HEAT: Hybrid Energy Based Model in the Feature Space for Out-of-Distribution Detection

Marc Lafon [1]
marc.lafon@lecnam.net

Clément Rambour [1]
clement.rambour@cnam.fr

Nicolas Thome [2]
nicolas.thome@isir.upmc.fr

[1]CEDRIC, Conservatoire National des Arts et Métiers, Paris, France
[2]ISIR, Sorbonne Université, Paris, France

Abstract

In this work, we study the out-of-distribution (OOD) detection problem through the use of the feature space of a pre-trained deep classifier. We show that learning the density of in-distribution (ID) features with an energy-based models (EBM) leads to competitive detection results. However, we found that the non-mixing of MCMC sampling during the EBM's training undermines its detection performance. To overcome this, we introduce HEAT, an energy-based correction of a mixture of class-conditional Gaussian distributions. We show that HEAT obtains favorable results when compared to a strong baseline like the KNN detector on the CIFAR-10/CIFAR-100 OOD detection benchmarks.

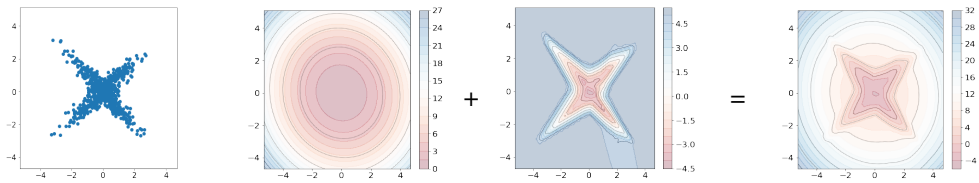
1 Introduction

Out-of-distribution (OOD) detection is a major safety requirement for the deployment of deep learning models in critical applications. In this work, we focus on out-of-distribution detection methods that use a pre-train classifier and we assume that OOD samples are not available neither during the training of the classifier nor to train the OOD detector.

Lee *et al.* [16] have proposed the Mahalanobis detector which amounts to model the in-distribution (ID) features with a mixture of class-conditional Gaussian distributions (MoG). In [24], the authors have shown that using the Mahalanobis distance on normalized features improves performance both for a supervised and self-supervised backbone. Recently, the authors in [26] have pointed out that a simple k-nearest-neighbors distance greatly improves detection performance, especially for OOD samples lying in the vicinity of ID samples. This result suggests that the Gaussian assumption may not be necessarily verified. Nonetheless, the Mahalanobis detector performs better on samples that are far-away from the ID samples. Hence, we would like to overcome the limitation of the Mahalanobis detector while conserving its strengths.

While showing great success on several image modeling tasks [5, 6, 7], energy-based models (EBMs) [14] are still under-explored to estimate the density of ID samples in the feature space of a pre-trained classifier. An EBM is an unnormalized density model entirely defined by an energy function. It can be learned by maximum likelihood estimation (MLE), which consists in diminishing the energy of real samples and increasing the energy of synthesized samples. These synthesized samples are generally generated via a gradient-based Markov Chain Monte Carlo (MCMC) sampling such as Stochastic Gradient Langevin Dynamics (SGLD) [28]. However, gradient-based MCMC sampling is known to struggle to sample all the modes during training, especially in high-dimensional space, thus harming density estimation.

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Toy dataset (b) Gaussian (c) Energy correction (d) HEAT

Figure 1: **Illustration of HEAT on a 2D toy dataset**. The energy from the Gaussian (a) is corrected by the energy-based model (b) to produce the energy of HEAT (c)

Building upon ideas from energy-based correction literature [1, 3, 6, 10, 22, 30], we propose to learn an energy-based model to refine a mixture of class-conditional Gaussian distributions in the feature space of a pre-trained classifier. The mixture of Gaussian distributions allows to cover all the modes and ensures a decreasing density function as we move away from the training data. On the other hand, the energy-based model will reshape the Gaussian densities, as illustrated in Figure 1 on a toy dataset, and will be less dependent on the mixing of MCMC sampling.

Contributions. Our contributions are threefold: (**1**) we are, to the best of our knowledge, the first to show that training an EBM in the feature space leads to competitive detection performance (**2**) we introduce HEAT, an hybrid energy-based model that improves both the Mahalanobis distance and the EBM, (**3**) we show that HEAT obtains comparable results on the CIFAR-10 and CIFAR-100 OOD detection benchmarks with a strong baselines like the KNN detector.

2 Hybrid energy-based model in feature space

We aim to estimate the density in the feature space $\mathcal{Z}$ of a pre-trained deep classifier $f = h \circ \phi$, where $\phi : \mathcal{X} \to \mathcal{Z}$ is the feature extractor and $h : \mathcal{Z} \to \mathcal{Y}$ the classification head. We suppose that we only have access to the in-distribution dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ used to train the classifier f.

2.1 Model

We would like to combine in one model the performance of the Mahalanobis detector on OOD examples that are far away from the ID data, and the flexibility of energy-based models to improve detection of closer OOD samples. To this end, we introduce HEAT, an energy-based correction of a reference distribution $q(\boldsymbol{z})$ which is a mixture of class-conditional Gaussian distributions:

$$p_\theta(\boldsymbol{z}) \propto \exp\left(-E_\theta(\boldsymbol{z})\right) q(\boldsymbol{z}), \quad \text{with} \quad q(\boldsymbol{z}) = \sum\nolimits_c \pi_c \mathcal{N}(\boldsymbol{z}; \mu_c, \Sigma), \tag{1}$$

where the energy-correction E_θ is a neural network with parameter θ, and where we use the empirical estimates for the parameters of the mixture and assume a tied covariance matrix, like in [16] (more details in appendix A.3). In the sequel, we will write $q(\boldsymbol{z}) \propto \exp\left(-E_{\mathcal{G}}(\boldsymbol{z})\right)$ with $E_{\mathcal{G}}(\boldsymbol{z}) = -\log \sum_c \exp\left(-(\boldsymbol{z}-\mu_c)^T \Sigma^{-1} (\boldsymbol{z}-\mu_c)\right)$ so that HEAT can be expressed as $p_\theta(\boldsymbol{z}) \propto \exp\left(-(E_\theta(\boldsymbol{z}) + E_{\mathcal{G}}(\boldsymbol{z}))\right)$.

2.2 Training

We train HEAT via maximum likelihood estimation (MLE) which amounts to perform stochastic gradient descent with the following loss[1]:

$$\mathcal{L}_{MLE} = \mathbb{E}_{\boldsymbol{z} \sim p_{in}}\left[E_\theta(\boldsymbol{z})\right] - \mathbb{E}_{\boldsymbol{z}' \sim p_\theta}\left[E_\theta(\boldsymbol{z}')\right], \tag{2}$$

where we write $\boldsymbol{z} \sim p_{in}$ for $\boldsymbol{z} = \phi(\boldsymbol{x})$ with $\boldsymbol{x} \sim p_{\text{data}}$. Note that the energy inside the expectations in (2) should be $E_\theta + E_{\mathcal{G}}$, but we discarded $E_{\mathcal{G}}$ as it does not depend on θ.

SGLD sampling. To compute $\mathcal{L}_{MLE}$ we must sample synthetic features with respect to our current model p_θ. We follow previous works [5, 9] and use stochastic gradient Langevin dynamics (SGLD) sampling [28]. An SGLD iteration step writes

$$\boldsymbol{z}_0 \sim q, \qquad \boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \alpha_t \nabla_{\boldsymbol{z}} (E_\theta + E_{\mathcal{G}})(\boldsymbol{z}_t) + \sqrt{\beta_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{3}$$

[1] see more details on EBMs training in the appendix

where α_t is the step size and β_t is the noise scale.

Because we would like to rely on $E_{\mathcal{G}}$ as we move away from ID samples, we initiate the SGLD samples with the reference distribution q instead of a normal or uniform distribution as usually done in the EBM literature. This way, E_θ will refine $E_{\mathcal{G}}$ only on high density areas. We stress the importance of taking gradient steps with respect to $E_\theta + E_{\mathcal{G}}$ in equation (3), contrarily with equation (2) where we could ignore $E_{\mathcal{G}}$.

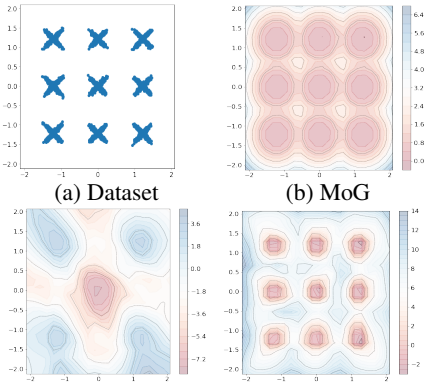
L_2-regularization. Without any modification to the loss, E_θ could dominate $E_{\mathcal{G}}$ which is an undesirable behavior. To avoid this, we add an $L2$-regularization on the magnitudes of E_θ which also helps stabilize training. The final loss is then $\mathcal{L}_{Tot} = \mathcal{L}_{MLE} + \alpha\, \mathcal{L}_{Reg}$, where α is an hyper-parameter controlling the strength of the L_2-regularization.

3 Results

In this section, we first present visual results on a 2D synthetic example to give a better grasp of our approach. Then, we compare HEAT against state-of-the-art baselines for OOD detection on image datasets and perform ablation experiments.

3.1 Synthetic datasets

First, we design a simple 2D synthetic dataset consisting of data points sampled from a "crossed" distribution (Fig. 1a). In Figure 1, we can visualize the result of training HEAT on this dataset. The energy of the HEAT model is the sum of the Gaussian energy and of the neural network's energy which learns a residual to produce the final energy. We can see in Figure 1d that the energy learned by HEAT is sharp nearby the training data and becomes smoother as we move away from the data.



(a) Dataset (b) MoG (c) EBM (d) HEAT

Figure 2: EBM's energy is biased due to SGLD non-mixing

The second synthetic experiment is designed to illustrate that EBMs tend to learn a biased energy function when the MCMC is not mixing, especially when there are multiple modes in the target distribution. We consider a dataset consisting of 9 cross datasets distributed according to a 3×3 grid (Fig. 2a). We can see on Figure 2c that the EBM struggles to cover all the modes while the mixture of Gaussian (Fig. 2b) and HEAT (Fig. 2d) both perfectly recover all the modes.

3.2 Image datasets

Baselines. We evaluate our approach against the following baselines: the maximum softmax probability (MSP) [11], ODIN [17], and Energy-logits [19]. We also compare against methods that aim to estimate the density of in-distribution samples in the feature space: SSD [24] and KNN [26]. Since they obtain better performances with normalized features (i.e. $\boldsymbol{z} = \phi(\boldsymbol{x})/\|\phi(\boldsymbol{x})\|$), we use normalize features for all such methods (including ours).

Evaluation metrics. We report the following standard metrics used in the literature [11]: the area under the receiver operating characteristic curve (AUC) and the false positive rate at a threshold corresponding to a true positive rate of 95% (FPR95).

Datasets. We conduct experiments using CIFAR-10 and CIFAR-100 datasets [13] as in-distribution datasets. For OOD datasets, we define three categories: near-OOD datasets, mid-OOD datasets and far-OOD datasets. These correspond to different levels of proximity with the ID datasets. For CIFAR-10 (resp. CIFAR-100), we consider `TinyImageNet`[2] and CIFAR-100 (resp. CIFAR-10) as near-OOD datasets. Then for both CIFAR-10 and CIFAR-100, we use `LSUN` [31] and `Places365` [32] datasets as mid-OOD datasets, and `Textures` [2] and `SVHN` [21] as far-OOD datasets.

[2] https://www.kaggle.com/c/tiny-imagenet

Table 1: **Results on CIFAR-10 & CIFAR-100.** All methods are based on a pre-trained ResNet-34 trained on the ID dataset only. ↑ indicates larger is better and ↓ the opposite. Best results are in bold.

Method	*Near-OOD* FPR95↓ / AUC ↑	*Mid-OOD* FPR95↓ / AUC ↑	*Far-OOD* FPR95↓ / AUC ↑	**Average** FPR95↓ / AUC ↑
MSP	57.0 / 88.0	51.6 / 91.0	36.0 / 94.4	48.2 / 91.2
ODIN	45.3 / 86.7	**34.1** / 91.3	18.6 / 95.5	**32.6** / 91.1
Energy-Logits	45.2 / 87.6	**34.7** / 91.8	17.8 / 96.0	**32.6** / 91.8
KNN	**42.3 / 90.5**	37.0 / **93.4**	16.7 / 97.3	**32.9 / 93.7**
SSD	51.8 / 89.3	46.8 / 91.8	**7.0 / 98.8**	35.1 / **93.3**
HEAT (ours)	47.5 / 89.9	41.0 / 92.5	8.7 / **98.6**	**32.4 / 93.7**

(a) CIFAR-10

Method	*Near-OOD* FPR95↓ / AUC ↑	*Mid-OOD* FPR95↓ / AUC ↑	*Far-OOD* FPR95↓ / AUC ↑	**Average** FPR95↓ / AUC ↑
MSP	79.2 / 77.1	**82.3 / 75.6**	67.1 / 83.7	76.1 / 78.8
ODIN	80.1 / 76.3	84.4 / 73.3	71.6 / 82.8	78.6 / 77.5
Energy-Logits	80.0 / 76.7	85.4 / 73.2	57.7 / 87.1	74.3 / 79.0
KNN	**78.5 / 78.6**	85.8 / **75.7**	47.7 / 90.8	70.7 / **82.0**
SSD	84.2 / 75.3	86.2 / 74.2	28.3 / **94.4**	66.2 / 81.3
HEAT (ours)	82.9 / 76.3	85.2 / 74.8	**27.0 / 94.6**	**65.0 / 81.9**

(b) CIFAR-100

Implementation details. All experiments were conducted using `PyTorch` [23]. We use a ResNet-34 classifier from the `timm` library [29] for both ID datasets. The classifier is trained for 200 epochs using SGD with momentum and learning rate 0.1. HEAT consists in a 6 layers MLP trained for 20 epochs with Adam with learning rate $5e$-6 (all training details are available in the appendix).

HEAT achieves good OOD detection performance. We can see in Table 1 that HEAT achieves the best average results on the CIFAR-100 benchmark and obtains average results which are comparable to the strong baseline KNN on the CIFAR-10 benchmark. As expected, HEAT and SSD behave similarly on far-OOD datasets, but HEAT improves over SSD on near and mid-OOD datasets. For instance, for mid-OOD datasets, HEAT reduces the FPR95 by 12.4% on CIFAR-10 and increases the AUC from 91.8% to 92.8% on CIFAR-10. On CIFAR-100, HEAT improves only marginally upon SSD and remains worse than KNN on near-OOD datasets and achieves comparable results on mid-OOD datasets.

Ablation. To assess the importance of our hybrid approach, we evaluate HEAT without its hybrid component. We thus drop the reference measure q in equation (1) and initiate the SGLD samples in (3) with a random normal distribution $z_0 \sim \mathcal{N}(0, I)$. In that case we recover a standard EBM trained with MLE via MCMC sampling. From Table 2, we can see that even without its hybrid component, the EBM achieves competitive performance with respect to previous methods on the CIFAR-10 benchmark. However, The EBM struggles to detect far-OOD samples on the CIFAR-100 benchmark. This confirms the qualitative insights we had drawn in section 3.1 as well as our design choices.

Table 2: Ablation on CIFAR-10 & CIFAR-100

Method	*Near-OOD* FPR95↓ / AUC ↑	*Mid-OOD* FPR95↓ / AUC ↑	*Far-OOD* FPR95↓ / AUC ↑	**Average** FPR95↓ / AUC ↑
HEAT w/o energy	51.8 / 89.3	46.8 / 91.8	**7.0 / 98.8**	35.1 / 93.3
HEAT w/o MoG	**44.5 / 90.2**	**33.5 / 93.2**	15.7 / 97.3	31.3 / 93.5
HEAT	47.5 / 89.9	41.0 / 92.5	8.7 / **98.6**	32.4 / **93.7**

(a) CIFAR-10

Method	*Near-OOD* FPR95↓ / AUC ↑	*Mid-OOD* FPR95↓ / AUC ↑	*Far-OOD* FPR95↓ / AUC ↑	**Average** FPR95↓ / AUC ↑
HEAT w/o energy	84.2 / 75.3	86.2 / 74.2	**28.3 / 94.4**	66.2 / 81.3
HEAT w/o MoG	**80.9** / 75.7	85.3 / 73.4	63.0 / 87.6	76.4 / 78.9
HEAT	82.9 / **76.3**	**85.2 / 74.8**	**27.0 / 94.6**	**65.0 / 81.9**

(b) CIFAR-100

4 Conclusion

We proposed HEAT, an hybrid energy-based model trained solely on in-distribution features of a pre-trained classifier for out-of-distribution detection. We showed that HEAT improves or is on par with strong baselines like SSD of KNN on the CIFAR-10 and CIFAR-100 OOD detection benchmarks. In future work, we plan to evaluate HEAT in a low data regime scenario, with different classifier backbones, on more challenging images datasets like Imagenet [4] and/or to extend it to tasks beyond image classification.

References

[1] Michael Arbel, Liang Zhou, and Arthur Gretton. Generalized Energy Based Models. In *International Conference on Learning Representations*, 2021. 2

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3, 10

[3] Jifeng Dai, Yang Lu, and Ying Nian Wu. Generative modeling of convolutional neural networks. *Statistics and its Interface*, 9(4):485–496, 2016. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 4

[5] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy-based models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 1, 2, 8

[6] Ruiqi Gao, Erik Nijkamp, Diederik P. Kingma, Zhen Xu, Andrew M. Dai, and Ying Nian Wu. Flow Contrastive Estimation of Energy-Based Models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2020. 1, 2

[7] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning Energy-based Models by Diffusion Recovery Likelihood. In *9th International Conference on Learning Representations*, Virtual Event, Austria, 2021. OpenReview.net. 1

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 9

[9] Will Grathwohl, Kuan-Chieh Wang, Jörn-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 *8th International Conference on Learning Representations*, 2020. 2, 8

[10] Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012. 2

[11] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of International Conference on Learning Representations*, 2017. 3, 9, 11

[12] Geoffrey E Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 1800(14):1771–1800, 2002. 8

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, pages 32–33, 2009. 3, 10

[14] Yann LeCun, Sumit Chopra, Raia Hadsell, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006. 1

[15] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018. 9

[16] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018. 1, 2

[17] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of International Conference on Learning Representations*, 2018. 3, 11

[18] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing The Reliability of Out-Of-Distribution Image Detection in Neural Networks. In *ICLR*, 2018. 9

[19] Weitang Liu, Xiaoyun Wang, John D Owens, and Yixuan Li. Energy-based Out-of-distribution Detection. In *Advances in Neural Information Processing Systems*, 2020. 3, 11

[20] Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010. 8

[21] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 3, 10

[22] Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, Srinivas Vasudevan, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. MCMC should mix: Learning energy-based model with neural transport latent space MCMC. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4, 10

[24] Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A unified framework for self-supervised outlier detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1, 3, 9, 10, 11

[25] Yang Song and Diederik P. Kingma. How to train your energy-based models. *CoRR*, abs/2101.03288, 2021. 8

[26] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 20827–20840. PMLR, 2022. 1, 3, 10, 11

[27] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. *Proceedings of the 25th International Conference on Machine Learning*, pages 1064–1071, 2008. 8

[28] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688, 2011. 1, 2, 8

[29] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 4, 10

[30] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2635–2644. JMLR.org, 2016. 2

[31] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3, 10

[32] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3, 10

A Appendix

A.1 Energy-based models

An energy-based model (EBM) is an unnormalized density model defined via its energy function $E_\theta : \mathbb{R}^m \to \mathbb{R}$ parameterized by a neural network with parameters θ. For $\boldsymbol{z} \in \mathbb{R}^m$, its probability density is given by the Boltzmann distribution

$$p_\theta(\boldsymbol{z}) = \frac{1}{Z_\theta} \exp\left(-E_\theta(\boldsymbol{z})\right), \tag{4}$$

where Z_θ is the partition function which is intractable in high dimension. We can train EBMs via maximum likelihood estimation:

$$\arg\max_\theta \log p_\theta(\mathcal{D}) = \arg\min_\theta \mathbb{E}_{\boldsymbol{z}\sim p_{in}}[-\log p_\theta(\boldsymbol{z})] \tag{5}$$

which can be approximated via stochastic gradient descent :

$$\theta_{i+1} = \theta_i - \lambda \nabla_\theta(-\log p_{\theta_i}(\boldsymbol{z})) \quad \text{with} \quad \boldsymbol{z} \sim p_{in} \tag{6}$$

Interestingly, $\nabla_\theta(-\log p_{\theta_i}(\boldsymbol{z}))$ can be computed without computing the intractable normalization constant Z_θ.

We have

$$\begin{aligned}
\nabla_\theta(-\log p_\theta(\boldsymbol{z})) &= \nabla_\theta E_\theta(\boldsymbol{z}) + \nabla_\theta \log Z_\theta \\
&= \nabla_\theta E_\theta(\boldsymbol{z}) + \frac{1}{Z_\theta}\nabla_\theta Z_\theta \\
&= \nabla_\theta E_\theta(\boldsymbol{z}) + \frac{1}{Z_\theta}\nabla_\theta \int_{\boldsymbol{z}} \exp(-E_\theta(\boldsymbol{z}))d\boldsymbol{z} \\
&= \nabla_\theta E_\theta(\boldsymbol{z}) + \frac{1}{Z_\theta}\int_{\boldsymbol{z}} \nabla_\theta \exp(-E_\theta(\boldsymbol{z}))d\boldsymbol{z} \\
&= \nabla_\theta E_\theta(\boldsymbol{z}) + \int_{\boldsymbol{z}} -\nabla_\theta E_\theta(\boldsymbol{z}) \frac{\exp(-E_\theta(\boldsymbol{z}))}{Z_\theta} d\boldsymbol{z} \\
&= \nabla_\theta E_\theta(\boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}'\sim p_\theta}[\nabla_\theta E_\theta(\boldsymbol{z}')].
\end{aligned}$$

Therefore, training EBMs via maximum likelihood estimation (MLE) amounts to perform stochastic gradient descent with the following loss:

$$\mathcal{L}_{MLE} = \mathbb{E}_{\boldsymbol{z}\sim p_{in}}\left[E_\theta(\boldsymbol{z})\right] - \mathbb{E}_{\boldsymbol{z}'\sim p_\theta}\left[E_\theta(\boldsymbol{z}')\right]. \tag{7}$$

Intuitively, this loss amounts to diminishing the energy for samples from the true data distribution $p(x)$ and to increasing the energy for synthesized examples sampled according from the current model. Eventually, the gradients of the energy function will be equivalent for samples from the model and the true data distribution and the loss term will be zero.

The expectation $\mathbb{E}_{\boldsymbol{z}'\sim p_\theta}\left[E_\theta(\boldsymbol{z}')\right]$ can be approximated through MCMC sampling, but we need to sample z' from the model p_θ which is an unknown moving density. To estimate the expectation under p_θ in the right hand-side of equation (7) we must sample according to the energy-based model p_θ. To generate synthesized examples from p_θ, we can use gradient-based MCMC sampling such as Stochastic Gradient Langevin Dynamics (SGLD) [28] or Hamiltonian Monte Carlo (HMC) [20]. In this work, we use SGLD sampling following [5, 9]. In SGLD, initial features are sampled from a proposal distribution p_0 and are updated for T steps with the following iterative rule:

$$\boldsymbol{z}_0 \sim p_0, \qquad \boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \alpha_t \nabla_{\boldsymbol{z}} E_\theta(\boldsymbol{z}_t) + \sqrt{\beta_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{8}$$

where α_t is the step size and β_t the noise scale. Therefore sampling from p_θ does not require to compute the normalization constant Z_θ either.

Many variants of this training procedure have been proposed including Contrastive Divergence (CD) [12] where $p_0 = p_{\text{data}}$, or Persistent Contrastive Divergence (PCD) [27] which uses a buffer to extend the length of the MCMC chains. We refer the reader to [25] for more details on EBM training with MLE as well as other alternative training strategies (score-matching, noise contrastive estimation, Stein discrepancy minimization, etc.).

A.2 OOD detection with energy-based models

Once trained on in-distribution features we use the learned energy as an uncertainty score to detect out-of-distribution samples. Given an input sample $\boldsymbol{x}^*$, we compute its feature representation $\boldsymbol{z}^* = \phi(\boldsymbol{x}^*)$ and the decision function for out-of-distribution detection is given by $G(\boldsymbol{z}^*) = \mathbf{1}\{E_\theta(\boldsymbol{z}^*) + E_{MoG}(\boldsymbol{z}^*) \geq \gamma\}$, where γ is a threshold which can be chosen so that at least 95 % of the in-distribution examples are correctly classified.

A.3 Mixture of Gaussian component details

In section 2.1 of the main paper, we introduced our model as an energy-based correction of a mixture of class-conditional Gaussian distributions. Here, we provide additional details on how we compute its parameters. The mixture writes

$$q(\boldsymbol{z}) = \sum_c \pi_c \mathcal{N}(\boldsymbol{z}; \mu_c, \Sigma), \tag{9}$$

with π_c the mixing coefficient, and μ_c and Σ are the means and the tied covariance matrix of the Gaussian distributions. All these parameters are estimated using in-distribution features:

$$\pi_c = \frac{N_c}{N}, \quad \mu_c = \frac{1}{N_c} \sum_{i:y_i=c} \boldsymbol{z}_i, \quad \Sigma = \frac{1}{N} \sum_c \sum_{i:y_i=c} (\boldsymbol{z}_i - \mu_c)(\boldsymbol{z}_i - \mu_c)^T, \tag{10}$$

where N_c is the number of training samples of class c and $\boldsymbol{z}_i = \phi(\boldsymbol{x}_i)$ for $\boldsymbol{x}_i \in \mathcal{D}$.

A.4 More details about baselines

In this section we give more details about the baseline methods we used for the comparative experiment on the image datasets (sec. 3.2 of the main paper).

MSP. Hendrycks *et al.* [11] have proposed to use the maximum softmax probability of the classifier as a baseline to detect OOD detection.

ODIN. [18] is a threshold-based OOD detector enhancing the MSP detector with temperature scaling and inverse adversarial perturbation. Both techniques aimed to increase in-distribution MSP higher than out-distribution MSP. Temperature scaling changes the softmax probabilities using a temperature $T > 0$:

$$S(\boldsymbol{x}; T) = \max_c \frac{\exp(f_c(\boldsymbol{x}, \theta)/T)}{\sum_{k=1}^K \exp(f_k(\boldsymbol{x}, \theta)/T)} \tag{11}$$

and input preprocessing consists in adding a small adversarial perturbation to samples:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \epsilon \, \mathrm{sign}(-\nabla_{\boldsymbol{x}} \log S(\boldsymbol{x}; T)) \tag{12}$$

SSD. Lee *et al.* [15] have proposed to fit class-conditional Gaussian distributions with a tied covariance matrix on the penultimate layer of the neural network classifier, and to compute the maximum Mahalanobis distance to each Gaussian center as the anomaly score:

$$M(\boldsymbol{x}) = \min_c -(\phi(\boldsymbol{x}) - \mu_c)^T \Sigma^{-1} (\phi(\boldsymbol{x}) - \mu_c). \tag{13}$$

In this work we implemented the SSD [24] score which is the Mahalanobis distance on normalized features.

In the spirit of ODIN, the authors in [15] also use adversarial perturbation with the Mahalanobis score:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \, \mathrm{sign}(\nabla_{\boldsymbol{x}} M(\boldsymbol{x})) \tag{14}$$

where ϵ is the strength of the perturbation and chosen to separate ID samples from OOD samples or negative samples generated by FGSM [8]. The authors have also reported improved detection performance using feature ensembling to combine the anomaly scores computed at several layers of the classifier into one unique score. We let the exploration of these techniques for future work.

KNN. Sun *et al.* [26] proposed non-parametric density estimation using nearest neighbors in the feature space of a pre-trained classifier for OOD detection. For test image $\boldsymbol{x}^*$, they compute its feature representation $\boldsymbol{z}^*$ and then the Euclidean distances between $\boldsymbol{z}^*$ and each feature of the training dataset. The uncertainty score they use for OOD detection is the negative k^{th} distance, with k an hyper-parameter.

Feature normalization. In the OOD detection with KNN paper [26], the author showed that feature normalization is critical for good performance. Similarly, [24] used normalized features for the Mahalanobis score. In the case of [24], the feature normalization was a byproduct of the self-supervised contrastive learning framework of their work and was not mentioned as a essential element. Their implementation using the supervised loss also used feature normalization. In this work, all results for Mahalanobis, KNN and HEAT are obtained with normalized feature, that is when we write z we assume $z = \phi(\boldsymbol{x})/\|\phi(\boldsymbol{x})\|$.

A.5 Experimental setup

A.5.1 Training details

In this section we provide more training details. All experiments were conducted using the `PyTorch` [23] library.

ID classifier training We use a ResNet-34 classifier from the `timm` library [29] for both CIFAR-10 and CIFAR-100 datasets. The classifier is trained for 200 epochs using SGD with Nesterov momentum and weight decay. The momentum factor is 0.9 and the weight decay coefficient is $5e$-4. The learning rate is initialized at 0.1 and reduced by a factor of 10 at 50% and 75% of training. We use random resized crops and random horizontal flips with default parameters on images as a form of data augmentation.

HEAT training HEAT consists in an MLP with 4 hidden layers trained for 20 epochs with Adam with learning rate $5e$-6. The network input dimension is 512 (which is the dimension of the penultimate layer of ResNet-34), the hidden dimension is 1024 and the output dimension is 1. For SGLD sampling, we use 20 steps with an initial step size of $1e$-6 linearly decayed to $1e$-7 and an initial noise scale of $1e$-3 linearly decayed to $1e$-4. We add a small Gaussian noise with std $1e$-3 to each input of the network to stabilize training as done in previous work. The L_2 coefficient is set to 10. We use temperature scaling on the mixture of Gaussian distributions energy with temperature $T_{\mathcal{G}} = 1e3$. The hyperparameters for the CIFAR-10 and CIFAR-100 models are identical.

EBM training For the EBM we conserve the same network architecture than for HEAT. We trained for 20 epochs with Adam with learning rate $5e$-5. For SGLD sampling, we use 200 steps with an initial step size of $1e$-2 linearly decayed to $1e$-3 and an initial noise scale of $1e$-2 linearly decayed to $1e$-3. We add a small Gaussian noise with std $1e$-3 to each input of the network to stabilize training as done in previous work. We use temperature scaling on neural network's energy with temperature $T = 1e$-2. The L_2 coefficient is set to 0.1. The hyperparameters for the CIFAR-10 and CIFAR-100 models are identical.

A.5.2 Datasets

ID datasets In section 3.2 of the main paper, the experiments are conducted using CIFAR-10 and CIFAR-100 datasets [13]. They consist in 32×32 natural images with 10 classes for CIFAR-10 and 100 classes for CIFAR-100. Both datasets contain 50,000 training images and 10,000 test images.

OOD datasets We consider the following OOD datasets: `TinyImageNet`, `LSUN` [31], `Places` [32], `Textures` [2] and `SVHN` [21]. We use the test set of each of the previous dataset for OOD detection. `TinyImageNet` is a subset of ImageNet containing images of 200 different classes, `LSUN` contains images from 10 scene categories, `Places365` contains images from 365 scene categories, `Textures` contains images representing 47 different texture categories and `SVHN` contains images representing street view house numbers (10 categories).

A.6 Detailed results

In this section we provide detailed results for each OOD datasets in Table 3 and Table 4.

Method	*Near-OOD*		*Mid-OOD*		*Far-OOD*		
	C-100	**TinyIN**	**LSUN**	**Places365**	**Textures**	**SVHN**	**Average**
	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑
MSP [11]	58.0 / 87.9	55.9 / 88.2	50.5 / 91.9	52.7 / 90.2	52.3 / 91.7	19.7 / 97.0	48.2 / 91.2
ODIN [17]	48.4 / 86.0	42.2 / 87.3	**32.6** / 92.3	35.6 / 90.4	29.4 / 92.6	7.8 / 98.3	32.6 / 91.1
Energy-Logits [19]	48.4 / 86.9	**41.9** / 88.2	33.7 / 92.6	35.7 / 91.0	30.7 / 92.9	4.9 / 99.0	32.6 / 91.8
KNN [26]	**47.9 / 90.3**	42.6 / **90.6**	36.1 / **94.1**	37.8 / **92.6**	25.2 / 96.0	8.1 / 98.6	32.9 / **93.7**
SSD [24]	52.6 / 89.0	50.9 / 89.5	47.1 / 92.4	46.4 / 91.2	**13.1 / 97.8**	**0.9 / 99.8**	35.1 / 93.3
EBM (ours)	**47.4** / 89.9	**41.6 / 90.4**	**32.7 / 93.9**	**34.3 / 92.4**	25.7 / 95.6	5.7 / 98.9	**31.3 / 93.5**
HEAT (ours)	50.0 / 89.6	45.0 / 90.2	40.7 / 93.1	41.4 / 91.9	16.4 / 97.2	1.1 / **99.8**	**32.4 / 93.7**

Table 3: **OOD detection results on CIFAR-10.**

Method	*Near-OOD*		*Mid-OOD*		*Far-OOD*		
	C-10	**TinyIN**	**LSUN**	**Places365**	**Textures**	**SVHN**	**Average**
	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑	FPR95↓ / AUC↑
MSP [11]	**80.0 / 76.6**	78.3 / 77.6	**83.5** / 74.7	**81.0 / 76.4**	72.1 / 81.0	62.0 / 86.4	76.1 / 78.8
ODIN [17]	81.4 / 76.4	78.7 / 76.2	86.1 / 72.0	82.6 / 74.5	62.4 / 85.2	80.7 / 80.4	78.6 / 77.5
Energy-Logits [19]	**80.6 / 76.9**	79.4 / 76.5	87.6 / 71.7	83.1 / 74.7	62.4 / 85.2	53.0 / 88.9	74.3 / 79.0
KNN [26]	81.0 / 76.1	**76.0 / 81.1**	89.5 / **75.7**	82.1 / **75.7**	58.2 / 88.0	37.2 / 93.6	70.7 / **82.0**
SSD [24]	85.6 / 73.6	82.7 / 77.0	87.8 / 73.8	84.6 / 74.5	36.6 / **92.4**	19.9 / **96.4**	66.2 / 81.3
EBM (ours)	83.7 / 72.6	78.1 / 78.8	87.6 / 72.6	83.0 / 74.2	73.7 / 84.8	52.2 / 90.4	76.4 / 78.9
HEAT (ours)	85.0 / 74.2	80.8 / 78.3	86.9 / 74.2	83.4 / 75.3	**35.6 / 92.6**	**18.3 / 96.6**	**65.0 / 81.9**

Table 4: **OOD detection results on CIFAR-100.**