Deep Hybrid Models: Bridging Discriminative and Generative Approaches

Volodymyr Kuleshov
Department of Computer Science
Stanford University
Stanford, CA 94305

Stefano Ermon
Department of Computer Science
Stanford University
Stanford, CA 94305

Abstract

Most methods in machine learning are described as either discriminative or generative. The former often attain higher predictive accuracy, while the latter are more strongly regularized and can deal with missing data. Here, we propose a new framework to combine a broad class of discriminative and generative models, interpolating between the two extremes with a multi-conditional likelihood objective. Unlike previous approaches, we couple the two components through shared latent variables, and train using recent advances in variational inference. Instantiating our framework with modern deep architectures gives rise to deep hybrid models, a highly flexible family that generalizes several existing models and is effective in the semi-supervised setting, where it results in improvements over the state of the art on the SVHN dataset.

1 INTRODUCTION

Modern machine learning techniques rely on the availability of large labeled datasets to achieve state-of-the-art results on tasks such as image classification (Deng et al., 2009), speech recognition (Amodei et al., 2015), and machine translation (Wu et al., 2016) However, obtaining such datasets is often expensive, which has driven significant interest in semi-supervised learning, a class of methods that can leverage unlabeled data to improve the performance of supervised classifiers (Chapelle et al., 2006).

Most semi-supervised algorithms (and, in fact, most machine learning algorithms) can be categorized into one of two general approaches, which differ in how they model input features x and their target labels y (Ng & Jordan, 2002).

Generative methods describe the relationship between x and y using a joint probability distribution $p(x, y)$. As a result, they handle arbitrary queries about the data, such as predicting unknown labels via $p(y|x)$, or imputing missing features x using the distribution $p(x)$. In the semi-supervised setting, they treat y as an unobserved latent variable and optimize the marginal likelihood of the data.

Discriminative methods, on the other hand, focus only on directly predicting y from x via the conditional distribution $p(y|x)$. If prediction is the only goal, this approach will use the model parameters more efficiently, and be more accurate on larger datasets (Ng & Jordan, 2002).

A Framework for Hybrid Models. Choosing between generative and discriminative techniques is a fundamental problem in machine learning. In this work, we propose a framework for designing probabilistic models that can interpolate between a purely generative and a purely discriminative approach. In a semi-supervised context, this enables us to jointly apply both categories of semi-supervised learning algorithms.

A standard approach proposed by several authors (McCallum et al., 2006; Lasserre et al., 2006) is to specify a joint model $p(x, y)$ and assign different weights to the posterior $p(y|x)$ and the marginal $p(x)$ during training. Clearly, this is possible only for simple models in which computing and optimizing $p(y|x)$ and $p(x)$ is tractable. Furthermore, sharing weights between $p(y|x)$ and $p(x)$ limits our modeling flexibility: it is unclear how to tie via shared parameters complex models such as modern neural networks.

Our approach allows choosing very general forms for $p(y|x)$ and $p(x)$ and instead couples them by introducing shared *latent variables* z into the joint model $p(x, y, z)$. The z can be thought of as a latent high-level representation useful for both the discriminative and generative components. We learn the resulting latent-variable model using approximate variational inference, which allows us to handle a wider range of models that otherwise would have been intractable.

Deep Hybrid Models. Instantiating our framework with modern deep architectures gives rise to deep hybrid models (DHMs), a highly flexible family that generalizes several

existing deep learning algorithms. For example, a DHM can be defined by modeling the joint probability $p(x, z)$ as a variational auto-encoder (Kingma & Welling, 2013) or a generative adversarial network (Goodfellow et al., 2014), while the discriminative model $p(y|x, z)$ can be a highly flexible convolutional neural network. Our only constraint is that both share the same latent z.

We train the discriminative and the generative component jointly using an objective that places different weights upon the two. This procedure can improve both discriminative accuracy as well the generative log-likelihood by sharing the latent representation z across both models.

Contributions. Our work proposes a new framework for training a broad class of hybrid discriminative-generative models, thus combining the strengths of the two modeling approaches. The practical advantages of our framework over existing approaches are:

- Greater flexibility in the specification of the hybrid model.
- Ability to handle modern machine learning methods based on deep neural networks.
- Compatibility with complex latent-variable models trained using approximate variational inference.
- Ability to apply discriminative and generative semi-supervised learning algorithms on the same model.

Outline. We give a background on generative versus discriminative models and their hybrid extensions in Section 2. In Section 3, we discuss the shortcomings of these methods and introduce our approach and its advantages. We perform an empirical study of our method in Section 4, and conclude with a discussion in Section 5.

2 BACKGROUND

2.1 DISCRIMINATIVE VS. GENERATIVE MODELS

Machine learning models can often be seen as expressing a relationship between features x (e.g., email messages) and labels y (e.g., whether they are spam or not) using a probability distribution p over x, y. Depending of the form of p, these models are characterized as being either generative and discriminative (Ng & Jordan, 2002).

Generative Models. A generative model learns the full relationship between the labels y and the features x, as captured by the full joint distribution $p(x, y)$. This gives the model p maximum flexibility at test-time: we may use $p(y|x)$ to predict labels y from x, impute a missing feature x_i from the other features x_{-i} using $p(x_i|x_{-i})$, as well as sample new data x from $p(x)$. Generative models also offer more flexibility to specify our prior inductive biases; encoding this prior knowledge into p may let us use scarce data more efficiently.

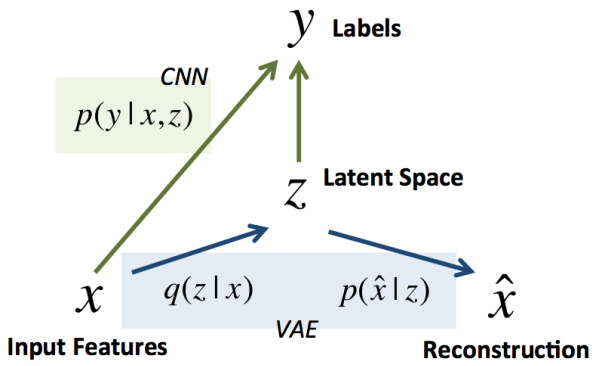

Figure 1: Computational graph describing a deep hybrid model $p(y|x, z)p(x, z)$, whose generative component $p(x, z)$ (blue) is parametrized by a variational auto-encoder with recognition network $q(z|x)$ and whose discriminative component $p(y|x, z)$ (green) is parametrized by a convolutional neural network. We train both models jointly, placing adjustable weights on the two components. The shared representation from $q(z|x)$ is used by both parts, effectively acting as a coupling agent.

Discriminative Models. If, on the other hand, we are only interested in predicting y from x, it is sufficient to fit a conditional model $p(y|x)$; we refer to this approach as discriminative modeling. Since discriminative models are not concerned with modeling $p(x)$, they may use their parameters more efficiently to capture the relationship $p(y|x)$. This makes them more suitable for purely supasdervised learning. Furthermore, by making fewer modeling assumptions, they may use data more efficiently. However, discriminative models are not suited for tasks like imputation.

2.2 HYBRID DISCRIMINATIVE-GENERATIVE MODELS

The inherent tradeoffs between discriminative and generative modeling have led to the development of hybrid models that attempt to combine the advantages of the two methods. There have been two main approaches to formulating such hybrid models.

Multi-Conditional Learning. Given a probabilistic model $p(x, y; \theta)$ with parameters θ, McCallum et al. (2006) propose optimizing the *multi-conditional likelihood*

$$\alpha \log p(y|x; \theta) + \beta \log p(x; \theta). \quad (1)$$

Here, $\log p(y|x; \theta)$ and $\log p(x; \theta)$ are, respectively, the posterior over y given x and the marginal over x that are

both derived from the model $p(x, y; \theta)$. The scalar weights $\alpha, \beta \geq 0$ control the weights assigned to the generative and discriminative components.

Crucially, for any $\alpha, \beta > 0$, we are optimizing a single $p(x, y; \theta)$, albeit with a novel objective. The model p is typically chosen to be a Markov Random Field for which $p(y|x; \theta)$ and $p(x; \theta)$ are tractable to compute and optimize.

Bayesian Parameter Coupling. Lasserre et al. (2006) propose an alternative formulation derived from a Bayesian perspective. Consider a joint model $p(x, y)$ with parameters θ and consider two independent parameter vectors θ_d, θ_g of the same type as θ. Lasserre et al. (2006) propose optimizing the Bayesian model

$$p(x, y, \theta_d, \theta_g) = p_{\theta_d}(y|x)p_{\theta_g}(x)p(\theta_d, \theta_g), \qquad (2)$$

where $p(\theta_d, \theta_g)$ is a *parameter coupling prior* that defines a joint distribution over x, y, θ_d, θ_g. Observe that when $p(\theta_d, \theta_g)$ is constant (i.e., the parameter sets are not tied), then we are effectively optimizing two independent models. At test-time, predictions for y will come from a purely discriminative model. Conversely, when $p(\theta_d, \theta_g)$ forces the two sets of weights to be identical, we are optimizing a single generative model.

2.3 SEMI-SUPERVISED LEARNING

In many applications of machine learning, labeled data is scarce, but we have access to large amounts of unlabeled data. Semi-supervised learning (Chapelle et al., 2006) aims to leverage this unlabeled data to improve the performance of purely supervised classifiers.

Discriminative Approaches. One general approach in semi-supervised learning consists in augmenting a discriminative classifier $p(y|x)$ with a regularizer whose goal is generally to place the decision boundary further away from the unlabeled data. Transductive SVMs (Chapelle et al., 2006) define an extension of the hinge loss for unlabeled examples. Entropy regularization (Grandvalet & Bengio, 2004) minimizes the entropy of $p(y|x)$, effectively encouraging the classifier to be maximally certain on unlabeled points. Recently, Laine & Aila (2016) proposed adding a regularizer that encourages the stability of a classifier's predictions over time.

Generative Approaches. Generative semi-supervised models instead formulate semi-supervised learning in the framework of latent variable models, where the variable y is treated as a latent variable for the unlabeled samples $\mathcal{U}$ and as a regular variable for labeled examples $\mathcal{L}$:

$$\sum_{x_i, y_i \in \mathcal{L}} \log p(x_i, y_i) + \sum_{x_i \in \mathcal{U}} \int_y \log p(x_i, y).$$

Recent algorithms implementing this approach include semi-supervised variational auto-encoders (Kingma & Welling, 2013) and auxiliary variable deep generative models (Maaløe et al., 2016).

Approximate Inference. Latent variable models are often optimized using the framework of *variational inference*. The idea is to approximate the intractable marginal likelihood $\log p(x) = \log \int_z p(k, z)$ via a variational lower bound

$$\log p(x) \geq \mathbb{E}_{q(z|x)} \left[\log p(x, z) - q(z|x) \right],$$

where $q(z|x)$ is an *approximate posterior* distribution over which we optimize. The difference between the above two terms can be shown to equal $KL(q(z|x)||p(z|x))$. Our work will heavily rely on this variational approach.

3 HYBRID MODELS VIA LATENT VARIABLE COUPLING

In this section, we propose a new way of interpolating between discriminative and generative models. We refer to our approach as *latent-variable coupling*.

Our framework allows the user to choose very general forms for $p(y|x)$ and $p(x)$; we only require them to be contain shared *latent variables* z. The z can be thought of as a latent high-level representation useful for both the discriminative and generative tasks. We learn the resulting joint model $p(x, y, z)$ using approximate variational inference, but assign different weights to each of the two components.

Compared to the previous work of McCallum et al. (2006) and Lasserre et al. (2006), our framework offers much greater modeling flexibility and scales to larger models, particularly ones that make significant use of latent variables. We present the details of this approach below.

3.1 GENERAL FRAMEWORK

Representation. Consider a generative probabilistic model $p(x, y, z)$ over variables $x, y, z \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. The x are fully-observed, while the y are labels that we only see during training, and the z are unobserved latent variables.

We assume that $p(x, y, z)$ has a parametric form specified by the decomposition

$$p(x, y, z) = p(y|x, z) \cdot p(x, z), \qquad (3)$$

where $p(y|x, z)$ and $p(x, z)$ are, respectively, the discriminative and generative components of the model and are directly specified by the user.

Dealing with latent variable models requires us to make use of approximate variational inference and to optimize an approximate posterior $q(z|x)$ to the true learning objective.

Slightly abusing notation, we will use $q(x, y, z)$ to denote the product of the (unknown) data distribution $q(x, y)$ and the user-specified approximate posterior $q(z|x)$; this will let us succinctly represent variational inference algorithms as minimizing a divergence $D\left(q(x, y, z)||p(x, y, z)\right)$ between q and p.

Learning. At training time, we are given a set of labeled examples $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ sampled from a data distribution $q(x, y)$; our model also naturally extends to the semi-supervised setting, as we shall see later. We propose training p using a multi-conditional learning objective (McCallum et al., 2006) of the form:

$$\alpha \cdot L_D\left[q(x, y, z), p(y|x, z)\right] + \beta \cdot L_G\left[q(z, x), p(x, z)\right], \tag{4}$$

where L_D and L_G are two functionals specifying losses for a given choice of p that depend on the true data distribution $q(x, y)$ and the approximate posterior $q(z|x)$.

The L_G term focuses on the generative component of the model and may be any f-divergence between $q(x, z)$ and $p(x, z)$ (Nowozin et al., 2016):

$$L_G\left[q(z, x), p(x, z)\right] = D_f\left(q(z, x)||p(x, z)\right). \tag{5}$$

We optimize this functional over $q(z|x)$ and $p(x, z)$; the minimum is attained by $q(z|x) = p(z|x)$ and $p(x) = q(x)$.

The L_D term lets us fit the discriminative component of the model. We may choose L_D to be any classification or regression loss, whose expectation is taken over $q(x, y, z)$:

$$L_D = \mathbb{E}_{q(x,y)}\mathbb{E}_{q(z|x)}\ell\left(y, p(y|x, z)\right). \tag{6}$$

For example, ℓ may be the log-loss, the ℓ_2-loss, a max-margin objective, or a ranking loss. This objective encourages the discriminative model to achieve high prediction accuracy.

Following the multi-conditional learning framework of McCallum et al. (2006), we assign scalar weights $\alpha, \beta > 0$ to each of the two loss functionals. By shifting weight from α to β, we may smoothly interpolate between a fully discriminative model and a fully generative model that ignores labels y. In between these two extremes is a generative model $p(x, y, z)$ that assigns equal weight to $p(y|x, z)$ and $p(x, z)$.

Next, we look at how to define $p(x, y, z)$ and optimize L_G, L_D. We propose two instantiations of our framework that are compatible with many common models and that are tractable to optimize.

3.1.1 Explicit Density Models

The standard approach for training p is to maximize the *marginal likelihood*

$$\log p(x, y) = \log \int_{z \in \mathcal{Z}} p(x, y, z). \tag{7}$$

In order to trade off the discriminative and generative components of the model, we may follow the approach of McCallum et al. (2006) and optimize the multi-conditional likelihood

$$\log \int_{z \in \mathcal{Z}} p(y|x, z)^\gamma p(x, z). \tag{8}$$

This objective assigns different relative weights to the discriminative and generative components of the model via a scalar $\gamma > 0$.

Unfortunately, this objective is intractable given our assumptions on p. We therefore apply the variational principle to obtain a lower bound:

$$\begin{aligned}
&\log \int_{z \in \mathcal{Z}} p(y|x, z)^\gamma p(x, z) \\
&= \log \int_{z \in \mathcal{Z}} \frac{p(y|x, z)^\gamma p(x, z)}{q(z|x)} q(z|x) \\
&\geq \mathbb{E}_{q(z|x)}\left[\gamma \log p(y|x, z) + \log p(x, z) - \log q(z|x)\right]
\end{aligned} \tag{9}$$

Note that for $\gamma = 1$, Equation 9 reduces to the standard evidence lower bound (ELBO). When $\gamma \neq 1$, this bound may no longer be tight.

The above objective is a special case of our framework if we choose

$$L_D = \mathbb{E}_{q(x,y)}\mathbb{E}_{q(z|x)}\left[\log p(y|x, z)\right] \tag{10}$$

$$\approx \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{q(z|x_i)}\left[\log p(y_i|x_i, z)\right] \tag{11}$$

$$L_G = \mathrm{KL}(q(x, z)||p(x, z)) \tag{12}$$

$$\approx \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{q(z|x_i)}\left[\log p(x_i, z) - \log q(z|x_i)\right], \tag{13}$$

and $\alpha = \gamma$, $\beta = 1$. The discriminative component is optimized using the log-loss, while the generative component is fit with the reverse KL divergence to $q(x, z)$.

Explicit density hybrid models are very general and encompass many techniques, including most common directed latent variable models, deep generative models, latent-variable Markov random fields with tractable gradients, etc. They can also be trained using recent advances in variational inference, such as stochastic gradient variational Bayes (Kingma & Welling, 2013).

3.1.2 Implicit Density Models

Alternatively, we may specify $p(x, z)$ through an implicit distribution, which only requires us to be able to take samples from a differentiable mechanism $p(x|z)$ operating over input samples from a simple prior $p(z)$. Generative adversarial networks (GANs) (Goodfellow et al., 2014) are perhaps the most well-known class of implicit models. This

approach will offer greater modeling flexibility and will potentially allow us to use a greater range of f-divergencies for L_G (Nowozin et al., 2016)

More concretely, we will focus on the following choice of loss functionals:

$$L_D = \mathbb{E}_{q(x,y)}\mathbb{E}_{q(z|x)}\left[\log p(y|x,z)\right] \tag{14}$$

$$\approx \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{q(z|x_i)}\left[\log p(y_i|x_i,z)\right] \tag{15}$$

$$L_G = \mathrm{JS}(q(x,z)||p(x,z)) \tag{16}$$

$$\approx \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{q(z|x_i)}\log D(x_i,z)+\mathbb{E}_{p(x,z)}\log(1-D(x,z)). \tag{17}$$

The discriminative functional L_D is based on the log loss, as in the previous section; the generative functional L_G is a GAN objective (Goodfellow et al., 2014). The GAN loss introduces an additional *discriminator* function $D(x) : \mathcal{X} \to [0, 1]$, whose goal is to discriminate two kinds of tuples: samples $(x, z) \sim p(x|z)p(z)$ from the implicit model $p(x, z)$ as well as real samples x together with their estimated latent $z \sim q(z|x)$. We train D to maximize L_G, whereas p, q are fit to minimize L_G. The resulting objective can be shown to approximate the Jensen-Shannon divergence $\mathrm{JS}(q(x, z)||p(x, z))$ in the limit as the discriminator tends towards optimality (Dumoulin et al., 2016). This objective can be extended to arbitrary f-divergences using standard techniques (Nowozin et al., 2016). Interestingly, it could be used to recover the earlier ELBO objective (Equation 9) within the GAN framework.

The advantage of the above objective over explicit models lies in its modeling flexibility: we may use any sampling mechanism $p(x|z)$. Moreover, in practice, GAN-based techniques tend to produce better-looking samples x and perform well on semi-supervised learning. We expect that implicit hybrid models will be preferred in these settings as well.

3.2 DEEP HYBRID MODELS

Instantiating the components $p(y|x, z)$, $p(x, z)$ using modern deep learning architectures gives rise to deep hybrid models (DHMs), a new family of deep learning algorithms. DHMs are highly flexible and interpolate between several existing models; in several settings they offer accuracy improvements, such as in semi-supervised learning. We examine several classes of DHMs below.

Explicit Density. Explicit density DHMs are naturally formed by combining a deep generative model for $p(x, z)$ with a feed-forward neural network for $p(y|x, z)$. In our experiments, we take $p(x, z)$ to be a auxiliary-variable deep generative model (Maaløe et al., 2016), which is a type of variational autoencoder (VAE; Kingma & Welling (2013)); the discriminative part $p(y|x, z)$ is a convolutional neural network (CNN).

Note that our training objective (10-13) trains each method with its typical objective: the CNN is trained with categorical cross-entropy, while the VAE is trained with the evidence lower-bound. Crucially, the two models are tied via the shared latent z, and hence are trained jointly. By assign each term a suitable weight, we may improve the performance of each individual model. The final joint $p(y|x, z)p(x, z)$ will interpolate betwen the two methods.

For example, while the generative model $p(z|x)$ learns a latent feature representation z for the input x, the discriminative component $p(y|x, z)$ uses these features to accurately predict the labels y. This form of multi-task learning may regularize both models, especially in low-data regimes.

Implicit Density. Implicit density models are inherently tied to neural networks, since they represent a the main way of optimizing implicit density objectives such as (14-17). Furthermore, Equation 17 is equivalent to the adversarially learned inference objective (ALI; Dumoulin et al. (2016)). We augment this construction with an extra discriminative model, in a way that is reminiscent of recent constructions used for semi-supervised learning (Salimans et al., 2016; Dumoulin et al., 2016). We expect implicit density models will inherit the advantages of current GAN-based models, such as visually pleasing samples and good semi-supervised performance.

4 EXPERIMENTS

Next, we study our framework empirically. We focus on deep hybrid models, and start with the fully labeled setting, before moving on to semi-supervised learning.

4.1 EXPERIMENTAL SETUP

4.1.1 Datasets

MNIST. The MNIST dataset consists of 60,000 labeled training images of digits of dimension 28×28 and 10,000 testing images.

SVHN. The Street View House Number (SVHN) dataset consists of 73,257 training examples of 32×32 real-world house number color images; the task is to classify the middle digit. We normalized the inputs in $[-1, 1]$.

CIFAR-10. The CIFAR-10 dataset consists of 50,000 training and 10,000 testing examples of 32×32 color images of different types of animals and vehicles (ten classes in total). Following Laine & Aila (2016), we normalized the input using a whitening transform.

4.1.2 Models

Explicit models. In our experiments with explicit models, we parametrized $p(y|x,z)$ using a neural network with three sets of convolutional layers with 128 filters of size 3×3, each followed by 2×2 max-pooling. We passed the output via a dense hidden layer of size 500, concatenated the result with z and fed the output into a softmax layer.

We parametrized the generative model $p(x,z)$ using an auxiliary-variable deep generative model (ADGM), a more expressive variant of the variational auto-encoder (Maaløe et al., 2016). In brief, an ADGM p has the form $p(a,x,z) = p(a|z)p(x|z)p(z)$, where a are auxiliary latent variables that we introduce into the approximating posterior $q(a,z) = q(z|a)q(a)$ to make it multimodal. We parametrize $q(z|a)$, $p(a|z)$, $p(x|z)$ using dense neural networks with a single hidden layer of size 500; the priors $q(a)$, $p(z)$ were unit normal random variables. We also set $\dim(a) = 10$ and $\dim(z) = 200$.

Implicit Models. In our experiments with implicit models, we used models based on the GAN architectures of Salimans et al. (2016) and Laine & Aila (2016), extending it to the adversarially learned inference (ALI) framework.

For the discriminator, we used two convolutional blocks of three convolutions, each respectively containing 128 and 256 3×3 filters. This was followed by two 1×1 convolutions and global average pooling. We used two dense layers of 256 and 128 filters for the discriminator of z. The two outputs were concatenated and fed to a last dense layer. We used the same architecture in the discriminative model $p(y|x,z)$, and following the approach of Salimans et al. (2016) and Dumoulin et al. (2016), we reuse $p(y|x,z)$ for the discriminator, by introducing an extra class for fake examples.

We use a generator $p(x|z)$ parametrized by a dense layer of size 8,912 followed by three deconvolutional layers with 256, 128, and 3 output filters. The generator $q(z|x)$ consisted of the same three layers in reverse order, followed by a 1×1 convolution and global average pooling.

4.2 INTERPOLATING BETWEEN GENERATIVE AND DISCRIMINATIVE

In this section, we will give examples of how combining both kinds of models improves performance. We start by looking at how a discriminative model benefits from being coupled with a generative component; then, we look at how generative models can use a discriminative signal.

4.2.1 Improvements to the Discriminative Model

To examine the effects of the hyperparameters α and β, we set $\beta = 1$ and varied $\alpha \in \{10^0, 10^1, ..., 10^5\}$ following Druck et al. (2007). We trained an explicit DHM on MNIST for 200 iterations with a learning rate of 0.0003 (decayed to 0 over the last 40 iterations), and evaluated the final test accuracy. We compared our method against a purely discriminative baseline ($\alpha = 1$, $\beta = 0$); note that the baseline uses the exact same number of parameters as our model (namely, both use $p(y|x,z)$ and $q(z|x)$). The results of this experiment are in Figure 2 (a).

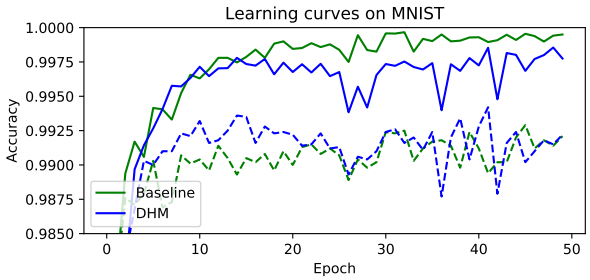

Figure 3: Learning curves of the hybrid (blue) and baseline (green) models on the training (solid line) and test sets (dashed line) for the first 50 epochs. The baseline reaches 0% training error much faster that the hybrid model.

Accuracy. The most obvious improvement over the baseline is an increase in discriminative accuracy. Adding a generative component reduces the test error from about 0.8% for the baseline to about 0.6-0.5%. Varying α does not significantly affect performance; in general, the method appears to be quite robust to this choice of hyperparameter, suggesting there is a lesser need for tuning.

To confirm our findings, we retrained the same network on the SVHN dataset; see Figure 2 (b). Test accuracy again improved from 92.12% to 92.73%, representing about a 0.5% gain in accuracy over same network trained in a purely discriminative fashion. We also noticed slight gains in performance from increasing α.

Regularization. The reason for this gain in accuracy appears to be due to a regularization effect. In Figure 3, we plot the learning curve of the baseline ($\alpha = 1$, $\beta = 0$) and the hybrid method with $\alpha = \beta = 1$. The training error for the baseline very quickly drops to zero; on the other hand, the training error of the hybrid model stays above zero during most of training, and especially at the beginning. Most importantly, the test error of the hybrid method is lower than that of the baseline, and is closer to the training error, indicating that the hybrid model generalizes better.

Intuitively, the network $q(z|x)$ must learn a representation that is useful for both predicting y and reconstructing the data x. This sort of multi-task objective prevents it from significantly overfitting on the prediction task.

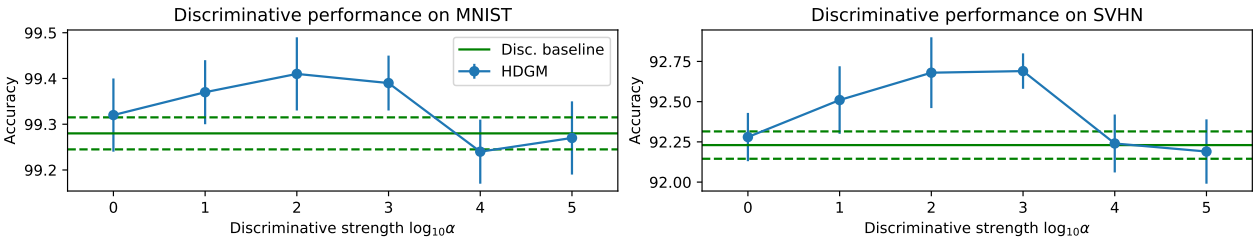

Figure 2: Hybrid model test accuracy for various discriminative weights α on the MNIST (left) and SVHN datasets (right).

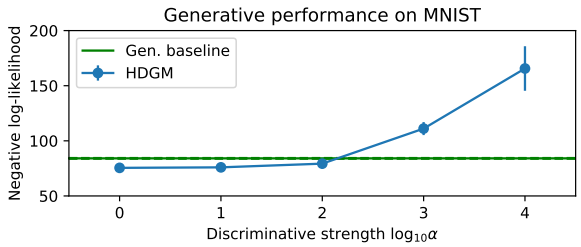

Figure 4: Generative log-likelihood of the hybrid and baseline models as a function of the strength of the discriminative component α on the MNIST dataset.

4.2.2 Improvements to the Generative Model

Conversely, adding a discriminative component to the generative model improves its ability to represent the data and learn useful features.

Generative Log-Likelihood. We evaluated the marginal log-likelihood of the data under the generative component $p(x)$ for each of the runs from the MNIST experiment in the previous section. Figure 4 shows the variational lower bound (ELBO) on the log-likelihood for $\alpha \in \{10^0, 10^1, ..., 10^5\}$, as well as a purely generative baseline with $\alpha = 0, \beta = 1$. The log-likelihood of the hybrid model improves over the baseline and varies with α. Our highest value for the ELBO is -75.5, which is, to the best of our knowledge, higher than any previous purely generative result on MNIST.

This experiment demonstrates that our way of coupling the two components via latent variables allows the discriminative signal to flow from the labels to the generative model, even though it does not share parameters with the discriminative component.

Feature Representation. Next, we trained a linear classifier to predict y from the latent representation learned by our hybrid model with $\alpha = \beta = 1$ and the purely unsupervised baseline ($\alpha = 0, \beta = 1$). In both cases, we used the representation specified by $q(z|x)$. We observed an improvement in accuracy from 97.5% to 98.4% when using the hybrid model. This again indicates that our latent-variable coupling method is effective at propagating signal between the two components of the model.

4.3 SEMI-SUPERVISED LEARNING

In the semi-supervised setting, DHMs can jointly benefit from both discriminative and generative semi-supervised algorithms. We may combine $p(y|x, z)$ with a regularizer (e.g. entropy regularization) while also learning the manifold of unlabeled data using $p(x, z)$. Interpolating between the two models may offer accuracy improvements.

Experimental Setup. We keep the labels of a small random fraction of training examples, and treat the remaining data as unlabeled. Following previous work, we choose 100 labeled points for MNIST, 1000 for SVHN, and 4000 for CIFAR-10. We report results averaged over 10 runs, with error bars corresponding to $\pm 2\sigma$.

Model. We use implicit density DHMs whose architecture we described earlier. We train $p(y|x, z)$ with Π-model discriminative regularization of Laine & Aila (2016), a discriminative semi-supervised technique that adds a penalty $||p(y|x_i) - p'(y|x_i)||_2$ on the difference between two successive applications p, p' of the same model to a given data point x_i (where $p(y|x_i)$ denotes the probability vector of the K classes). The difference between p and p' originates from the internal stochasticity of the model, e.g. dropout regularization, Monte-Carlo sampling, etc.

We use a similar set of hyper-parameters across the three datasets, with $\alpha = \beta = 1$, and the same Π-model hyperparameters as proposed by Laine & Aila (2016).

Results. Our results are summarized in Table 1. We achieve close to state-of-the-art performance on the three datasets. We slightly improve the previous error rate rate on MNIST, while on SVHN we achieve a full 1% in error reduction on the test set. On CIFAR-10, we observe performance comparable to previous state-of-the-art generative models, which is slightly worse than a fully-discriminative approach based on temporal ensembling (Laine & Aila, 2016).

Method	Accuracy
VAE (Kingma et al., 2014)	$3.33 \pm 0.14\%$
SDGM (Maaløe et al., 2016)	$1.91 \pm 0.10\%$
Ladder Network (Rasmus et al.)	$1.06 \pm 0.37\%$
ADGM (Maaløe et al., 2016)	$0.96 \pm 0.02\%$
Improved GAN (Salimans et al., 2016)	$0.93 \pm 0.07\%$
Implicit DHM (ours)	$0.91 \pm 0.06\%$

Table 1: Semi-supervised error on MNIST (100 labels).

Method	Accuracy
VAE (Kingma et al., 2014)	$36.02 \pm 0.10\%$
SDGM (Maaløe et al., 2016)	$16.61 \pm 0.24\%$
Improved GAN (Maaløe et al., 2016)	$8.11 \pm 1.3\%$
ALI (Dumoulin et al., 2016)	$7.42 \pm 0.65\%$
Π-model (Laine & Aila, 2016)	$5.45 \pm 0.25\%$
Implicit DHM (ours)	$4.45 \pm 0.35\%$

Table 2: Semi-supervised error on SVHN (1000 labels).

Method	Accuracy
Ladder Network (Rasmus et al.)	$20.40 \pm 0.47\%$
Improved GAN (Maaløe et al., 2016)	$18.63 \pm 2.32\%$
ALI (Dumoulin et al., 2016)	$17.99 \pm 1.62\%$
Π-model (Laine & Aila, 2016)	$16.55 \pm 0.29\%$
Implicit DHM (ours)	$19.34 \pm 1.05\%$

Table 3: Semi-supervised error on CIFAR-10 (4000 labels).

5 DISCUSSION

Latent-Variable Coupling. Central to our work is a new framework for interpolating between discriminative and generative models. Whereas previous hybrid models were based on parameter sharing (McCallum et al., 2006; Lasserre et al., 2006), our framework combines any two models, as long as they can both incorporate the shared latent variables z. Thus, $p(y|x,z)$ and $p(x,z)$ can be both arbitrarily complex and still be tractable to optimize.

This increased flexibility enables us to define deep hybrid models (DHMs), an instantiation of our framework with modern deep learning architectures.

Multi-Task Regularization. Deep hybrid models jointly train two deep learning models (e.g. a variational autoencoder and a convolutional neural network) and combine some of their advantages. In the end, we obtain a discriminative and a generative model, as well as the joint model defined by the combination of the two. Crucially, these are all trained jointly with a new multi-conditional objective that can improve over standard procedures.

Our approach can be viewed as multi-task regularization, where we train the model to use latent variables z that are useful for both predicting labels y and reconstructing inputs x. This effect is strongest when a large fraction of the data is unlabeled as well as when the architecture of $p(y|x,z)$ relies strongly on z.

Semi-Supervised Learning. Since DHMs involve training both a discriminative and a generative component, we may naturally use both discriminative and generative semi-supervised setting approaches jointly on the same hybrid model. For example, we may use an entropy regularizer on the discriminative predictor while also learning the data manifold with the generative model.

By interpolating between discriminative and generative semi-supervised learning algorithms (by assigning them weights α, β), we may improve classification accuracy. Our framework is also sufficiently general to be combined with most semi-supervised learning algorithms, including improved algorithms that will be proposed in the future.

Alternative Formulations. A different approach to trade-off discriminative and generative modeling power is to assign weights to $p(y|x)$ and $p(x)$ after the latent z variables have been marginalized out, as in the following objective:

$$\alpha \log \int_z p(y,z|x) + \beta \log \int_z p(z,x). \tag{18}$$

Both components can be optimized using variational inference; however the variational lower bound for the first term requires us to efficiently compute $p(y,z|x) = p(y|z,x)p(z|x)$. Since in most modern generative models, the posterior $p(z|x)$ is intractable, it is not immediately obvious how to apply this formulation to such models.

Shu et al. (2016) recently proposed a different formulation of hybrid models. It centers on parameter sharing, and hence is less general than our framework. Our approach also naturally extends to implicit models. Interestingly, their results suggest that choosing $p(y|x,z)$ to be independent of x may be preferable in the context of deep hybrid models.

6 CONCLUSION

In this work, we have proposed a framework for training a broad class of hybrid discriminative-generative models that combines the strengths of the two modeling approaches using latent variable coupling. Our framework offers greater modeling flexibility relative to previous methods and is compatible with modern deep learning architectures and complex latent-variable models. It enables us to apply discriminative and generative semi-supervised learning algorithms on the same model, which results in accuracy improvements over the state-of-the-art on the SVHN dataset. Our modeling ideas are sufficiently general to be combined with most semi-supervised learning algorithms, including improved algorithms that will be proposed in the future.

References

Amodei, Dario, Anubhai, Rishita, Battenberg, Eric, Case, Carl, Casper, Jared, Catanzaro, Bryan C., Chen, Jingdong, Chrzanowski, Mike, Coates, Adam, Diamos, Greg, Elsen, Erich, Engel, Jesse, Fan, Linxi, Fougner, Christopher, Han, Tony, Hannun, Awni Y., Jun, Billy, LeGresley, Patrick, Lin, Libby, Narang, Sharan, Ng, Andrew Y., Ozair, Sherjil, Prenger, Ryan, Raiman, Jonathan, Satheesh, Sanjeev, Seetapun, David, Sengupta, Shubho, Wang, Yi, Wang, Zhiqian, Wang, Chong, Xiao, Bo, Yogatama, Dani, Zhan, Jun, and Zhu, Zhenyao. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595, 2015.

Chapelle, Olivier, Schlkopf, Bernhard, and Zien, Alexander. *Semi-Supervised Learning*. 2006.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Druck, Gregory, Pal, Chris, Zhu, Xiaojin, and McCallum, Andrew. Semi-supervised classification with hybrid generative/discriminative methods. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*, number 0, pp. 280–289, 2007. Submitted to KDD 2007.

Dumoulin, Vincent, Belghazi, Ishmael, Poole, Ben, Lamb, Alex, Arjovsky, Martin, Mastropietro, Olivier, and Courville, Aaron. Adversarially learned inference. *CoRR*, abs/1606.00704, 2016.

Ghahramani, Zoubin, Welling, Max, Cortes, Corinna, Lawrence, Neil D., and Weinberger, Kilian Q. (eds.). *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014.

Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets. In Ghahramani et al. (2014), pp. 2672–2680.

Grandvalet, Yves and Bengio, Yoshua. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pp. 529–536, 2004.

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

Kingma, Diederik P., Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In Ghahramani et al. (2014), pp. 3581–3589.

Laine, Samuli and Aila, Timo. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016.

Lasserre, Julia A., Bishop, Christopher M., and Minka, Thomas P. Principled hybrids of generative and discriminative models. pp. 87–94, 2006. doi: 10.1109/CVPR.2006.227.

Maaløe, Lars, Sønderby, Casper Kaae, Sønderby, Søren Kaae, and Winther, Ole. Auxiliary deep generative models. In Balcan, Maria-Florina and Weinberger, Kilian Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1445–1453. JMLR.org, 2016.

McCallum, Andrew, Pal, Chris, Druck, Greg, and Wang, Xuerui. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *Proceedings of AAAI '06: American Association for Artificial Intelligence National Conference on Artificial Intelligence*, pp. 433–439, 2006.

Ng, Andrew Y and Jordan, Michael I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 2:841–848, 2002.

Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. In Lee, Daniel D., Sugiyama, Masashi, von Luxburg, Ulrike, Guyon, Isabelle, and Garnett, Roman (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 271–279, 2016.

Rasmus, Antti, Berglund, Mathias, Honkala, Mikko, Valpola, Harri, and Raiko, Tapani. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 3546–3554.

Salimans, Tim, Goodfellow, Ian J., Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2226–2234, 2016.

Shu, Rui, Bui, Hung H, and Ghavamzadeh, Mohammad. Bottleneck conditional density estimation. *arXiv preprint arXiv:1611.08568*, 2016.

Wu, Yonghui, Schuster, Mike, Chen, Zhifeng, Le, Quoc V., Norouzi, Mohammad, Macherey, Wolfgang, Krikun, Maxim, Cao, Yuan, Gao, Qin, Macherey, Klaus,

Klingner, Jeff, Shah, Apurva, Johnson, Melvin, Liu, Xiaobing, ukasz Kaiser, Gouws, Stephan, Kato, Yoshikiyo, Kudo, Taku, Kazawa, Hideto, Stevens, Keith, Kurian, George, Patil, Nishant, Wang, Wei, Young, Cliff, Smith, Jason, Riesa, Jason, Rudnick, Alex, Vinyals, Oriol, Corrado, Greg, Hughes, Macduff, and Dean, Jeffrey. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.