APPENDIX A
SUPERVISED CLASSIFICATION FOR THE MNIST DATASET

The MNIST dataset consists of 28×28 pixel images of handwritten digits and their corresponding labels. The input dimension is therefore $28 \times 28 = 784$, and each label is one of the numerals from 0 to 9. The following list summarizes the ranges our hyper parameter search:

- RPT: $\epsilon = [1.0, 50.0]$,
- adversarial training (with L_∞ norm constraint): $\epsilon = [0.05, 0.1]$,
- adversarial training (with L_2 norm constraint): $\epsilon = [0.05, 5.0]$, and
- VAT: $\epsilon = [0.05, 5.0]$.

All experiments were conducted with $\alpha = 1$ except when checking the effects of α in Section 4.2. Training was conducted using mini-batch SGD based on ADAM [21]. We chose the mini-batch size of 100 and used the default values of [21] for the tunable parameters of ADAM. We trained the NNs with 60,000 parameter updates. For the base learning rate in validation, we selected the initial value of 0.002, and adopted the schedule of exponential decay with rate 0.9 per 600 updates. We repeated the experiments 10 times with different random seeds for the weight initialization and reported the mean and standard deviation of the results.

APPENDIX B
SUPERVISED CLASSIFICATION FOR CIFAR-10 DATASET

The CIFAR-10 dataset consists of $32 \times 32 \times 3$ pixel RGB images of categorized objects (cars, trucks, planes, animals, and humans). The number of training examples and test examples in the dataset are 50,000 and 10,000, respectively. We used 10,000 out of 50,000 training examples for validation and we applied ZCA whitening prior to the experiment. We also augmented the training dataset by applying random 2×2 translation and random horizontal flip. We trained the Conv-Large model (See Table7) over 300 epochs with batch size 100. For training, we used ADAM with essentially the same learning rate schedule as the one used in [36]. In particular, we set the initial learning rate of ADAM to be 0.003 and linearly decayed the rate over the last half of training. We repeated the experiments 3 times with different random seeds for the weight initialization and reported the mean and standard deviation of the results.

APPENDIX C
SEMI-SUPERVISED CLASSIFICATION FOR THE MNIST DATASET

For semi-supervised learning of MNIST, we used the same network as the network used for supervised learning; however, we added zero-mean Gaussian random noise with 0.5 standard deviation to the hidden variables during the training. This modification stabilized the training on semi-supervised learning with VAT. We experimented with two sizes of labeled training samples, $N_l \in \{100, 1000\}$, and observed the effect of N_l on the test error. We used the validation set of fixed size(1,000), and used all the training samples, excluding the validation set and labeled training samples, to train the NNs; that is, when N_l = 100, the unlabeled training set N_{ul} had the size of $60,000 - 100 - 1,000 = 58,900$.

We searched for the best hyperparameter ϵ from $[0.05, 10.0]$. All experiments were conducted with $\alpha = 1$ and $K = 1$. For the optimization method, we again used ADAM-based mini-batch SGD with the same hyperparameter values that we used in supervised setting. We note that the likelihood term can be computed from labeled data only.

We used two separate mini-batches at each step: one mini-batch of size 64 from labeled samples to compute the likelihood term, and another mini-batch of size 256 from both labeled and unlabeled samples to compute the regularization term. We trained the NNs over 100,000 parameter updates, and started to decay the learning rate of ADAM linearly after we 50,000-th update. We repeated the experiments 3 times with different random seeds for the weight initialization and for the selection of labeled samples. We reported the mean and standard deviation of the results.

APPENDIX D
SEMI-SUPERVISED CLASSIFICATION FOR THE SVHN AND CIFAR-10 DATASETS

The SVHN dataset consists of $32 \times 32 \times 3$ pixel RGB images of house numbers and their corresponding labels (0–9). The number of training samples and test samples within the dataset are 73,257 and 26,032, respectively. We reserved a sample dataset of size 1,000 for validation. From the remainder, we selected sample dataset of size 1,000 as a labeled dataset in semi-supervised training. Likewise in the supervised learning, we conducted ZCA preprocessing prior to the semi-supervised learning of CIFAR-10. We also augmented the training datasets with a random 2×2 translation. For CIFAR-10 exclusively, we also applied random horizontal flip as well. For the labeled dataset, we used 4,000 samples randomly selected from the training dataset, from which we selected 1,000 samples for validation. We repeated the experiment three times with different choices of labeled and unlabeled datasets on both SVHN and CIFAR-10.

For each benchmark dataset, we decided on the value of the hyperparameter ϵ based on the validation set. We also used a mini-batch of size 32 for the calculation of the negative log-likelihood term and used a mini-batch of size 128 for the calculation of $\mathcal{R}_{\mathrm{vadv}}$ in Eq. (8). We trained each model with 48,000 updates. This corresponds to 84 epochs for SVHN

TABLE 7: CNN models used in our experiments on CIFAR-10 and SVHN, based on [24], [36], [38]. All the convolutional layers and fully connected layers are followed by batch normalization [19] except the fully connected layer on CIFAR-10. The slopes of all lReLU [28] functions in the networks are set to 0.1.

Conv-Small on SVHN	Conv-Small on CIFAR-10	Conv-Large
32×32 RGB image		
3×3 conv. 64 lReLU	3×3 conv. 96 lReLU	3×3 conv. 128 lReLU
3×3 conv. 64 lReLU	3×3 conv. 96 lReLU	3×3 conv. 128 lReLU
3×3 conv. 64 lReLU	3×3 conv. 96 lReLU	3×3 conv. 128 lReLU
2×2 max-pool, stride 2 dropout, $p = 0.5$		
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 256 lReLU
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 256 lReLU
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 256 lReLU
2×2 max-pool, stride 2 dropout, $p = 0.5$		
3×3 conv. 128 lReLU	3×3 conv. 192 lReLU	3×3 conv. 512 lReLU
1×1 conv. 128 lReLU	1×1 conv. 192 lReLU	1×1 conv. 256 lReLU
1×1 conv. 128 lReLU	1×1 conv. 192 lReLU	1×1 conv. 128 lReLU
global average pool, 6×6 → 1×1		
dense 128 → 10	dense 192→ 10	dense 128→ 10
10-way softmax		

and 123 epochs for CIFAR-10. We used ADAM for the training. We set the initial learning rate of ADAM to 0.001 and linearly decayed the rate over the last 16,000 updates. The performance of CNN-Small and CNN-Large that we reported in Section 4.1.2 are all based on the trainings with data augmentation and the choices of ϵ that we described above.

On SVHN, we tested the performance of the algorithm with and without data augmentation, and used the same setting that we used in the validation experiments for both Conv-Small and Conv-Large. For CIFAR-10, however, the models did not seem to converge with 48,000 updates; so we reported the results with 200,000 updates. We repeated the experiments 3 times with different random seeds for the weight initialization and for the selection of labeled samples. We reported the mean and standard deviation of the results.