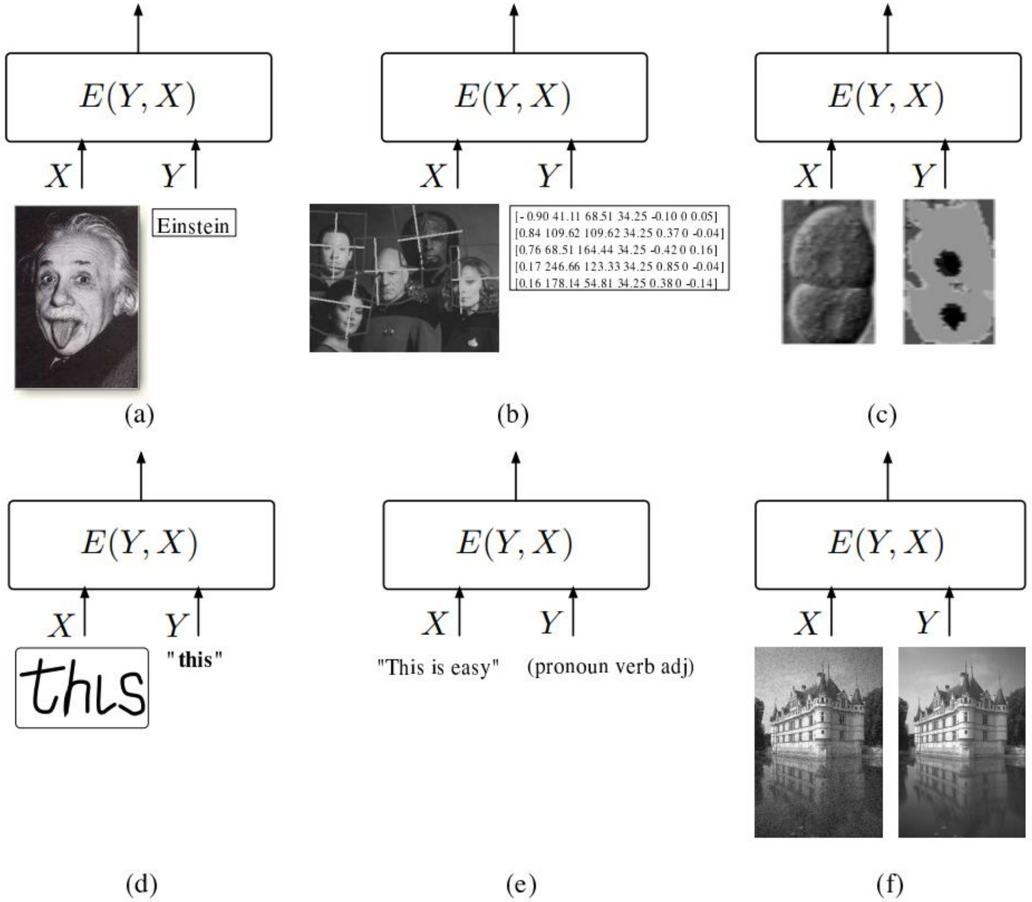
Energy in Secret

for both generative and discriminative modeling

Zhiyuan Tang

2020.3.23

Recap: Energy Based Model (EBM)



$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X).$$

Easy

A tutorial on energy-based learning

Recap: Energy Based Model (EBM)

1. *Prediction, classification, and decision-making*: "Which value of Y is most compatible with this X?' This situation occurs when the model is used to make hard decisions or to produce an action. For example, if the model is used to drive a robot and avoid obstacles, it must produce a single best decision such as "steer left", "steer right", or "go straight".

2. *Ranking*: "Is Y_1 or Y_2 more compatible with this X?" This is a more complex task than classification because the system must be trained to produce a complete ranking of all the answers, instead of merely producing the best one. This situation occurs in many data mining applications where the model is used to select multiple samples that best satisfy a given criterion.

3. *Detection*: "Is this value of Y compatible with X?" Typically, detection tasks, such as detecting faces in images, are performed by comparing the energy of a *face* label with a threshold. Since the threshold is generally unknown when the system is built, the system must be trained to produce energy values that increase as the image looks less like a face.

4. *Conditional density estimation*: "What is the conditional probability distribution over $\mathcal{Y}$ given X?" This case occurs when the output of the system is not used directly to produce actions, but is given to a human decision maker or is fed to the input of another, separately built system.

$$P(Y|X) = \frac{e^{-\beta E(Y,X)}}{\int_{y\in\mathcal{Y}} e^{-\beta E(y,X)}},$$

Hard

A tutorial on energy-based learning

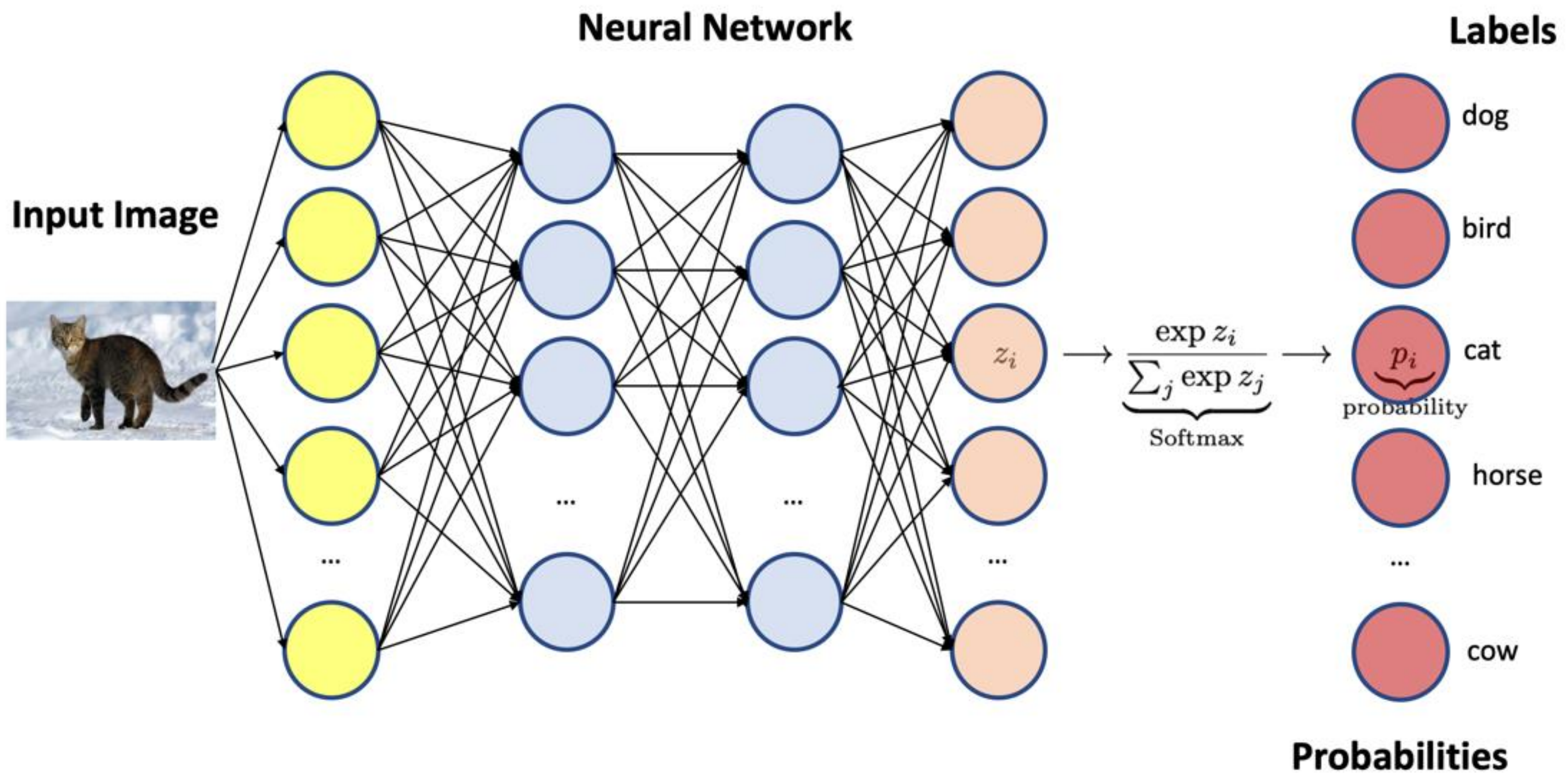
Reference 1

- Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One, ICLR 2020

Purpose

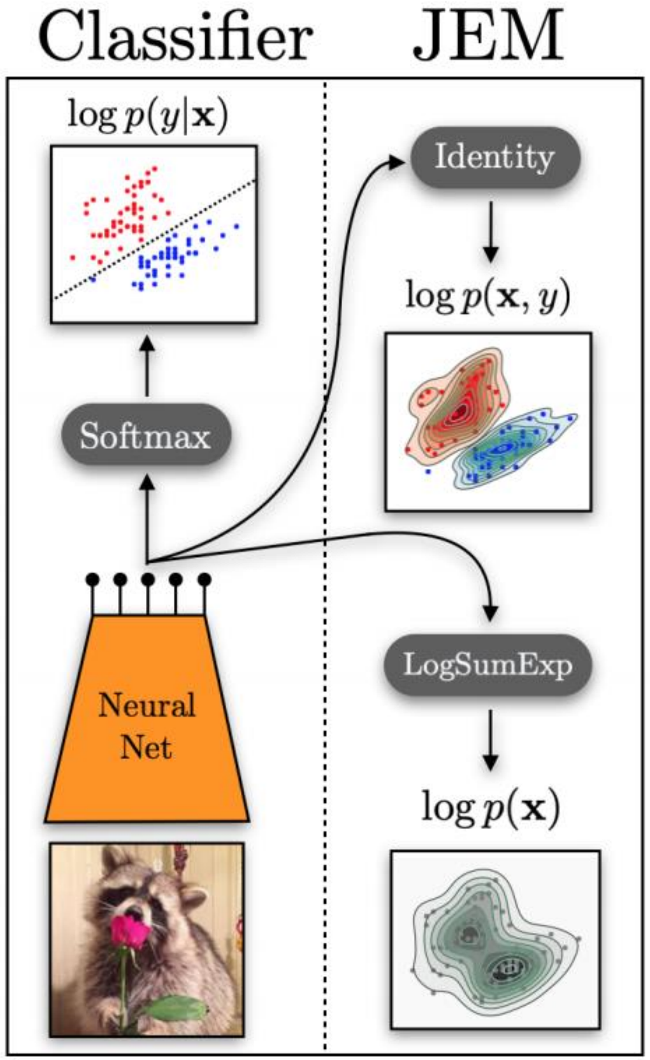
- Standard discriminative classifier += Energy based model
- Compute p(y|x), p(x), p(x|y) with the same model

Softmax in a standard classifier



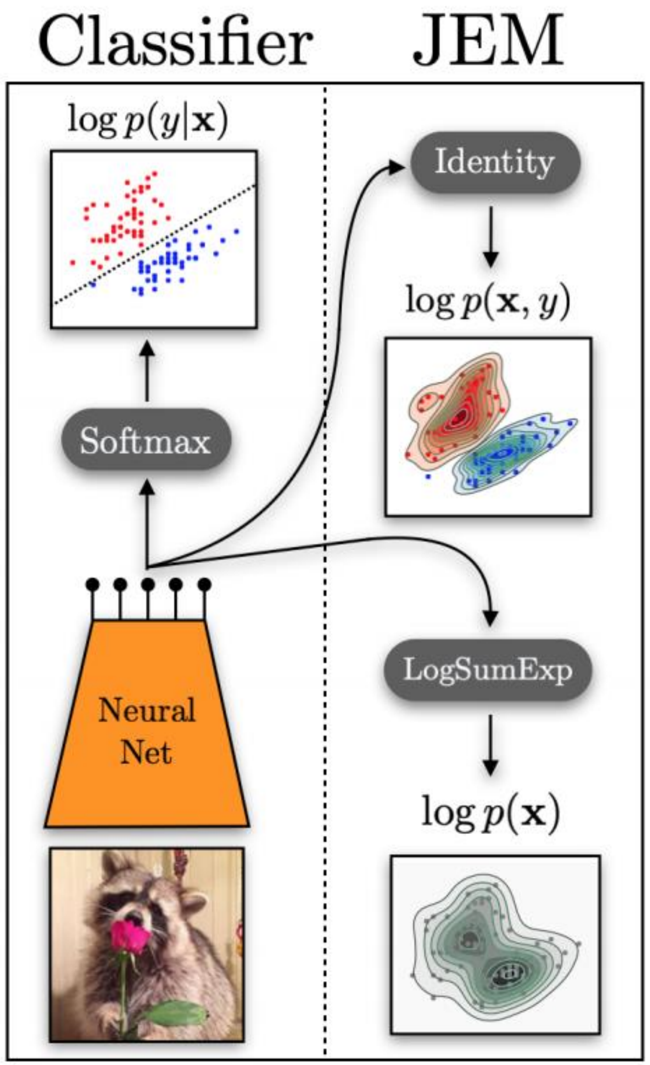
$$p_\theta(y \mid \mathbf{x}) = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{\sum_{y'} \exp\left(f_\theta(\mathbf{x})[y']\right)},$$

Define EMB with logits



$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)} ,$$

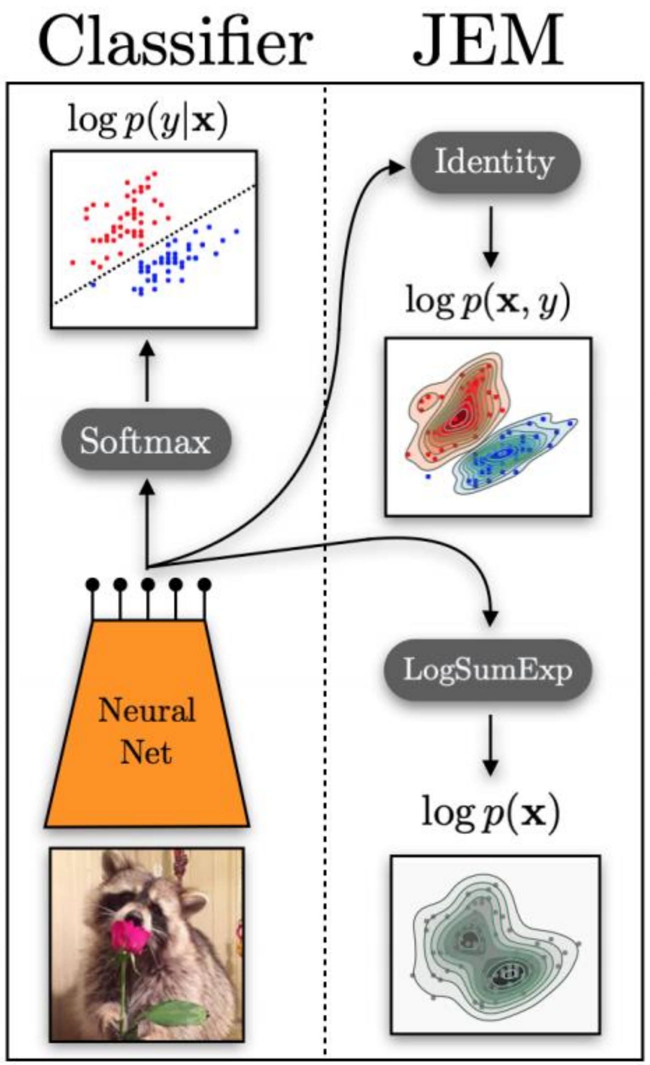
Define EMB with logits



$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)},$$

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y)$$

Define EMB with logits



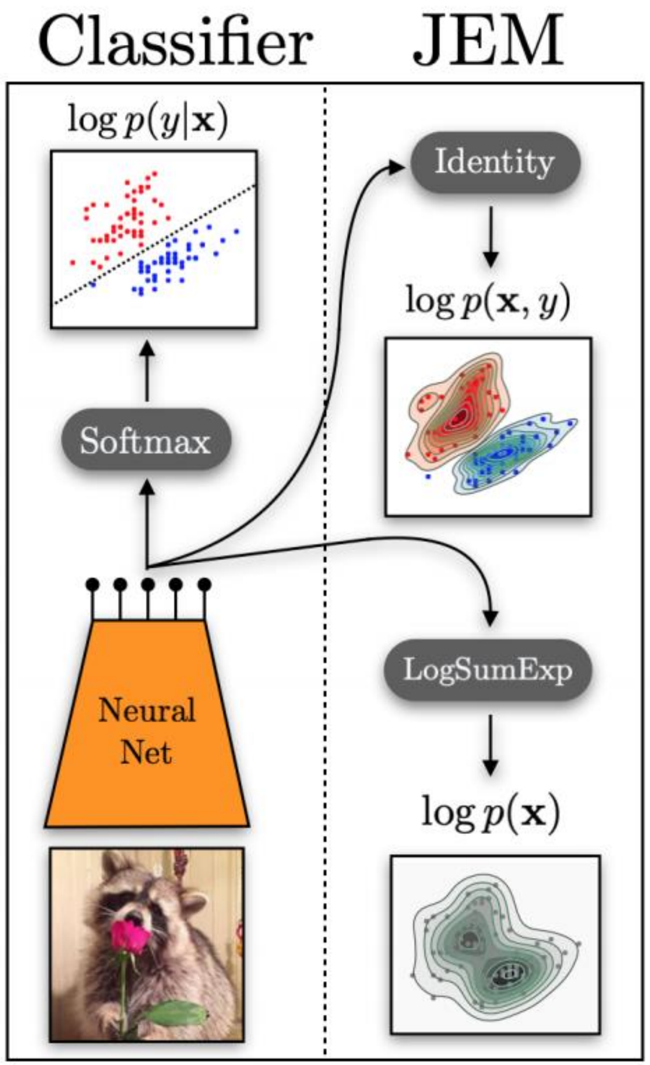
$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)},$$

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y)$$

$$p_\theta(\mathbf{x}, y) = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{Z(\theta)},$$

Definition

Define EMB with logits



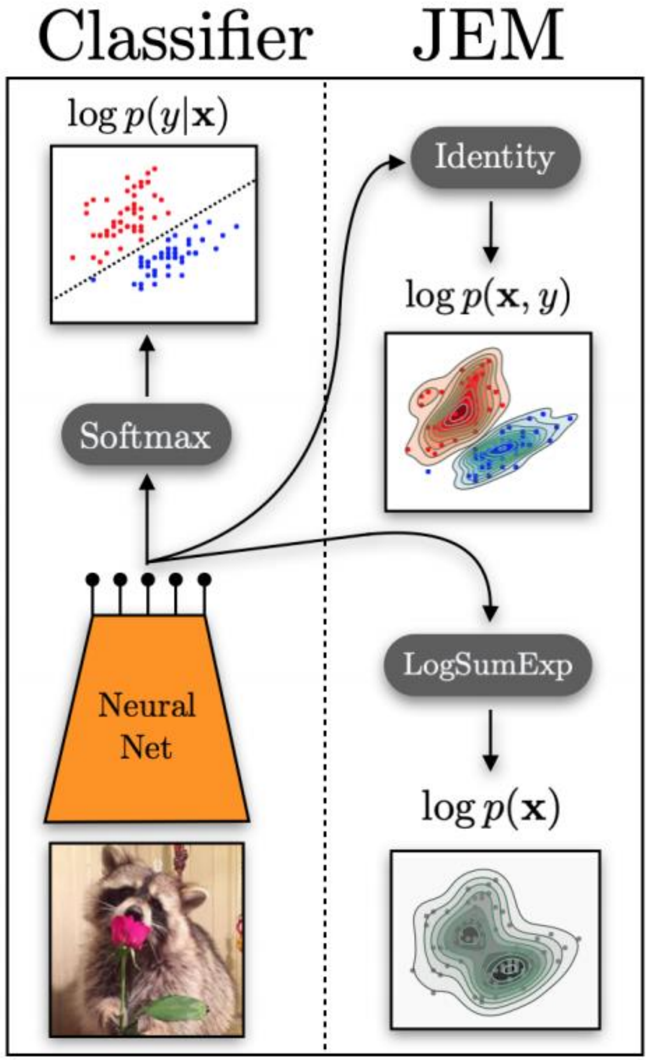
$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)},$$

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y)$$

$$p_\theta(\mathbf{x}, y) = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{Z(\theta)},$$ Definition

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \frac{\sum_y \exp\left(f_\theta(\mathbf{x})[y]\right)}{Z(\theta)}.$$

Define EMB with logits



$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)},$$

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y)$$

$$p_\theta(\mathbf{x}, y) = \frac{\exp(f_\theta(\mathbf{x})[y])}{Z(\theta)},$$

Definition

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \frac{\sum_y \exp(f_\theta(\mathbf{x})[y])}{Z(\theta)}.$$

$$E_\theta(\mathbf{x}) = -\text{LogSumExp}_y(f_\theta(\mathbf{x})[y]) = -\log \sum_y \exp(f_\theta(\mathbf{x})[y]).$$

Optimization

Generative + Discriminative

$$\log p_\theta(\mathbf{x}, y) = \log p_\theta(\mathbf{x}) + \log p_\theta(y|\mathbf{x}).$$

Cross-entropy

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

$$\mathbf{x}_0 \sim p_0(\mathbf{x}), \qquad \mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\alpha}{2}\frac{\partial E_\theta(\mathbf{x}_i)}{\partial \mathbf{x}_i} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \alpha)$$

Stochastic Gradient Langevin Dynamics (SGLD)

All together

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

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

Hybrid modeling

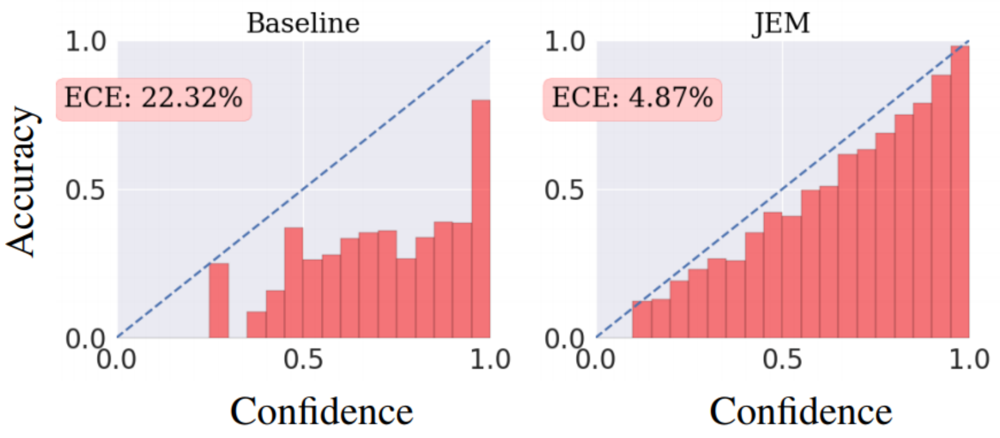


Figure 4: CIFAR100 calbration results. ECE = Expected Calibration Error (Guo et al., 2017), see Appendix E.1.

Calibration

Applications – Out-of-domain detection

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

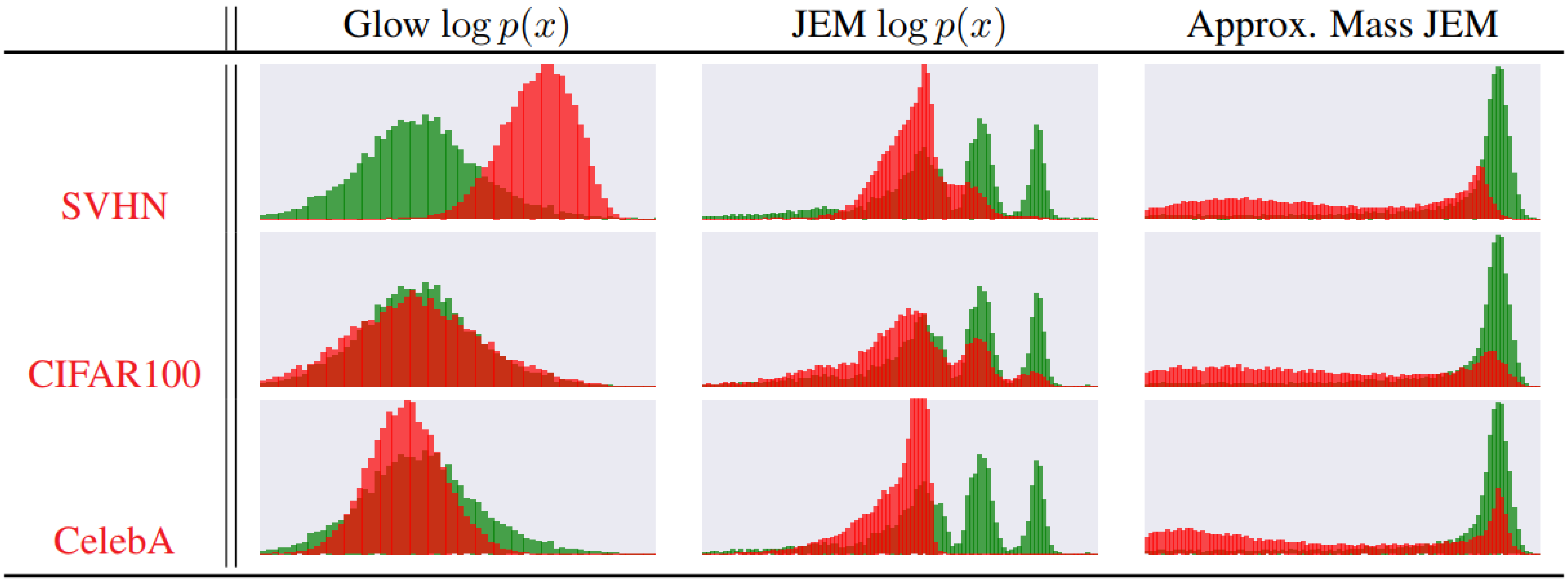


Table 2: Histograms for OOD detection. All models trained on CIFAR10. Green corresponds to the score on (in-distribution) CIFAR10, and red corresponds to the score on the OOD dataset.

Applications – Out-of-domain detection

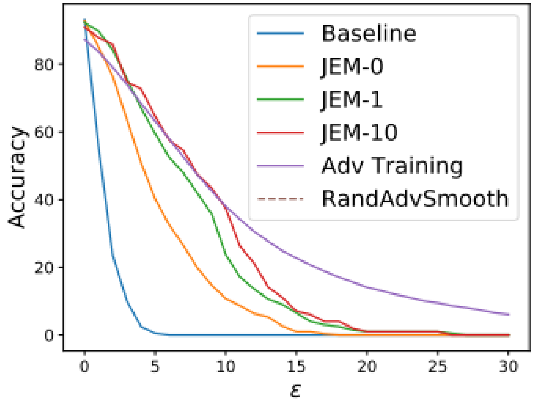
$s_\theta(\mathbf{x})$	Model	SVHN	CIFAR10 Interp	CIFAR100	CelebA
$\log p(\mathbf{x})$	Unconditional Glow	.05	.51	.55	.57
	Class-Conditional Glow	.07	.45	.51	.53
	IGEBM	.63	**.70**	.50	.70
	JEM (Ours)	**.67**	.65	**.67**	**.75**
$\max_y p(y\|\mathbf{x})$	Wide-ResNet	**.93**	**.77**	.85	.62
	Class-Conditional 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.

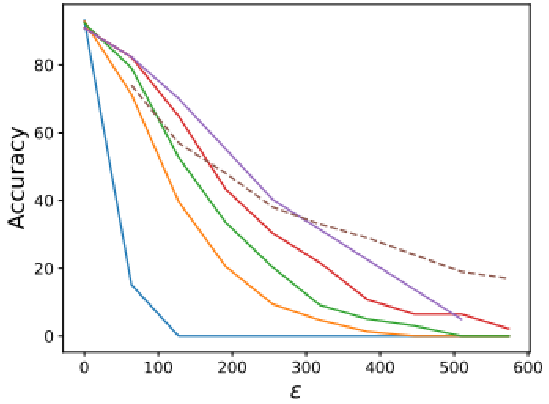
Robustness against adversarial examples

$$\tilde{\mathbf{x}} = \mathbf{x} + \delta,$$

$$||\tilde{\mathbf{x}} - \mathbf{x}||_p < \epsilon$$



(a) L_∞ Robustness



(b) L_2 Robustness

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

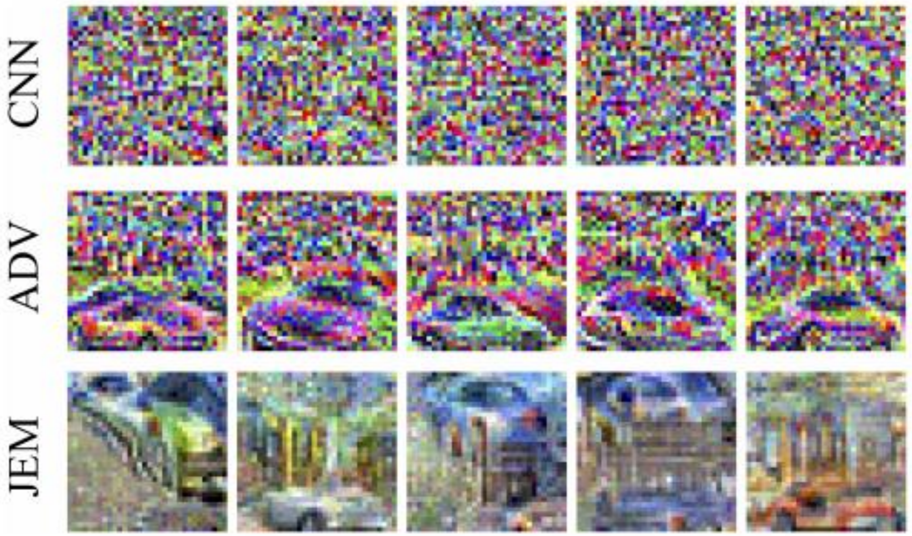


Figure 6: **Distal Adversarials.** Confidently classified images generated from noise, such that: $p(y = \text{“car”}|\mathbf{x}) > .9$.

Unstability

- The models used to generate the results in this work regularly diverged throughout training, requiring them to be **restarted** with lower learning rates or with increased regularization.

Reference 2

- Your GAN is Secretly an Energy-based Model and You Should use Discriminator Driven Latent Sampling, arXiv.

To be continued …