# Semi-Supervised Training of Conditional Score Models

Chandramouli S Sastry
Vector Institute and Dalhousie University.
chandramouli.sastry@gmail.com

Sri Harsha Dumpala
Vector Institute and Dalhousie University.
sriharsha.d.ece@gmail.com

Sageev Oore
Vector Institute and Dalhousie University.
osageev@gmail.com

## Abstract

*Score-based generative models have emerged as state-of-the-art generative models for both conditional and unconditional generation. In this paper, we propose and evaluate a framework for semi-supervised training of (classifier-guided) conditional score-models: more specifically, the training data consists of fewer labeled examples along with several unlabeled examples. We consider the following semi-supervised learning settings: a) classical multi-class semi-supervised learning setting wherein the labeled dataset consists of a few examples from each class; b) the binary positive-unlabeled learning setting wherein a few examples from the positive class are labeled and the unlabeled examples are a mix of positive and negative examples. We describe how the pretrained unconditional score-model can be used for label-efficient training of the classifier. We demonstrate that the resulting model can be used both as a conditional generative model as well as a classifier.*

## 1. Introduction

Score Models are unnormalized probabilistic models that model the probability density in terms of its score function – that is, the gradient of the log-likelihood. Score models are advantageous as compared to contemporary generative models such as GANs, VAEs, Autoregressive Models and Normalizing Flows as they require neither adversarial optimization nor restricted architecture families, while achieving state-of-the-art performance across multiple modalities. Score models can be trained using any score-matching objective such as the implicit score-matching [11], the sliced score-matching [24] or the denoising score-matching [27] methods.

In this work, we build upon Song et al. [25] which uses the denoising score-matching objective. As described in Section 2.2, class-conditional score-models can either be constructed using a pre-trained unconditional score-model by training a classifier separately, or the class-conditional score-model can be learnt directly with class-conditioning as input; these are referred to as classifier-guided and classifier-free models respectively. In this work, we use classifier-guided score models as this allows us to reuse a pre-trained larger score-model while training a smaller classifier-model: for example, Song et al. [25] generate class-conditional CIFAR10 examples using a score-model having 107M parameters and a classifier-model having 1.5M parameters.

In this paper, we propose a framework for semi-supervised training of classifier-guided conditional score-models: more specifically, the training data consists of a largely unlabelled dataset, along with a relatively small number of labelled examples. We consider the following semi-supervised learning settings: (a) *classical multi-class semi-supervised learning* wherein the labeled dataset consists of a few examples from each class, and (b) *binary positive-unlabeled learning* in which a few examples from the positive class are labeled and the unlabeled examples are a mix of positive and negative examples The final model can be used both as a classifier and a generative model.

## 2. Background

### 2.1. Unconditional Score-based SDE models

Score models are probabilistic models of the data that enable sampling and exact inference of log-likelihoods. Song et al. [25] propose a framework generalizing Multi-scale score matching [22, 23] and Denoising Diffusion Probabilistic Models [10]. Concretely, the framework consists of two components: 1) the forward-diffusion (i.e., data to noise) stochastic process, and 2) a learnable score-function that can then be used for the reverse-diffusion (i.e., noise to data) stochastic process.

The forward diffusion stochastic process $\{\mathbf{x}(t)\}_{t\in[0,T]}$, which starts at data and ends at a tractable noise distribution

such that the noise $\mathbf{x}(T)$ is independent of data $\mathbf{x}(0)$, is defined with a stochastic-differential-equation (SDE) of the following form

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ denotes a standard Wiener process. The drift coefficient $\mathbf{f}(\mathbf{x}, t)$ and diffusion coefficient $g(t)$ are usually manually specified without learnable parameters such that we can obtain closed-form solutions to the forward-diffusion SDE. For example, if $\mathbf{f}$ is linear in $\mathbf{x}$, the solution to the SDE is a gaussian distribution whose mean $\mu(t)$ and standard deviation $\sigma(t)$ can be exactly computed. We use $p_t(\mathbf{x}|\mathbf{x}_0)$ to denote the probability density function of $\mathbf{x}(t)$ when the diffusion is seeded at $\mathbf{x}_0$, and we denote the marginal probability density function of $\mathbf{x}(t)$ by $p_t(\mathbf{x})$.

In order to generate samples from $p_0(\mathbf{x})$ starting with samples from $p_T(\mathbf{x})$, we have to solve the following reverse diffusion SDE [2]:

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\, dt + g(t)\, d\bar{\mathbf{w}}, \tag{2}$$

where $d\bar{\mathbf{w}}$ is a standard Wiener process when time flows from T to 0, and $dt$ is an infinitesimal negative timestep. In practice, the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is estimated by a neural network $s_\theta(\mathbf{x}, t)$ that is trained to optimize the following score-matching loss:

$$\int_0^T \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}[\lambda(t)||\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, t)||_2^2]dt \tag{3}$$

where $\lambda(t)$ is a positive real number introduced to balance out the score-matching objective across various time steps. Using samples from the training dataset, we can define an empirical density function for $t = 0$ as $p_0(\mathbf{x}) = \frac{1}{N}\sum_{\mathbf{x}_i \in p_{data}} \delta(\mathbf{x} - \mathbf{x}_i)$ and then obtain samples from $p_t(\mathbf{x})$ by first sampling $\mathbf{x}(0) \sim p_0$ and then solving the forward-diffusion SDE (Eq. 1). If the solution to the SDE is a Gaussian distribution whose means and covariances can be determined in a closed-form, we can empirically define $p_t(\mathbf{x})$ as a mixture of $N$ gaussians; for such SDE's, we can also estimate the score-function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ in the closed-form for evaluating the score-matching loss: this is usually referred to as denoising score matching as the score-function points in the denoising direction.

## 2.2. Class-Conditional Score-based SDE models

Given a data distribution whose samples can be classified into $C$ classes, class-conditional score-models are trained to estimate $\nabla_{\mathbf{x}} \log p(\mathbf{x}, t|y)$ where $y \in [1, C]$ is the class label. Classifier-free conditional models directly learn $s_\theta(\mathbf{x}, t|y)$ by taking $y$ as an additional input. On the other hand, *Classifier-guided* conditional models learn the probability distribution $p(y|\mathbf{x}, t)$ using a classifier and then combine this with the learnt unconditional score (i.e., $s_\theta(\mathbf{x}, t)$) using Bayes rule: $p(\mathbf{x}, t|y) = \frac{p(y|\mathbf{x},t)p(\mathbf{x},t)}{p(y)}$: applying log on both sides and taking the derivative with respect to $\mathbf{x}$, we get

$$s_\Theta(\mathbf{x}, t|y) = \nabla_{\mathbf{x}} \log p_\phi(y|\mathbf{x}, t) + s_\theta(\mathbf{x}, t) \tag{4}$$

where $\phi$ denotes the parameters of the classifier and $\Theta = \{\theta, \phi\}$. Song et al. [25] suggest a simple sum of the cross-entropy losses sampled at different scales for training the classifier $p_\phi$:

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,T) \\ (\mathbf{x}_0,y)\sim p_0(\mathbf{x}) \\ \mathbf{x}\sim p_t(\mathbf{x}|\mathbf{x}_0)}} [-\log p_\phi(y|\mathbf{x}, t)] \tag{5}$$

However, subsequent studies such as [7] have identified that the score $s_\Theta(\mathbf{x}, t|y)$ does not yield good conditional samples and propose scaling up the classifier gradient in order to produce higher fidelity samples at the cost of diversity. Chao et al. [5] refer to this as the score-mismatch issue and instead suggest adding the following Denoising Likelihood Score Matching (DLSM) term to the classifier training:

$$\mathcal{L}_{\text{DLSM}} = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,T) \\ (\mathbf{x}_0,y)\sim p_0(\mathbf{x}) \\ \mathbf{x}\sim p_t(\mathbf{x}|\mathbf{x}_0)}} [\lambda(t)||\nabla_{\mathbf{x}} \log p_t(\mathbf{x}|y) - s_\Theta(\mathbf{x}, t|y)||_2^2] \tag{6}$$

In this work, we follow [5] and use the sum of cross-entropy loss $\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{DLSM}}$ for training the classifier $p_\phi$.

## 2.3. Semi-Supervised Learning

Semi-supervised learning aims to train machine learning models by using a predominantly unlabelled training set, where only a small proportion of the data are labeled. The general strategy in semi-supervised learning is to bootstrap the learning process using labeled data and to then use unlabeled data along with their label *guesses* as additional labeled training samples; in order to prevent overfitting, the training is often accompanied by regularization and data augmentations wherever applicable. Semi-supervised learning algorithms for generative models and classifier models usually differ due to the architecture constraints (e.g., invertibility in Normalizing Flows) and loss-objectives (e.g., Adversarial Loss for GANs, ELBO Loss for VAEs). In classifier-guided score models, however, the training objective is almost identical to training a classifier and we briefly review state-of-the-art semi-supervised methods for training classifiers in the Supplementary Material. In summary, current top-performing semi-supervised methods like MixMatch [4], UDA [28] and FixMatch [21] for classifiers derive their improvements using MixUp augmentation or strong augmentations for training; however, using strongly augmented or mixed-up images as input to the classifier model for computing the classification and score-matching objective would cause the model to learn to generate from the distribution of strongly-augmented images instead of clean images.

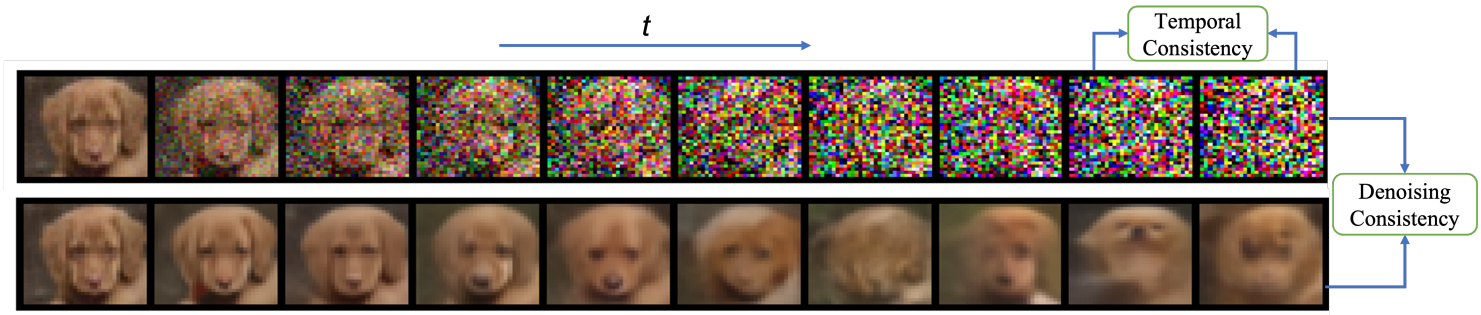


Figure 1. The upper row of images shows the progression of noisy samples obtained through the forward diffusion process. Each image in the second (lower) row shows the result of applying one step of denoising (guided by the score network) to the image directly above it. Two loss terms, both applied as an L2 distance penalty at the penultimate layer, help the classifier's representation learning from a set of combined labeled and unlabeled examples: (1) *Denoising Consistency* encourages both of these images to have a similar representation. (2) *Temporal Consistency* encourages images that are close in diffusion time (i.e. in noise level) to have a similar representation. Taken together, the temporal and denoising consistency losses effectively propagate information across all noise scales and their corresponding denoised samples, thus reconciling the learned representations.

### 2.4. Semi-Supervised Generative Nets

Kingma et al. [13] represents one of the early works on semi-supervised training of joint classifier and generative models with a Variational Autoencoder. FlowGMM is an elegant method for training Normalizing Flows in a semi-supervised setting wherein they propose to maximize marginal likelihoods for unlabeled data and maximize class-conditional likelihoods for labeled data. D2C [20], Diffusion-AE [18], and FSDM [9] represent some of the recent efforts on few-shot diffusion models. D2C is a latent-variable conditioned decoder whereas Diffusion-AE is a latent-variable conditioned Diffusion model: in order to introduce class-conditioning, these models train a classifier using the frozen latent-representations and use rejection-sampling for class-conditional generation. Similar to Diffusion-AE, FSDM is a latent-variable conditioned diffusion model that uses Vision Transformer to encode a set of images into a conditioning vector. While FSDM does not support inference of classes on test examples, the classification accuracy in D2C and Diffusion-AE is limited as the latent-variable encoder is frozen and cannot be fine-tuned without retraining the entire pipeline. In contrast, we use a vanilla diffusion model with a flexible classifier architecture to introduce conditioning.

## 3. Semi-supervised Conditional Score Models

In this section, we describe our framework for learning Conditional Score Models with partial supervision. We consider two main settings of partial supervision: the classical semi-supervised learning setting and the positive-unlabeled learning setting. In the classical semi-supervised learning setting, the training data consists of labeled and unlabeled examples, wherein the labeled examples consists of a few samples from each class. On the other hand, the positive-unlabeled learning setting involves learning a binary classifier trained on labeled positive samples, together with an unlabeled training set containing samples from both positive and negative classes. Typically, a positive-unlabeled learning problem is solved in two steps: a) estimating the proportions of positive and negative classes— this is known as the mixture proportion estimation step; and b) training a binary classifier using this information. In this work, we follow previous work (e.g., Acharya et al. [1]) and primarily focus on training a binary classifier assuming that the mixture proportion estimation step has already been performed and the class prior is known.

Consider a time-conditional classifier network $p_\phi : \mathbb{R}^{D+1} \rightarrow \mathbb{R}^C$ that takes $\mathbf{x} \in \mathbb{R}^D$ and $t \in [0, T]$ as input. The core idea is to minimize $\mathcal{L}_{\text{Total}}$ over both labeled and unlabeled data: for the unlabeled samples, we infer the labels using the classifier $p_\phi$ on clean samples and use them for computing $\mathcal{L}_{\text{Total}}$.

Let $\mathbf{x}_L$ and $\mathbf{x}_U$ denote the labeled and unlabeled samples. For the labeled samples, we can directly estimate the $\mathcal{L}_{\text{Total}}$ using ground truth labels obtained from the dataset. We derive pseudo-labels for unlabeled examples using confidence thresholding following FixMatch and use these for computing the $\mathcal{L}_{\text{Total}}$ on the unlabeled examples. In addition, we implement the following consistency losses (Figure 5):

- Denoising Consistency Loss: Consider a sample $\mathbf{x} \sim p_t(\mathbf{x}|\mathbf{x}_0)$ such that $\mathbf{x}_0 \sim \mathbf{x}_U$ and the corresponding denoised example $\tilde{\mathbf{x}} = \mathbf{x} + \sigma(t)^2 s_\theta(\mathbf{x}, t)$. Noting that the score-network "sees" the image $\tilde{x}$ in the noisy input $\mathbf{x}$, we propose that $p_\phi(y|\mathbf{x}, t)$ and $p_\phi(y|\tilde{\mathbf{x}}, t)$ should be identical. Furthermore, this aligns the classifier with the score-model in the sense that the classifier learns how the score-model would denoise the given image and can adjust the class-conditional score accordingly.

- Temporal Consistency Loss: Consider a sample $\mathbf{x}_1 \sim p_t(\mathbf{x}|\mathbf{x}_0)$ such that $\mathbf{x}_0 \sim \mathbf{x}_U$ and another sample $\mathbf{x}_2 \sim p_{t+\delta(t)}(\mathbf{x}|\mathbf{x}_0)$ where, for some small threshold $\Delta$, $\delta(t)$ is chosen such that $\sigma(t+\delta(t))-\sigma(t) \leq \Delta$. We regulate that $p_\phi(y|\mathbf{x}_1, t)$ and $p_\phi(y|\mathbf{x}_2, t)$ should be identical.

We enforce these consistency losses by minimizing the L2 distance of the network representations in the penultimate layer. In all our experiments, we use a confidence threshold of 0.95 to generate pseudo-labels and $\Delta = 0.01$ for the temporal consistency loss.

**PU Learning** In the positive unlabeled setting, we additionally minimize the cross-entropy between the class-averages obtained on unlabeled examples diffused to time $\tau$ and the supplied class prior.

## 4. Experiments

We evaluate our framework on MNIST, SVHN and CIFAR10 datasets in the classic semi-supervised setting and compare with both generative and discriminative models trained in a semi-supervised setting. We use SSL-VAE and FlowGMM as the baselines for generative semi-supervised methods and $\Pi$ Model [19], Pseudo-Labelling [15], Mean Teacher [15], MixMatch [4] and FixMatch [21] as baselines for discriminative semi-supervised methods. We used the VE-SDE for the forward-diffusion as defined in [25] with the noise scale $\sigma_t$ ranging from 0.01 to 50.0. We use NCSN++ network for the unconditional score network $s_\theta$: for MNIST and SVHN, we train a 62.8M parameter network for learning $s_\theta$ while we used the pretrained checkpoint of the deeper NCSN++ network containing 107M parameters for CIFAR10 – open-sourced by [25]. For the classifier network $p_\phi$, we use WideResNet 28-2 with 1.5M parameters and use InstanceNormPlus (see [22]) instead of BatchNorm for the normalization. We trained the classifier network using the AdamW optimizer with a learning rate of 1e-3 and weight decay set to 5e-4: for MNIST and SVHN, we trained the network for 48k steps while we trained the network for 200k steps for CIFAR10 reducing the learning rate to 2e-4 from 130k steps onwards. For all datasets, we used a labeled batch-size of 64 and unlabeled batch-size of 192.

The semi-supervised classification accuracies are summarized in Table 1: we report the average over 3 runs. We observe that our model outperforms the generative modeling baselines in terms of classification accuracy while remaining competitive with the discriminative semi-supervised models.

For the Positive-Unlabeled experiments, we conduct experiments on MNIST and SVHN by selecting one of the 10 classes as the positive class. We report the F1-scores for different proportions of labels in Figure 2: we observe that our model generalizes well given few positive examples and class prior. We also compare our model accuracy with other PU baselines in Table 2: here, the classifier is trained to classify between odd and even digits and 1k odd examples are provided as positive samples.

| Method | Dataset ($n_l/n_u$) | | |
|---|---|---|---|
| | MNIST ($1k/59k$) | SVHN ($1k/72k$) | CIFAR10 ($4k/46k$) |
| SSL-VAE [13] | 97.6 | 63.98 | - |
| FlowGMM [12] | **99.0** | 86.44 | 80.9 |
| Score-SSL (Ours) | **99.1** | **96.2** | **87.3** |
| $\Pi$ Model[19] | - | 92.46 | 85.99 |
| Pseudo-Labelling[15] | - | 90.96 | 83.91 |
| Mean Teacher[26] | - | 96.58 | 90.81 |
| MixMatch[4] | - | 96.5 | 93.58 |
| FixMatch[21] | - | 97.72 | 95.74 |

Table 1. Semi-supervised Classification Accuracy: The table shows the semi-supervised classification accuracies with $n_l$ labels. The first block includes semi-supervised generative models as baselines whereas the second block includes accuracies from standard semi-supervised discriminative models for reference.

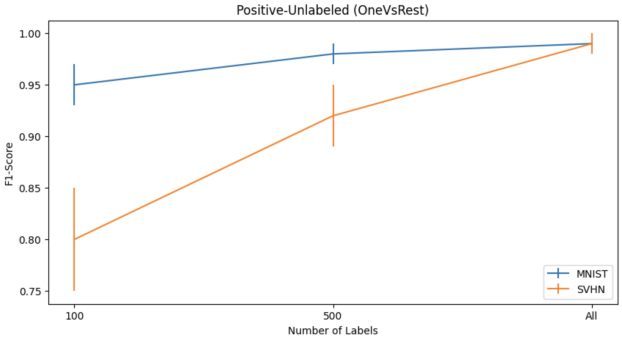


Figure 2. Positive-Unlabeled Learning Results: The graph shows the F1-scores on MNIST and SVHN for OneVsRest Positive-Unlabeled training setup. Specifically, we select one of the 10 classes as positive, label 100 or 500 of them and treat the remaining as negative. We report the mean and variance of the F1-score across 10 models. For reference, we also show the F1-score when the entire training data is available.

| | PU-MNIST (OddvsEven) |
|---|---|
| PvU [8] | 91.10±0.92 |
| uPU [17] | 91.14±0.87 |
| nnPU [14] | 91.83±0.79 |
| puNCE [1] | 94.7±0.19 |
| Score-SSL(Ours) | **98.8±0.05** |

Table 2. Classification accuracy results on PU-MNIST: We randomly choose 1k examples of Odd digits as positive examples and treat the rest as unlabeled. We repeat the experiment 3 times.

We have included the generated class-conditional images in the supplementary material.

**Conclusion** In this work, we propose a new framework for semi-supervised training of class-conditional scores: importantly, it allows us to make use of a pretrained score-network in defining the denoising consistency loss that helps improve the classifier generalization. Our evaluations show that the test-accuracies are better than previous semi-supervised generative models and comparable to state-of-the-art semi-supervised discriminative methods.

## References


[1] A. Acharya, S. Sanghavi, L. Jing, B. Bhushanam, D. Choudhary, M. Rabbat, and I. Dhillon. Positive unlabeled contrastive learning, 2022. 3, 4

[2] B. D. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 2

[3] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 6

[4] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 2, 4, 6

[5] C.-H. Chao, W.-F. Sun, B.-W. Cheng, Y.-C. Lo, C.-C. Chang, Y.-L. Liu, Y.-L. Chang, C.-P. Chen, and C.-Y. Lee. Denoising likelihood score matching for conditional score-based data generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LcF-EEt8cCC. 2

[6] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017. 6

[7] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf. 2, 6

[8] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008. 4

[9] G. Giannone, D. Nielsen, and O. Winther. Few-shot diffusion models, 2022. 3

[10] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 1

[11] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html. 1

[12] P. Izmailov, P. Kirichenko, M. Finzi, and A. G. Wilson. Semi-supervised learning with normalizing flows, 2019. 4

[13] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models, 2014. 3, 4

[14] R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30, 2017. 4

[15] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 4

[16] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. 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. 6

[17] G. Niu, M. C. du Plessis, T. Sakai, Y. Ma, and M. Sugiyama. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. *Advances in neural information processing systems*, 29, 2016. 4

[18] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[19] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015. 4

[20] A. Sinha, J. Song, C. Meng, and S. Ermon. D2c: Diffusion-denoising models for few-shot conditional generation, 2021. 3

[21] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2, 4, 6

[22] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019. 1, 4

[23] Y. Song and S. Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020. 1

[24] Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020. 1

[25] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS. 1, 2, 4

[26] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 4

[27] P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 1

[28] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 2, 6

## A. Background: Semi-Supervised Classification Methods

At a high-level, discriminative semi-supervised methods can be differentiated in terms of their strategy to obtain label guesses (also referred to as pseudo-labels) and in how the unlabeled examples along with their labeled guesses are introduced into the training loss. For example, MixMatch [4] applies $K$ augmentations to an unlabeled image to form an expectation of the label which is then 'sharpened' and used as the pseudo-label; finally, the labeled and unlabeled examples undergo MixUp before being used to compute the classification loss. Different from MixMatch, FixMatch[21] selects unlabeled images whose confidence is above some pre-determined threshold and assigns pseudo-labels; finally, the classification loss is computed on strongly augmented (i.e., heavily distorted) images instead of the clean images. Other SOTA SSL algorithms such as ReMixMatch [3] and UDA [28] also use strong-augmentation before computing the classifier loss; in fact, VAT [16] and BadGAN [6] suggest that training of semi-supervised classifier benefits most from corrupted images instead of clean images. However, using strongly augmented or mixed-up images as input to the classifier model and using them in computing the classification and score-matching objective would cause the model to learn to generate from the distribution of strongly-augmented images instead of clean images.

## B. Generated Images

In the following, we show a grid of images sampled from the unconditional score-network with guidance from the semi-supervised classifiers: for each class, we sample 10 images. We scale the classifier-gradients with $s = 4.0$ following Dhariwal and Nichol [7]. We trained the MNIST and SVHN classifiers with 1k labeled examples while the CIFAR10 classifier was trained with 4k labeled examples.

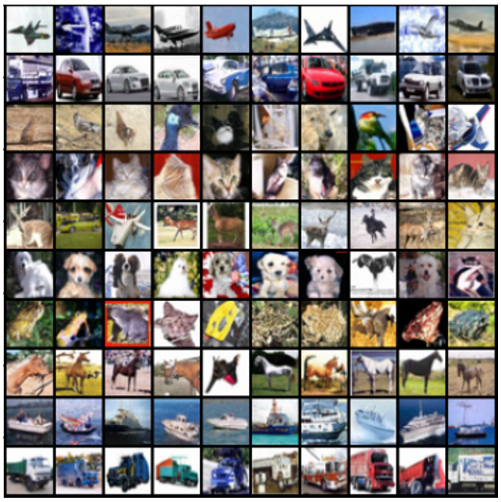

Figure 3. CIFAR10 Samples

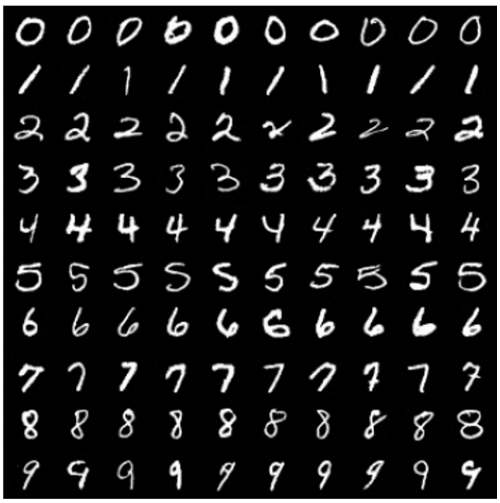

Figure 4. MNIST Samples

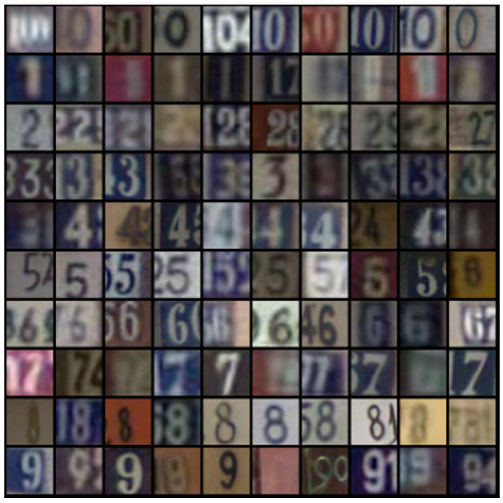

Figure 5. SVHN Samples