Your classifier is secretly an energy based model and you should treat it like one (ICLR 2020)

2020-04-24
hyounguk.shon@kaist.ac.kr

Abstract

- This paper proposes Joint Energy-based Model. (JEM)
- JEM models the joint distribution $p(\boldsymbol{x}, y)$ by reinterpreting softmax-based classifier architecture as an energy-based model (EBM).
- JEM is trained to estimate (unnormalized) $p(\boldsymbol{x})$ and $p(y|\boldsymbol{x})$. Therefore, JEM can be trained on both unlabeled and labeled data.
- JEM improves classifiers in terms of calibration, robustness, OOD detection, while enabling it to generate samples.
- JEM achieves performance rivaling the SOTA in both generative and discriminative tasks with a single model.

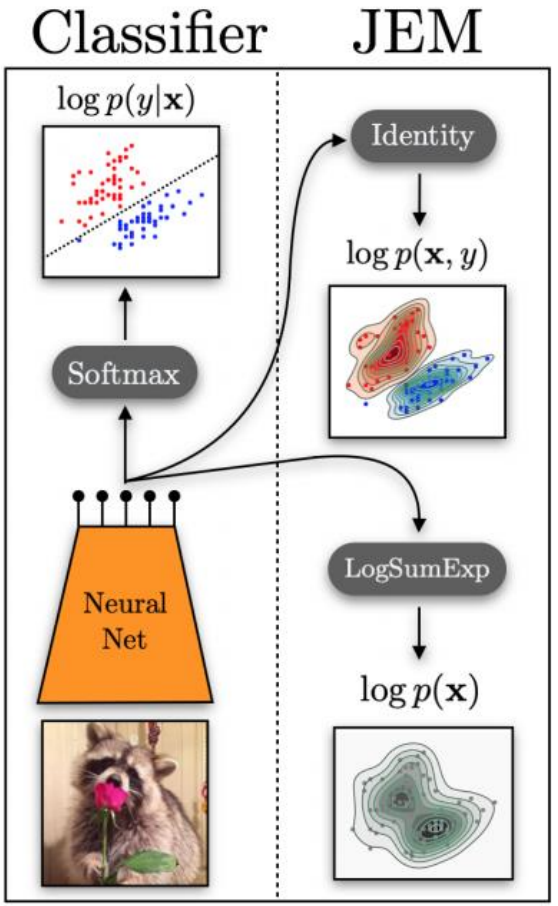


Figure 1: Visualization of our method, JEM, which defines a joint EBM from classifier architectures.

Introduction to EBM (I)

- Energy-based Model (EBM) [1]:
  - EBM estimates a probability density function by introducing an energy function $E_\theta(\boldsymbol{x})$.
  - Any arbitrary $p(\boldsymbol{x})$ can be modeled by a function $E_\theta(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}$.

$$p_\theta(\boldsymbol{x}) \equiv \frac{\exp\left(-E_\theta(\boldsymbol{x})\right)}{Z_\theta} \qquad \text{where } Z_\theta = \int_{\boldsymbol{x}} \exp\left(-E_\theta(\boldsymbol{x})\right)$$

- Inference and Training on EBM:
  - Notice that Z_θ is a normalizing constant. For most choices of $E_\theta(\boldsymbol{x})$, it is difficult to compute or estimate Z_θ. Therefore it is intractable to compute the normalized likelihood.
  - Therefore, we must rely on other methods to train EBMs.

[1] Y LeCun et al., A tutorial on energy-based learning. Predicting structured data, 2006.

Introduction to EBM (II)

- The gradient of the log-likelihood is expressed as follows:

$$\frac{\partial \log p_\theta(\boldsymbol{x})}{\partial \theta} = \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}\left[\frac{\partial E_\theta(\boldsymbol{x})}{\partial \theta}\right] - \frac{\partial E_\theta(\boldsymbol{x})}{\partial \theta}$$

- It is hard to compute the expectation because we cannot easily draw samples from $p_\theta(\boldsymbol{x})$. So we must resort to MCMC sampling to estimate the expectation.
- Recent successes on training large-scale EBM are based on Stochastic Gradient Langevin Dynamics (SGLD) sampler [2], which draws a sample through iterative refinement:

$$\boldsymbol{x}_0 \sim \mathcal{U}(\boldsymbol{x}), \qquad \boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{\alpha}{2}\frac{\partial E_\theta(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \alpha)$$

[2] M Welling et al., Bayesian learning via stochastic gradient langevin dynamics. ICML, 2011.

What your classifier is hiding (I)

- Standard softmax architectures model conditional probability by predicting logits of likelihood and passing though a softmax layer which normalizes the likelihood.

$$p(y|\boldsymbol{x}) = \frac{\exp(f_\theta(\boldsymbol{x})[y])}{\sum_y \exp(f_\theta(\boldsymbol{x})[y])}$$

- $f_\theta(\boldsymbol{x})[y]$ indicates the logit value of the input $\boldsymbol{x}$ that corresponds to a class y.

What your classifier is hiding (II)

- Our goal is to model the joint probability $p(\boldsymbol{x}, y)$ in the form of an EBM.

$$p_\theta(\boldsymbol{x}, y) = \frac{\exp\big(-E_\theta(\boldsymbol{x}, y)\big)}{Z_\theta} \quad \text{where} \int_{\boldsymbol{x},y} \exp\big(-E_\theta(\boldsymbol{x}, y)\big)$$

- First we factorize the joint probability and replace $p(\boldsymbol{x})$ by marginalizing:

$$p(\boldsymbol{x}, y) = p(\boldsymbol{x})p(y|\boldsymbol{x}) = \left(\sum_y p(\boldsymbol{x}, y)\right) p(y|\boldsymbol{x})$$

- If we choose $E_\theta(\boldsymbol{x}, y) = -f_\theta(\boldsymbol{x})[y]$ and plug into EBM, we get:

$$\frac{\exp(f_\theta(\boldsymbol{x})[y])}{Z_\theta} = \frac{\sum_y \exp(f_\theta(\boldsymbol{x})[y])}{Z_\theta} p(y|\boldsymbol{x})$$

- This leads us back to the softmax classifier:

$$p(y|\boldsymbol{x}) = \frac{\exp(f_\theta(\boldsymbol{x})[y])}{\sum_y \exp(f_\theta(\boldsymbol{x})[y])}$$

- In other words, we have revealed a generative model hidden within every standard discriminative model!

What your classifier is hiding (III)

- Notice that in standard softmax classifiers, shifting the logits does not affect the prediction. However with JEM, shifting the logits will affect $p_\theta(\boldsymbol{x}, y)$. This is because JEM utilizes the extra degree of freedom hidden in softmax classifiers to model $p_\theta(\boldsymbol{x})$.

$$\text{softmax}(\boldsymbol{h}) = \frac{\exp(h_i)}{\sum_y \exp(h_i)} = \frac{\exp(h_i + \alpha)}{\sum_y \exp(h_i + \alpha)}$$

- Overview of the results:

$$p_\theta(\boldsymbol{x}, y) = \frac{\exp(f_\theta(\boldsymbol{x})[y])}{Z_\theta} \qquad \log p_\theta(\boldsymbol{x}, y) = f_\theta(\boldsymbol{x})[y] - \log Z_\theta$$

$$p_\theta(\boldsymbol{x}) = \frac{\sum_y \exp(f_\theta(\boldsymbol{x})[y])}{Z_\theta} \qquad \log p_\theta(\boldsymbol{x}) = \text{LogSumExp}_y\, f_\theta(\boldsymbol{x})[y] - \log Z_\theta$$

$$p_\theta(y|\boldsymbol{x}) = \frac{\exp(f_\theta(\boldsymbol{x})[y])}{\sum_y \exp(f_\theta(\boldsymbol{x})[y])}$$

- Where,

$$E_\theta(\boldsymbol{x}, y) = -f_\theta(\boldsymbol{x})[y] \qquad E_\theta(\boldsymbol{x}) = -\text{LogSumExp}_y f_\theta(\boldsymbol{x})[y] \qquad Z_\theta = \int_{\boldsymbol{x}, y} \exp(f_\theta(\boldsymbol{x})[y])$$

Optimization (I)

- We optimize $p_\theta(y|\boldsymbol{x})$ using cross-entropy, and optimize $\log p_\theta(\boldsymbol{x})$ using SGLD to estimate the expectation.

$$\mathcal{L}_{classifier}(\boldsymbol{x}, y) = \text{CrossEntropy}(f_\theta(\boldsymbol{x}), y)$$

$$\mathcal{L}_{generator}(\boldsymbol{x}) = -\log p_\theta(\boldsymbol{x})$$

$$objective = \mathcal{L}_{classifier}(\boldsymbol{x}, y) + \mathcal{L}_{generator}(\boldsymbol{x})$$

- Following [3] we use persistent contrastive divergence [4] to estimate the expectation since it gives an order of magnitude savings in computation compared to seeding new chains at each iteration as in Nijakamp et al., this cost comes at the cost of decreased training stability. The trade-offs are discussed in appendix H.2.

[3] Y Du et al., Implicit generation and generalization in energy-based models. arXiv, 2019.
[4] T Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. ICML, 2008

Optimization (II)

Algorithm 1 JEM training: Given network f_θ, SGLD step-size α, SGLD noise σ, replay buffer B, SGLD steps η, reinitialization frequency ρ

while not converged **do**
  Sample $\mathbf{x}$ and y from dataset
  $L_{\text{clf}}(\theta) = \text{xent}(f_\theta(\mathbf{x}), y)$
  Sample $\widehat{\mathbf{x}}_0 \sim B$ with probability $1 - \rho$, else $\widehat{\mathbf{x}}_0 \sim \mathcal{U}(-1, 1)$ ▷ Initialize SGLD
  for $t \in [1, 2, \ldots, \eta]$ **do** ▷ SGLD
    $\widehat{\mathbf{x}}_t = \widehat{\mathbf{x}}_{t-1} + \alpha \cdot \frac{\partial \text{LogSumExp}_{y'}(f_\theta(\widehat{\mathbf{x}}_{t-1})[y'])}{\partial \widehat{\mathbf{x}}_{t-1}} + \sigma \cdot \mathcal{N}(0, I)$
  end for
  $L_{\text{gen}}(\theta) = \text{LogSumExp}_{y'}(f(\mathbf{x})[y']) - \text{LogSumExp}_{y'}(f(\widehat{\mathbf{x}}_t)[y'])$ ▷ Surrogate for Eq 2
  $L(\theta) = L_{\text{clf}}(\theta) + L_{\text{gen}}(\theta)$
  Obtain gradients $\frac{\partial L(\theta)}{\partial \theta}$ for training
  Add $\widehat{\mathbf{x}}_t$ to B
end while

Applications

- Achieved performance rivaling to SOTA in both discriminative and generative modeling.
- Added benefits to discriminative tasks: uncertainty qualification, OOD detection, robustness to adversarial examples.
- All architectures are based on WideResNet with BN removed to ensure deterministic output. This slightly increases error of a WRN-28-10 from 4.2% to 6.4% on CIFAR-10 and from 2.3% to 3.4% on SVHN.

Hybrid Modeling (I)

- A classifier architecture can be trained as an EBM to show competitive performance as both a classifier and generative model.
- JEM can achieve near-SOTA in both classification and generation, while outperforming other hybrid models. (CIFAR-10)

Class	Model	Accuracy% ↑	IS↑	FID↓
Hybrid	Residual Flow	70.3	3.6	46.4
	Glow	67.6	3.92	48.9
	IGEBM	49.1	8.3	**37.9**
	JEM $p(\mathbf{x}\|y)$ factored	30.1	6.36	61.8
	JEM (Ours)	**92.9**	**8.76**	38.4
Disc.	Wide-Resnet	95.8	N/A	N/A
Gen.	SNGAN	N/A	8.59	25.5
	NCSN	N/A	8.91	25.32

Table 1: CIFAR10 Hybrid modeling Results. Residual Flow (Chen et al., 2019), Glow (Kingma & Dhariwal, 2018), IGEBM (Du & Mordatch, 2019), SNGAN (Miyato et al., 2018), NCSN (Song & Ermon, 2019)

Hybrid Modeling (II)

- Class-conditional samples generated from JEM:

- The paper includes further interesting qualitative analyses in the appendix.

Figure 7: Class-conditional Samples. Left to right: CIFAR10, SVHN.

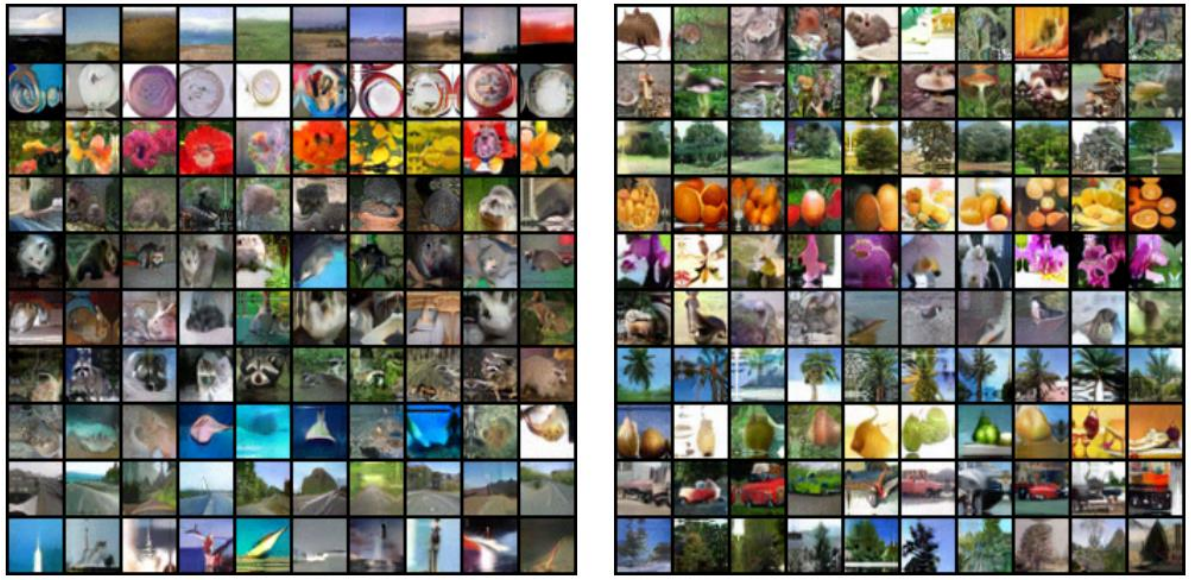


Figure 8: CIFAR100 Class-conditional Samples.

Calibration

- A classifier is calibrated if its confidence $\max_y p(y|\boldsymbol{x})$ is consistent with its error rate.
- Calibration is important in mission-critical applications, for example, where the confidence could be used to decide whether to defer the task to a human. Therefore, a well-calibrated but less accurate classifier can be considerably more useful than the other.
- Recent classifiers have grown more accurate but considerably less calibrated. [5] On the contrary, JEM notably improves classification while retaining high accuracy.

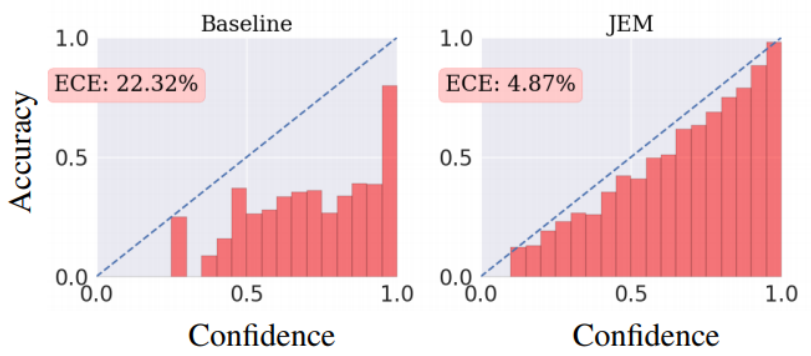


Figure 4: CIFAR100 calbration results. ECE = Expected Calibration Error (Guo et al., 2017), see Appendix E.1.

Model	Accuracy %	ECE %
Baseline	74.2	22.32
JEM	72.2	4.87

[5] C Guo et al., On calibration of modern neural networks. In ICML, 2017.

Out-of-distribution Detection (I)

- In OOD detection, the goal to estimate an in-distributioness score $s_\theta(\boldsymbol{x}) \in \mathbb{R}$. We apply JEM to: input density estimation, predictive distribution estimation, approximate mass.
- Input density: this approach fits a density model $p(\boldsymbol{x})$ on the data, and consider examples with low likelihood to be OOD. This approach is currently not competitive on high-dimensional data.
- [6] showed that tractable deep generative models [7, 8] can assign higher densities to OOD examples than in-distribution examples. Conversely, Du & Mordatch have shown that the likelihoods from EBMs can be reliably used as a predictor for OOD inputs. JEM consistently assigns higher likelihood to ID samples than OOD samples. One possible explanation for JEM's improvement over IGEBM is its ability to incorporate labeled information during training.

[6] E Nalisnick et al., Do deep generative models know what they don't know? arXiv, 2018.
[7] D P Kingma et al., Glow: Generative flow with invertible 1x1 convolutions. In NIPS, 2018.
[8] T Salimans et al., Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv, 2017.

Out-of-distribution Detection (II)

- Predictive Distribution: $s_\theta(\boldsymbol{x}) = \max_y p_\theta(y|\boldsymbol{x})$
- OOD detection by utilizing a classifier's predictive distribution have shown many success in [9, 10, 11, 12]. Since JEM is a competitive classifier, we find it performs on par or beyond the performance of a strong baseline classifier.

[9] Y Gal et al., Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. ICML, 2016.
[10] K Wang et al., Adversarial distillation of bayesian neural network posteriors. ICML, 2018.
[11] S Liang et al., Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv 2017.
[12] D Hendrycks et al., A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv 2016.

Out-of-distribution Detection (III)

- Approximate Mass:

$$s_\theta(\boldsymbol{x}) = -\left\| \frac{\partial \log p_\theta(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_2$$

- It is possible that a sample has high likelihood $p_\theta(\boldsymbol{x})$ under a distribution yet be nearly impossible to be sampled. Real samples lie in typical set, which is the area of high probability mass.
- For a high-likelihood sample outside of the typical set, we expect the density to change rapidly around it, thus the norm of the gradient of the log-density will be large compared to examples in the typical set. We propose an alternative OOD score based on this property:
- For EBMs (JEM and IGEBM), we find this predictor greatly outperforms our own and other generative model's likelihoods. For tractable likelihood methods we find this predictor is anti-correlated with the model's likelihood and neither is reliable for OOD detection.

Out-of-distribution Detection (IV)

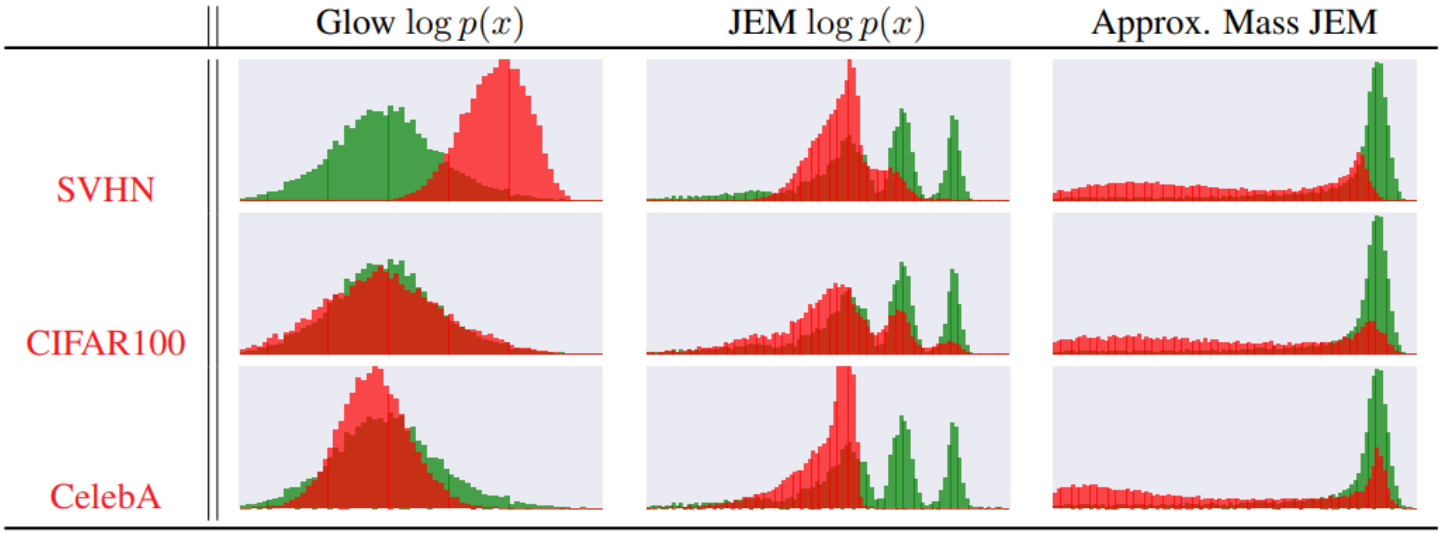


Table 2: Histograms for OOD detection. All models trained on CIFAR10. Green corresponds to the score on (in-distribution) CIFAR10, and red corresponds to the score on the OOD dataset.

In-distribution	Out-of-distribution
CIFAR10	SVHN
	CIFAR100
	CelebA

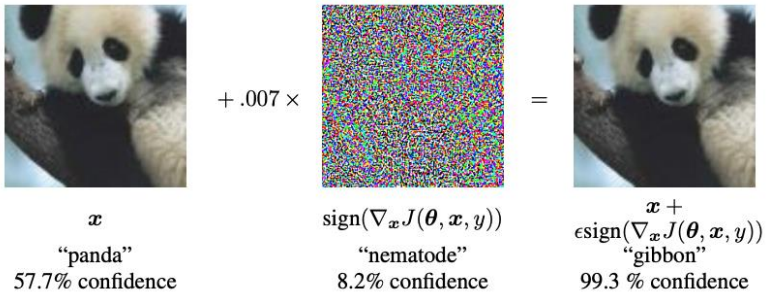
Out-of-distribution Detection (V)

$s_\theta(\mathbf{x})$	Model	SVHN	CIFAR10 Interp	CIFAR100	CelebA
$\log p(\mathbf{x})$	Unconditional Glow	.05	.51	.55	.57
	Class-Conditional Glow	.07	.45	.51	.53
	IGEBM	.63	**.70**	.50	.70
	JEM (Ours)	**.67**	.65	**.67**	**.75**
$\max_y p(y\|\mathbf{x})$	Wide-ResNet	**.93**	**.77**	.85	.62
	Class-Conditional Glow	.64	.61	.65	.54
	IGEBM	.43	.69	.54	.69
	JEM (Ours)	.89	.75	**.87**	**.79**
$\left\|\left\|\frac{\partial \log p(\mathbf{x})}{\partial \mathbf{x}}\right\|\right\|$	Unconditional Glow	**.95**	.27	.46	.29
	Class-Conditional Glow	.47	.01	.52	.59
	IGEBM	.84	.65	.55	.66
	JEM (Ours)	.83	**.78**	**.82**	**.79**

Table 3: OOD Detection Results. Models trained on CIFAR10. Values are AUROC.

Robustness against adversarial attacks

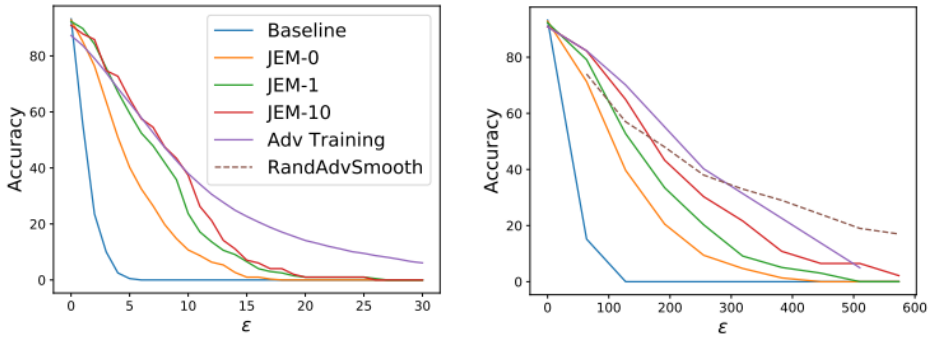
- Similarities of EBM training to Adversarial training:
  - We use a gradient-based optimization to generate examples which activate a high-level network activation. Then optimize the network to minimize the generated example's effect on the activation.
  - Given these connections one may wonder if JEM would be more robust to adversarial examples than a standard model. This behavior has been demonstrated in prior work on EBMs.
- Adversarial Attacks [13]
  - Adversarial threats are modeled as perturbed inputs: $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \delta$ where $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_p < \varepsilon$
  - Closeness with regards to L_p distance does not imply that adversarial examples reside within areas of high density according to the model distribution.



[13] I Goodfellow et al., Explaining and harnessing adversarial examples. arXiv 2014

Robustness against adversarial attacks

- Perturbation-based adversarial examples
  - We propose to run a few iterations of the sampling procedure seeded at the given input. This should be able to transform low-probability inputs to a nearby point of high probability, “undoing” any adversarial attack and enabling the model to classify robustly.
  - We test JEM with 0, 1, 10 steps of sampling seeded at the input.
  - Our model is considerably more robust than a baseline with standard classifier training. JEM with 0 steps refinement is noticeably more robust than the baseline.
  - Adv and RandAdvSmooth [14] is trained to be robust to the specific norm. However, we attack the same JEM model with both norms and observe competitive robustness in both cases.



(a) L_∞ Robustness (b) L_2 Robustness

Figure 5: Adversarial Robustness Results with PGD attacks. JEM adds considerable robustness.

[14] H Salman et al., Provably robust deep learning via adversarially trained smoothed classifiers. arXiv 2019.

Robustness against adversarial attacks

- Distal adversarial examples [15]
  - Non-robust models have tendency to classify non-sensical inputs with high confidence.
  - Starting from noise we generate images to maximize $p_\theta(y = \text{car}|\boldsymbol{x})$.
  - The baseline confidently classifies unstructured noise images.
  - The L2-adversarially trained ResNet with $\epsilon = 0.5$ confidently classifies somewhat structured, but unrealistic images.
  - JEM does not confidently classify nonsensical images, instead, car attributes and natural image properties visibly emerge.

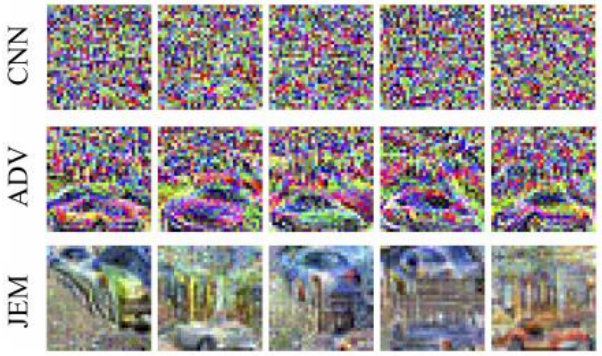


Figure 6: **Distal Adversarials.** Confidently classified images generated from noise, such that: $p(y = \text{“car”}|\mathbf{x}) > .9$.

[15] L Schott et al., Towards the first adversarially robust neural network model on mnist. arXiv 2018.

Limitations

- EBMs can be very challenging to work with:
  - Difficulty in evaluation: Since normalized likelihoods cannot be computed, it is hard to even verify that learning is taking place at all. In some domains such as images, samples could be drawn to assess learning, however this is far from a generalizable strategy.
  - Difficulty in training: The gradient estimators we use to train JEM are quite unstable and are prone to diverging if the sampling and optimization parameters are not tuned correctly. The models demonstrated in this work regularly diverged throughout training, requiring them to be restarted with lower learning rates or with increased regularization.
- While this may seem prohibitive, we believe that the results presented in this work are sufficient to motivate the community to find solutions to these issues as any improvement in the training of energy based models will further improve the results we have presented in this work.

Conclusion

- We have presented JEM, a novel reinterpretation of standard classifier architectures which retains the strong performance of SOTA discriminative models while adding the benefits of generative modeling approaches.
- Our work is based upon recent works on scaling techniques for training EBMs to high-dimensional data.
- We have demonstrated the utility of incorporating this type of training into discriminative models.

References

- Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations* (ICLR), 2020