# Joint Energy-Based Models for Semi-Supervised Classification

**Stephen Zhao** [1] **Jörn-Henrik Jacobsen** [1] **Will Grathwohl** [1]

## Abstract

Replacing discriminative classifiers which model $p(y|\mathbf{x})$ with energy-based models of the joint distribution over data and labels $p(\mathbf{x}, y)$ has recently been shown to produce models with better calibrated uncertainty, robustness, and out-of-distribution detection abilities while also retaining the strong predictive performance of discriminative baselines. We further explore the capabilities of energy-based classifiers for semi-supervised learning. We find our approach works well across domains and in settings where other recently proposed semi-supervised learning methods do not perform well.

## 1. Introduction

Semi-supervised learning (SSL) is a core problem in machine learning. In most real-world settings, unlabeled data can be obtained for small fraction of the cost of labeled data. Unfortunately, unlabeled examples are not straightforward to leverage in discriminative learning, leading most compelling applications of machine learning today to be the result of large-scale supervised learning.

Despite this, considerable progress has recently been made in SSL. Most of these approaches rely on data-augmentation strategies heavily tuned for image data (Berthelot et al., 2019; Sohn et al., 2020) leading to impressive performance in this domain but providing limited application outside of it. A standout approach is Virtual Adversarial Training (VAT) (Miyato et al., 2018) which does not rely on data-augmentation and instead enforces norm-bounded perturbation insensitivity on unlabeled data. While this requires far less domain knowledge, it too may be overly tuned to the image domain in which the $l_2$ and $l_{\text{inf}}$ norm are reasonable choices but this may be not true in other domains.

A more domain-agnostic approach to SSL is based on generative models. We train a model of $p(\mathbf{x}, y)$. When we observe labels $y$, we maximize $p(\mathbf{x}, y)$, and when the label is unobserved we marginalize it out and maximize $p(\mathbf{x})$. Unfortunately, when used for classification, conditional generative models tend to perform much worse than their discriminative counterparts (Fetaya et al., 2019).

Recently, energy-based models (EBMs) (Du & Mordatch, 2019; Xie et al., 2016; Nijkamp et al., 2019) have become a promising approach for generative modeling. Grathwohl et al. (2019) have demonstrated that unlike other classes of generative models, EBMs can be used to build conditional generative models which perform on par with the state-of-the-art discriminative models at classification while rivaling GANs at generative modeling.

In this work we extend the method of Grathwohl et al. (2019), JEM, and apply it to SSL. We find that JEM classifiers provide noticeable benefit to SSL, perform comparably to VAT in the image domain, and outperform VAT on non-image data, such as arbitrary tabular data.

## 2. Related Work

### 2.1. Virtual Adversarial Training

Virtual Adversarial Training (VAT) is a recently proposed method for SSL which stands apart from other successful methods in that it does not require pre-specified data-augmentation. VAT enforces classifiers to be invariant within an $\epsilon$-ball of an unlabeled input $x$ with respect to an $\ell_p$-norm. This is achieved by finding the example $x'$ within the ball which maximally changes the model's output and then enforcing the model's predictions to be the same at both points. For a model which outputs a distribution over $k$ classes as a function $f(x)$, the training objective for unlabeled data is:

$$x' = x + \underset{||r||_p < \epsilon}{\text{argmax}} \left[ D_{KL}(f(x)||f(x+r)) \right]$$
$$\mathcal{L}(x) = D_{KL}(f(x)||f(x')). \quad (1)$$

The method's hyperparameters include the chosen norm, perturbation size $\epsilon$, and the weight of the unlabeled objective compared to the supervised objective. This method has been quite successful, but later we discuss settings where these choices may be too restrictive.

[1]University of Toronto and Vector Institute. Correspondence to: Stephen Zhao <stephen.zhao@mail.utoronto.ca>, Will Grathwohl <wgrathwohl@cs.toronto.edu>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning. 

### 2.2. Energy-Based Models and JEM

As observed in Grathwohl et al. (2019), a typical classifier using a softmax activation function can be interpreted as a generative energy-based model. Energy-based models (LeCun et al., 2006) express any probability density $p(\mathbf{x})$ for $\mathbf{x} \in R^D$ in terms of:

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)} \tag{2}$$

where $E_\theta(\mathbf{x}) : R^D \to R$ is known as the energy function, and $Z(\theta) = \int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x}))$. A standard classifier models

$$p_\theta(y|\mathbf{x}) = \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{y'} \exp(f_\theta(\mathbf{x})[y'])} \tag{3}$$

where $f_\theta : R^D \to R^K$ and $K$ is the number of classes. The same parametric function $f_\theta$ can be reinterpreted to define a joint distribution $p_\theta(\mathbf{x}, y)$ as follows:

$$p_\theta(\mathbf{x}, y) = \frac{\exp(f_\theta(\mathbf{x})[y])}{Z(\theta)} \tag{4}$$

We can obtain $p_\theta(\mathbf{x})$ by marginalizing out $y$, resulting in:

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \frac{\sum_y \exp(f_\theta(\mathbf{x})[y])}{Z(\theta)} \tag{5}$$

which is an energy based model, where $E_\theta(\mathbf{x}) = -\log(\sum_y \exp(f_\theta(\mathbf{x})[y]))$.

A Joint Energy-based Model (JEM) that works jointly as a discriminative and generative model can be trained using the above formulation, by factoring the joint log likelihood:

$$\log p_\theta(\mathbf{x}, y) = \log p_\theta(\mathbf{x}) + \log p_\theta(y|\mathbf{x}) \tag{6}$$

where $p_\theta(y|\mathbf{x})$ is optimized in the same way as a typical classifier and $p_\theta(\mathbf{x})$ is optimized as an energy based model using Persistent Contrastive Divergence (PCD) (Tieleman, 2008) with samples drawn using Stochastic Gradient Langevin Dynamics (Welling & Teh, 2011). JEMs were shown to combine the advantages of discriminative and generative models, achieving near state-of-the-art performance in classification and generative tasks simultaneously, while achieving better calibration, out-of-distribution detection, and adversarial robustness than a standard classifier.

## 3. Proposed Approach

Motivated by the calibration and robustness of energy-based classifiers, we now investigate whether these benefits translate into improved performance in SSL, where we have limited labeled data. To adapt the JEM training procedure to this setting, labeled data points are trained using the factorization in Eq. 6 above, optimizing both $\log p_\theta(\mathbf{x})$ and the standard classification term $\log p_\theta(y|\mathbf{x})$, whereas for unlabeled data points, we optimize just $\log p_\theta(\mathbf{x})$. In this way, unlabeled data also helps us better model the joint distribution. The generative modeling term can be thought of intuitively as a form of regularization or consistency enforcer, dependent on the shape of the data distribution. This should help the model avoid overfitting on the limited training data and generalize better to unlabeled and unseen data.

### 3.1. Beyond Pre-Specified Invariance

Most recent SSL approaches work by enforcing the classifier to be invariant to a pre-specified set of transformations. Berthelot et al. (2019) and Sohn et al. (2020) use traditional data-augmentation for images such as random shifts and color changes. Miyato et al. (2018) enforces their model to be invariant to norm-bounded perturbations, requiring specification of a suitable $\ell_p$-norm. We believe JEM provides similar benefits while making far fewer assumptions. In less studied domains, powerful data-augmentation strategies are not known so these approaches cannot be applied. Similarly, in many domains there may not exist a single norm and perturbation size where a decent classifier can be learned (see Figure 1 for an illustrative toy example). In fact, it can be proven that finding an optimal norm and perturbation size even on relatively well-understood data like natural images is impossible (Tramèr et al., 2020). In consequence, augmentation and adversarial-training based approaches always require many heuristic decisions, which are mainly limited to domains where humans have a strong intuition for the structure of the data.

By tying the classifier to the log-density of the unconditional data distribution, we enforce the classifier's decisions to be invariant in areas where the data density is relatively constant. This forces the classifier's decision boundary to lie in an area where the data density is low. Since we learn this density alongside our classifier on unlabeled data, this pushes the decision boundary to not cut though the modes of the data, providing strong semi-supervised classification results. This behavior is illustrated in Figure 1.

## 4. Training Details

As in (Grathwohl et al., 2019), we optimize $\log p_\theta(y|\mathbf{x})$ using the standard cross-entropy loss, and we optimize $\log p_\theta(\mathbf{x})$ using the well-known estimator:

$$\frac{\partial \log p_\theta(\mathbf{x})}{\partial \theta} = \mathbb{E}_{p_\theta(\mathbf{x}')}\left[\frac{\partial E_\theta(\mathbf{x}')}{\partial \theta}\right] - \frac{\partial E_\theta(\mathbf{x})}{\partial \theta} \tag{7}$$

where the expectation is approximated with a sampler based on Stochastic Gradient Langevin Dynamics (SGLD)

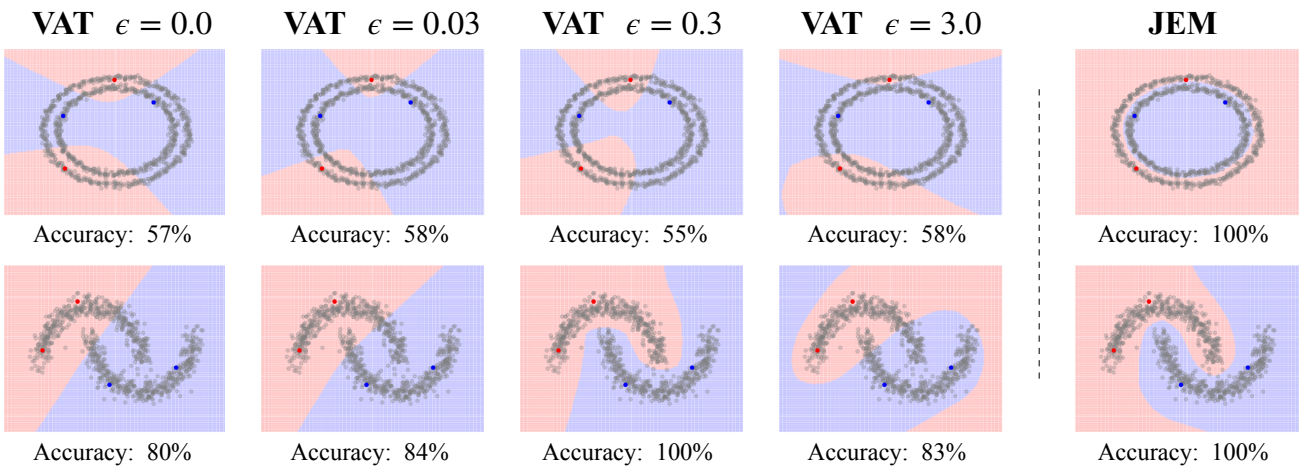


*Figure 1.* Comparison of VAT with various $\epsilon$ in $\ell_2$ norm and JEM on the concentric circle (top row) and two moons dataset (bottom row). Blue and red dots denote labeled data, grey dots denote unlabeled data, background red and blue denote learned decision boundaries. Note how JEM only places decision boundaries in low density regions. VAT is agnostic to the underlying data density and only concerned with learning a smooth map, whose smoothness is determined by hand-chosen $\epsilon$. For the two moons dataset we can find an optimal $\epsilon$ that gives 100% test accuracy. However, for the concentric circles dataset, $\ell_2$ distance is semantically meaningful, making it impossible to find a good choice of $\epsilon$, hence VAT fails. JEM achieves 100% accuracy on both datasets as it does not make any assumptions about semantic meaning of a certain norm-bounded perturbation.

(Welling & Teh, 2011) which generates samples following:

$$\mathbf{x}_0 \sim p_0(\mathbf{x}), \qquad \epsilon_i \sim \mathcal{N}(0, \beta^2)$$
$$\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\alpha}{2}\frac{\partial E_\theta(\mathbf{x}_i)}{\partial \mathbf{x}_i} + \epsilon_i. \tag{8}$$

For a proper Langevin diffusion we set $\alpha = \beta^2$. For high dimensional distributions this leads to prohibitively small step sizes $\alpha$ causing the sampler to be too slow to work with. In practice the sampler is tempered which equates to decoupling $\alpha$ and $\beta$. Typically $\beta$ is set to a sufficiently small value to allow samples to resemble data (0.01 is typical for images) and then $\alpha$ is tuned for stable training.

We use PCD (Tieleman, 2008), with a replay buffer and random restarts as in Du & Mordatch (2019); Grathwohl et al. (2019). In all experiments we use a buffer with 10,000 samples and random restart probability 0.05. At each training iteration the buffer samples are updated for 40 steps. Fewer steps could be used to achieve similar accuracy to our reported results but training was less stable.

## 5. Experiments

We demonstate the performance of semi-supervised JEM on a number of datasets and domains. We begin with a 2D toy example which demonstrates how and why JEM performs well at SSL and why it works in settings where VAT fails. Next we focus on two standard benchmark datasets for SSL; MNIST and SVHN. Finally, to demonstrate that our approach has promise outside of the image domain we provide results on tabular data from the UCI data repository.

We compare the performance of JEM against three baselines: a standard regularized classifier trained only on the labeled data, VAT (Miyato et al., 2018), and the semi-supervised variational auto-encoder (VAE) (Kingma et al., 2014). For the VAE, we focus on the best-performing stacked model (M1 + M2) which uses representations from a latent-feature discriminative model (M1) as embeddings for a generative semi-supervised model (M2). For all experiments, we keep network architectures and as many hyperparameters as we can constant. Code is available here: `https://github.com/Silent-Zebra/JEM`.

### 5.1. Visualizing the Advantages of JEM on Toy Data

We start with toy datasets consisting of two rings or two half-moons, visualizing the results in Figure 1. We train using only 4 labeled examples. Our baseline classifier (VAT $\epsilon = 0.0$) achieves poor performance even with strong regularization. After a thorough hyperparameter search, VAT achieves strong performance on the moons dataset but fails on the rings dataset. Conversely, JEM is able to achieve 100% accuracy on both datasets. Full experimental details can be found in Appendix A.1.

We can intuitively understand why VAT fails on the rings data. All members of each class lie very close to the optimal decision boundary (in-between the rings). If VAT's $\epsilon$ is larger than this distance, this will encourage the classifier's decision to remain constant across this decision boundary,

| ALGORITHM | TEST ACCURACY |
|---|---|
| BASELINE CLASSIFIER | 86.0% ±1.6% |
| JEM | 95.4% ±0.3% |
| VAT | **98.4% ± 0.3%** |
| VAE (M1 + M2) | 96.7% ±0.1% |

*Table 1.* TEST ACCURACY FOR JEM, VAT, AND BASELINE CLASSIFIER ON MNIST WITH 100 LABELS.

| ALGORITHM | TEST ACCURACY |
|---|---|
| BASELINE CLASSIFIER | 62.7% ±0.5% |
| VAT | 62.8% ±0.6% |
| JEM | **66.0% ± 0.7%** |
| VAE (M1 + M2) | 64.0% ±0.1% |

*Table 2.* TEST ACCURACY FOR JEM, VAT, AND BASELINE CLASSIFIER ON SVHN WITH 1000 LABELS.

resulting in incorrect predictions. On the other hand, if $\epsilon$ is small, smoothness far from the labeled data cannot be enforced, leading to incorrect predictions on data far from the labeled examples. Conversely, JEM learns that the data density is relatively constant around both rings but low in-between, and places the decision boundary in the low density region between the two rings.

### 5.2. 100-Labels MNIST

The MNIST dataset with 100 labeled examples is a standard benchmark task for SSL algorithms. As in Miyato et al. (2018) we treat the data as permutation-invariant, meaning we do not use convolutional architectures. Baseline MLP architectures with strong regularization perform poorly (with a 14% error rate) when trained on only 100 examples. We show results averaged over 5 random seeds in Table 1. JEM significantly outperforms the baseline classifier (reducing the error rate below 5%). VAT performs best, possibly because of its stronger inductive bias. Surprisingly, JEM performs nearly as well despite making fewer modeling assumptions.

### 5.3. 1000-Labels SVHN

SVHN represents a more challenging dataset, with larger, more natural images. As with MNIST we treat this data in the permutation-invariant setting and do not use convolutional models. Results are shown in Table 2. On this dataset we again find JEM improves performance over the baseline classifier, demonstrating that JEM training provides benefits even when using models with limited inductive biases, limited expressive capacity, and on more challenging datasets.

JEM outperforms VAT and the VAE (Kingma et al., 2014). While the baseline, JEM and VAT share the same architecture, the stacked (M1 + M2) VAE model is deeper and wider, thus it is not directly comparable. Despite its strong performance on MNIST, we found VAT to provide only a marginal improvement on SVHN. We found smaller $\epsilon$ values to work well on this dataset compared to MNIST (1.0 compared to 4.0). Note that the VAT results reported here are with our MLP architecture, whereas the original VAT paper reports results using a Conv-Net architecture.

### 5.4. Tabular Data

We take two large datasets from the UCI dataset repository commonly used for regression (Gal & Ghahramani, 2016; Hernández-Lobato & Adams, 2015); Protein Structure Prediction and Year Prediction MSD. We convert them to classification tasks by binning the targets into 10 equally weighted buckets. We preprocess the inputs by standardizing each feature to have mean 0 and standard-deviation 1. We perform semi-supervised classification using a labeled subset with 100 examples and treat the remainder of the data as unlabeled. Results can be seen in Table 3. In this setting we find that VAT in fact decreases performance (for all hyperparameter settings tested) over the baseline. Conversely, JEM provides a modest improvement in test performance.

On tabular datasets such as these, the distributions of each of the inputs may be considerably different. This means that a different scale of sensitivity may be needed for each feature. VAT enforces invariance to perturbations of a given norm in any direction, weighting each feature equally. In the image domain, the per-pixel image statistics are roughly identical so this assumption may hold, explaining VAT's strong performance with images. This assumption does not hold on these tabular datasets, providing an explanation as to why VAT decreases performance over the baseline here.

| DATA (# UNLABELED) | BASELINE | JEM | VAT |
|---|---|---|---|
| PROTEIN (45,730) | 17.5 % | **19.6%** | 17.0 % |
| YEAR (515,345) | 15.6 % | **17.1%** | 13.1% |

*Table 3.* TEST ACCURACY FOR JEM, VAT, AND BASELINE CLASSIFIER ON TABUALR DATASETS WITH 100 LABELS.

## 6. Conclusion

We have shown that recent advances in energy-based models can be leveraged for SSL. This approach requires much less domain-specific knowledge compared to recent SSL approaches based on data-augmentation (Berthelot et al., 2019) or adversarial training (Miyato et al., 2018). JEM performs on par with VAT on multiple image datasets and outperforms it on domains other than images.

## References


Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Fetaya, E., Jacobsen, J.-H., Grathwohl, W., and Zemel, R. Understanding the limitations of conditional generative models. *arXiv preprint arXiv:1906.01171*, 2019.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pp. 1861–1869, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370*, 2019.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071. ACM, 2008.

Tramèr, F., Behrmann, J., Carlini, N., Papernot, N., and Jacobsen, J.-H. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. *arXiv preprint arXiv:2002.04599*, 2020.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In *International Conference on Machine Learning*, pp. 2635–2644, 2016.

# A. Experimental Details

## A.1. Toy Data Experiments

All networks had 4 layers with 500 units and used ReLU activations. All models were trained with the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.001 and default hyperparameters.

We experimented with dropout and batch normalization to regularize the baseline classifier and VAT but this did not improve accuracy.

For VAT, we search over choices of the perturbation size hyperparameter $\epsilon \in [0.01, 0.03, 0.1, 0.3, 1.0, 3.0]$. We find that $\epsilon = 0.03$ performed the best.

For JEM we apply slight L2 regularization on the energy outputs, which helps stabilize training; the same performance can be achieved without L2 regularization on the energy outputs. We set the strength of this regularization to 0.001.

## A.2. MNIST

For all models (baseline classifier, JEM, and VAT), we used a neural net consisting of a 4-layer MLP with 500 hidden units at each fully connected layer and ReLU activation function, and we applied preprocessing of 4-pixel padding, random crop, and logit transform ($\log(x) - \log(1 - x)$). We found the logit transform improved performance for all models (baseline classifier, JEM, and VAT). We trained over 200 epochs and report the test accuracy which corresponds to the epoch with highest validation accuracy. We used a learning rate of 0.0002 in all experiments.

Batch-norm and dropout were applied to the baseline classifier and VAT models. Entropy regularization (Miyato et al., 2018) was not found to be helpful for VAT or JEM (possibly because of our use of the logit transform).

VAT models had equal weighting of the regularization (LDS) loss and the classification loss.

VAE results were taken directly from (Kingma et al., 2014). For the M1+M2 model, the overall algorithm, including network architecture, preprocessing (the VAE uses PCA), and multi-stage training are different from our setup and thus results are not directly comparable.

For JEM we temper our MCMC sampler. This equates to using a larger stepsize for the SGLD sampler compared to the amount of noise added. We use stepsize $\alpha = 2.0$ and $\beta^2 = 0.01^2$. We use an equal weighting of the $p(\mathbf{x})$ loss and $p(y|\mathbf{x})$.

Hyperparameter search was done on the learning rate in all settings, weighting of the JEM objective, weighting of the LDS loss in VAT, epsilon used in VAT, and we report the best results in Table 1. Different activation functions (Leaky ReLU, Swish, Softplus) were not found to impact performance.

## A.3. SVHN

In all of our experiments (classifier, JEM, and VAT), we used a neural net consisting of a 3-layer MLP, with 1000 hidden units in each fully connected layer and ReLU activation function. We applied preprocessing of 4-pixel padding, random crop, normalization and Gaussian noise. We trained over 200 epochs and report the test accuracy which corresponds to the epoch with highest validation accuracy. We used a learning rate of 0.0002 in all experiments.

Batch-norm and dropout were applied to the baseline classifier and VAT models. For JEM we apply slight L2 regularization on the energy outputs, which greatly stabilizes training. We set the strength of this regularization to 0.01. MCMC sampling parameters are identical to our MNIST experiments.

## A.4. Tabular Data

Tabular data was pre-processed by standardizing each feature to have mean 0 and standard-deviation 1. The two datasets used are meant for regression tasks. The target values over the training set were binned into 10 equally weighted histograms to convert the regression task to a classification task. Labeled subsets were created by taking 10 examples at random from each of the 10 classes. A validation set of 100 examples was also selected in this way. All other data was treated as unlabeled.

All models used a 3-layer MLP with 500 hidden units and ReLU activations. No other pre-processing was used.

For the JEM models a small $l_2$ penalty was place on the energy with weight 0.01. We used a replay buffer of 10,000 examples and set the SGLD parameters $\alpha = 0.00125, \beta = 0.05$.

For VAT we searched over $\epsilon \in [0.01, 0.03, 0.1, 0.3, 1.0]$.

Models were trained for 100 epochs and we report the test accuracy that corresponds to the training epoch with highest validation accuracy.