to the constraints that are imposed by the geometry of the space the optimization is performed on, that is, $||f(x)|| = 1$ when the samples move on a hypersphere, the loss encourages the patterns in the ID data have maximum distance to the samples of the AUX data, i.e., they concentrate on opposing poles of the hypersphere. A demonstration of this mechanism can be found in Appendix B.4 and B.5

Table 2: Comparison of OOD detection performance on CIFAR-100 of Hopfield Boosting compared to POEM (Ming et al., 2022), EBO (Liu et al., 2020), and MSP-OE (Hendrycks et al., 2019b) on a ResNet-18 encoder. ↓ indicates "lower is better" and ↑ indicates "higher is better". Standard deviations are estimated across five independent training runs.

	Hopfield Boosting (ours)		POEM		EBO		MSP-OE	
OOD Dataset	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
SVHN	$\mathbf{8.65}^{\pm\mathbf{2.57}}$	$\mathbf{97.68}^{\pm\mathbf{0.34}}$	$33.59^{\pm 4.12}$	$94.06^{\pm 0.51}$	$36.33^{\pm 3.30}$	$92.93^{\pm 0.80}$	$19.86^{\pm 7.71}$	$95.74^{\pm 1.78}$
LSUN-Crop	$\mathbf{10.87}^{\pm\mathbf{3.03}}$	$\mathbf{97.15}^{\pm\mathbf{0.61}}$	$15.72^{\pm 3.46}$	$96.85^{\pm 0.60}$	$21.06^{\pm 3.49}$	$95.79^{\pm 0.69}$	$32.88^{\pm 1.43}$	$92.85^{\pm 0.37}$
LSUN-Resize	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.55^{\pm 0.10}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.57^{\pm 0.09}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.57^{\pm 0.03}$	$0.03^{\pm 0.02}$	$\mathbf{99.97}^{\pm\mathbf{0.00}}$
Textures	$3.26^{\pm 0.20}$	$98.94^{\pm 0.07}$	$\mathbf{2.89}^{\pm\mathbf{0.32}}$	$\mathbf{98.97}^{\pm\mathbf{0.08}}$	$5.07^{\pm 0.60}$	$98.15^{\pm 0.17}$	$10.34^{\pm 0.44}$	$97.42^{\pm 0.09}$
iSUN	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.56^{\pm 0.09}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.59^{\pm 0.09}$	$\mathbf{0.00}^{\pm\mathbf{0.00}}$	$99.57^{\pm 0.03}$	$0.08^{\pm 0.03}$	$\mathbf{99.96}^{\pm\mathbf{0.01}}$
Places 365	$19.86^{\pm 0.96}$	$\mathbf{95.60}^{\pm\mathbf{0.19}}$	$\mathbf{18.39}^{\pm\mathbf{0.68}}$	$95.03^{\pm 0.71}$	$26.68^{\pm 2.44}$	$91.35^{\pm 0.78}$	$45.96^{\pm 0.95}$	$87.77^{\pm 0.17}$
Mean	**7.11**	**98.08**	11.38	97.35	14.86	96.23	18.19	95.62

D Proofs

Lemma D.1. *We consider the energy function*

$$\begin{aligned}
\mathrm{E}(\boldsymbol{\xi}) &= -\,\beta^{-1}\,\log\big(\,\exp(\,\beta\,\mathrm{lse}(\beta, \boldsymbol{I}^T\boldsymbol{\xi})\,) + \exp(\,\beta\,\mathrm{lse}(\beta, \boldsymbol{O}^T\boldsymbol{\xi})\,)\,\big) && (24)\\
&= -\,\beta^{-1}\,\log\left(\exp\left(\beta\,\beta^{-1}\log\left(\sum_{i=1}^{N}\exp(\beta\boldsymbol{i}_i^T\boldsymbol{\xi})\right)\right) + \exp(\,\beta\,\beta^{-1}\log\left(\sum_{i=1}^{N}\exp(\beta\boldsymbol{o}_i^T\boldsymbol{\xi})\right)\right) && (25)\\
&= -\,\beta^{-1}\,\log\left(\sum_{i=1}^{N}\exp(\beta\boldsymbol{i}_i^T\boldsymbol{\xi}) + \sum_{i=1}^{N}\exp(\beta\boldsymbol{o}_i^T\boldsymbol{\xi})\right) && (26)\\
&= -\,\mathrm{lse}(\beta, (\boldsymbol{I} \parallel \boldsymbol{O})^T\boldsymbol{\xi}) && (27)
\end{aligned}$$

E Further Experimental Details

E.1 Hyperparameter Selection

We use a separate validation process with an AUX data set for model selection. That is, we validate the model on MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and ImageNet-RC with different augmentations than in training, as well as Gaussian and uniform noise.

We set the initial learning rate to 0.1 and decay it during the training process with a cosine schedule without warm restarts (Loshchilov & Hutter, 2016). We use a single value for β throughout the training and evaluation process and for all OOD data sets. We tuned the value of β for each in-distribution data set separately by selecting the value of β from the set $\{2, 4, 8\}$ that performed best in the validation process. We set λ, the weight for the OOD loss $\mathcal{L}_{\mathrm{OOD}}$ to 1.

E.2 LSUN-Resize and iSUN datasets

What is striking is that all methods achieve perfect FPR95 results on the LSUN-Resize and iSUN data sets. The LSUN-Resize data set has previously been found to give misleading results due to image artifacts resulting from the resizing procedure (Tack et al., 2020). We hypothesize that there exists a similar issue with the iSUN data set, as in our experiments, LSUN-Resize and iSUN behaved in a very similar fashion.

E.3 Results on CIFAR-100

On CIFAR-100 (see Table 2), we observe that Hopfield Boosting surpasses the previously best method by a relative 37% on the mean FPR95 metric. The improvement in the SVHN data set is especially striking, improving the FPR95 from 33.59 to 8.65 compared to POEM. For the LSUN-Resize and iSUN data sets we observe a similar behavior to the results from CIFAR-10, where all methods achieve a perfect result on the FPR95 metric.

F Notes on Langevin Sampling

Another method that is appropriate for earlier acquired models is to sample the posterior via the Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011). This method is efficient since it iteratively learns from small mini-batches Welling & Teh (2011); Ahn et al. (2012). See basic work on Langevin dynamics Welling & Teh (2011); Ahn et al. (2012); Teh et al. (2016); Xu et al. (2018). A cyclical stepsize schedule for SGLD was very promising for uncertainty quantification Zhang et al. (2020). Larger steps discover new modes, while smaller steps characterize each mode and perform the posterior sampling.

G Classifiers - energy perspective

It is known that deep classifiers can be considered energy-based models (see LeCun et al. (2006); Grathwohl et al. (2020)). In the following, we will shortly introduce the energy-based perspective. Assume we are given a neural network-based classifier C modeling $p(y|\boldsymbol{\xi})$ with $y \in \mathcal{Y}$ where $\mathcal{Y}$ is a discrete set of K classes. We denote the output of the final linear layer of the classifier (i.e., the logits) as $\boldsymbol{l}(\boldsymbol{\xi}) = \boldsymbol{W}^T \boldsymbol{\xi}$ where $\boldsymbol{W} \in \mathbb{R}^{D \times K}$ are learnable parameters and $\boldsymbol{\xi} \in \mathbb{R}^D$ is the output of the network's penultimate layer. Thus, the logit of $\boldsymbol{\xi}$ being assigned to a specific, given y is $l(\boldsymbol{\xi}, y) = (\boldsymbol{W}^T \boldsymbol{\xi})_y$. The energy $E(\boldsymbol{\xi}, y)$ is defined as the negative logit:

$$E(\boldsymbol{\xi}, y) = -l(\boldsymbol{\xi}, y) = -(\boldsymbol{W}^T \boldsymbol{\xi})_y \tag{28}$$

An energy is connected to a probability density via

$$p(\boldsymbol{\xi}, y) = \frac{1}{Z} \exp(-\beta E(\boldsymbol{\xi}, y)) \tag{29}$$

where β is the inverse temperature and Z is the so-called partition function: $Z = \sum_{y \in \mathcal{Y}} \int_{\boldsymbol{\xi}} \exp(-\beta E(\boldsymbol{\xi}, y))$. Dividing by Z ensures that the probability density $p(\boldsymbol{\xi}, y)$ is normalized (i.e., integrates to 1). By rearranging the terms of Equation (29) we obtain

$$E(\boldsymbol{\xi}, y) = -\beta^{-1} \log p(\boldsymbol{\xi}, y) - \beta^{-1} \log Z \tag{30}$$

Under the assumption that $\boldsymbol{\xi}$ can be assigned to one of the classes in $\mathcal{Y}$ (i.e. $\boldsymbol{\xi}$ is in-distribution) we have that

$$p(\boldsymbol{\xi}) = \sum_{y \in \mathcal{Y}} p(\boldsymbol{\xi}, y) = \sum_{y \in \mathcal{Y}} \frac{1}{Z} \exp(-\beta E(\boldsymbol{\xi}, y)) = \frac{1}{Z} \sum_{y \in \mathcal{Y}} \exp(-\beta E(\boldsymbol{\xi}, y)) \tag{31}$$

and therefore the energy $E(\boldsymbol{\xi})$ corresponding to the marginal probability $p(\boldsymbol{\xi})$ equates to

$$E(\boldsymbol{\xi}) = -\beta^{-1} \log p(\boldsymbol{\xi}) - \beta^{-1} \log Z = -\beta^{-1} \ \log \left(\sum_{y \in \mathcal{Y}} \exp(-\beta E(\boldsymbol{\xi}, y)) \right) \tag{32}$$

$$E(\boldsymbol{\xi}) = -\mathrm{lse}(\beta, \boldsymbol{l}(\boldsymbol{\xi})) = -\mathrm{lse}(\beta, \boldsymbol{W}^T \boldsymbol{\xi}) \tag{33}$$

In the context of out-of-distribution detection, the energy $E(\boldsymbol{\xi})$ can be interpreted as follows: When a sample $\boldsymbol{\xi}$ is in-distribution, the energy $E(\boldsymbol{\xi})$ will be low as there is at least one class y whose logit $(\boldsymbol{W}^T \boldsymbol{\xi})_y$ is high. In contrast, for out-of-distribution samples the energy $E(\boldsymbol{\xi})$ will be high, because all logits $(\boldsymbol{W}^T \boldsymbol{\xi})_y$ are low if there is no object visible that matches any of the classes. Therefore, $s(\boldsymbol{\xi}) = -E(\boldsymbol{\xi})$ is a suitable score for out-of-distribution detection (Liu et al., 2020). However, they do not take the full in-distribution data set into account.