A PROBABILISTIC MODEL FOR DISCRIMINATIVE AND NEURO-SYMBOLIC SEMI-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Strong progress has been achieved in semi-supervised learning (SSL) by combining several methods, some of which relate to properties of the data distribution $p(x)$, others to the model outputs $p(y|x)$, e.g. minimising the entropy of unlabelled predictions. Focusing on the latter, we fill a gap in the standard text by introducing a probabilistic model for *discriminative* semi-supervised learning, mirroring the classical generative model. Several SSL methods are theoretically explained by our model as inducing (approximate) strong priors over parameters of $p(y|x)$. Applying this same probabilistic model to tasks in which labels represent binary attributes, we theoretically justify a family of *neuro-symbolic* SSL approaches, taking a step towards bridging the divide between statistical learning and logical reasoning.

1 INTRODUCTION

In semi-supervised learning (SSL), a mapping is learned from data points x to labels y given labelled pairs (x^l, y^l) and unlabelled x^u. Such tasks are of practical importance since unlabelled data are often cheaper to acquire and/or more abundant than labelled data. For unlabelled data to help predict labels, the distribution of x must contain information relevant to the prediction (Chapelle et al., 2009; Zhu & Goldberg, 2009). State-of-the-art SSL algorithms (e.g. Berthelot et al., 2019b;a) combine multiple methods, many of which can be seen to depend on properties of $p(x)$, although some rely on properties of the model outputs $p(y|x)$. Methods of the latter type include entropy minimisation (Grandvalet & Bengio, 2005), mutual exclusivity (Sajjadi et al., 2016a; Xu et al., 2018) and pseudo-labelling (Lee, 2013), each of which appends a function of the predictions on unlabelled data to a typical discriminative supervised loss function. We propose a probabilistic model that supplements existing theory to theoretically justify such *discriminative* semi-supervised learning (DSSL) methods.

Neuro-symbolic learning is a broad field aiming to combine statistical learning and logical reasoning. Approaches often introduce probability into a logical framework (De Raedt et al., 2007; Manhaeve et al., 2018), or logic into statistical learning models, e.g. neural networks (Rocktäschel et al., 2015; van Krieken et al., 2019). Much progress has been made in integrating the two paradigms, often intersecting with semi-supervised learning (Xu et al., 2018; van Krieken et al., 2019), but a fully principled understanding eludes. We show that our probabilistic model for SSL extends to neuro-symbolic SSL, theoretically explaining methods that augment a supervised loss function with a function based on logical constraints, taking an important step towards bridging the 'connectionism/symbolism' divide.

Central to our common probabilistic model are parameters θ^x of the distribution $p(\mathrm{y}|x)$ (where y is a random variable over labels), as predicted by statistical models, e.g. neural networks with an appropriate loss function. Since each x induces a possibly unique label distribution with an associated θ^x, sampling from $p(x)$ implicitly defines a distribution $p(\theta)$ over ground truth parameters, and so over the predictions expected of an accurate model. Thus, if available, $p(\theta)$ provides a prior over parameters predicted for unlabelled data, against which such predictions and the prediction model itself can be evaluated and potentially improved. Further, in practice, $p(\theta)$ often avoids much of the complexity of $p(x)$ to have a concise analytical form known *a priori*. Alternatively, an empirical estimate of the distribution $p(\theta)$ may be learned from parameter estimates for labelled data, fitting the intuition that unlabelled data predictions should resemble those of labelled data. We formalise this within a hierarchical probabilistic model that explains *discriminative SSL*, which relies on properties of $p(\theta)$ and is orthogonal and complementary to SSL approaches relying on $p(x)$, e.g. *data augmentation* and *consistency regularisation* (Sajjadi et al., 2016b; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2018), as state-of-art results demonstrate (Berthelot et al., 2019b;a).

We consider the explicit form of $p(\theta)$ for classification tasks in which classes are mutually exclusive, i.e. each x only ever pairs with a particular y, hence $y|x$ is deterministic and $p(\theta)$, a distribution over the simplex, reduces to a *discrete* distribution over its vertices. Where labels indicate a single class, e.g. for MNIST, CIFAR and ImageNet image datasets, this simple observation theoretically explains and unifies SSL methods of entropy minimisation, mutual exclusivity and pseudo-labelling as approximating continuous relaxations of $p(\theta)$. Where labels are binary vectors, e.g. representing (co-occurring) image features, allowed chess configurations or relations between entities in a knowledge base, only certain labels (i.e. attribute combinations) may be *valid*, e.g. according to rules of the game or the laws of nature. Here, $p(\theta)$ is again a discrete distribution over vertices of the simplex, but restricted to vertices corresponding to *valid* labels. Where valid attribute combinations can be defined with logical rules, those rules can be translated into simple continuous functions that compound to give a continuous relaxation of $p(\theta)$. As such, under our probabilistic model for SSL, a prior over model predictions can be derived from, or interpreted in terms of, logical rules, theoretically justifying statistical learning models that regularise outputs using logical rules. Our key contributions are:

- to provide a probabilistic model for *discriminative* semi-supervised learning, comparable to that for classical generative methods, contributing to current theoretical understanding of SSL;
- to consider the analytical form of the distribution over parameters $p(\theta)$, by which we explain several SSL methods, including entropy minimisation as used in state-of-art SSL models; and
- to show that our probabilistic model also unifies neuro-symbolic approaches to SSL in which the distribution $p(\theta)$ over expected model outputs can be defined in terms of logical rules, providing a theoretically principled understanding of integrating 'connectionist' and 'symbolic' methods.

2 Background and related work

Notation: $x_i^l \in \boldsymbol{X}^l$, $y_i^l \in \boldsymbol{Y}^l$ are labelled data pairs, $i \in \{1...N_l\}$; $x_j^u \in \boldsymbol{X}^u$, $y_j^u \in \boldsymbol{Y}^u$ are unlabelled data samples and their (unknown) labels, $j \in \{1...N_u\}$; $\mathcal{X}, \mathcal{Y}$ are domains of x and y. θ^x parameterises the distribution $p(\mathrm{y}|x)$, and is a realisation of a random variable θ. That is, for each x, a parameter θ^x defines a distribution over associated y, and $p(\theta)$ is the distribution over such parameters.

2.1 Semi-supervised learning

Semi-supervised learning is a well established field, described by a number of surveys and taxonomies (Seeger, 2006; Zhu & Goldberg, 2009; Chapelle et al., 2009; Van Engelen & Hoos, 2020). SSL methods have been categorised by how they adapt supervised learning algorithms (Van Engelen & Hoos, 2020); or their assumptions (Chapelle et al., 2009), e.g. that data of each class form a cluster/manifold, or that data of different classes are separated by low density regions. It has been proposed that all such assumptions are variations of *clustering* (Van Engelen & Hoos, 2020). Whilst 'clustering' itself is not well defined (Estivill-Castro, 2002), from a probabilistic perspective this suggests that SSL methods assume $p(x)$ to be a *mixture* of conditional distributions that are distinguishable by some property, e.g. connected dense regions. This satisfies the condition that for unlabelled x_u to help in learning to predict y from x, the distribution of x must contain information relevant to the prediction (Chapelle et al., 2009; Zhu & Goldberg, 2009). In this work, we distinguish SSL methods by whether they rely on direct properties of $p(x)$ or on properties that manifest in $p(\theta)$, the distribution over parameters of $p(\mathrm{y}|x;\theta^x)$, for $x \sim p(x)$. State-of-art models (Berthelot et al., 2019b;a) combine methods of both types.

A canonical SSL method that relies on explicit assumptions of $p(x)$ is the classical generative model:

$$p(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{X}^u) = \int_{\psi,\pi} p(\psi,\pi) p(\boldsymbol{X}^l|\boldsymbol{Y}^l,\psi) p(\boldsymbol{Y}^l|\pi) \underbrace{\sum_{\boldsymbol{Y}^u \in \mathcal{Y}^{N_u}} p(\boldsymbol{X}^u|\boldsymbol{Y}^u,\psi) p(\boldsymbol{Y}^u|\pi)}_{p(\boldsymbol{X}^u|\psi,\pi)} \quad (1)$$

Parameters ψ, π of $p(x|y)$ and $p(y)$ are learned from labelled and unlabelled data, e.g. by the EM algorithm, and predictions $p(y|x) = p(x|y)p(y)/p(x)$ follow by Bayes' rule. Figure 1 *(left)* shows the corresponding graphical model. Whilst generative SSL has an appealing probabilistic rationale, it is rarely used in practice, similarly to its counterpart for fully supervised learning, in large part because $p(x)$ is often complex yet must be accurately described (Grandvalet & Bengio, 2005; Zhu & Goldberg, 2009; Lawrence & Jordan, 2006). Properties of $p(x)$ also underpin *data augmentation* and *consistency regularisation* (Sajjadi et al., 2016b; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2018), in which true x samples are adjusted, using implicit domain knowledge of $p(x|y)$, to generate artificial samples of the same class, whether or not that class is known. Some

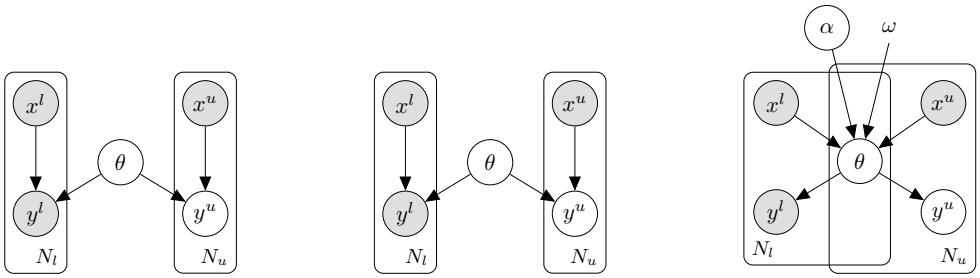

Figure 1: Graphical models for: generative SSL *(left)*; discriminative SSL (previous (Chapelle et al., 2009)) *(centre)*; discriminative SSL (ours) *(right)*. Shading indicates *observed* variables, which are otherwise *latent*.

SSL methods consider $p(x)$ in terms of components $p(x|z)$ for a latent representation z that helps to predict y (Kingma et al., 2014; Rasmus et al., 2015). We focus on SSL methods that supplement a discriminative supervised loss function with a function of the unlabelled data predictions, implicitly relying on properties of the distribution over parameters θ^x, e.g.:

Entropy minimisation (Grandvalet & Bengio, 2005) assumes classes are 'well separated'. As a proxy for class overlap, the entropy of predictions for unlabelled data is added to a discriminative supervised loss function ℓ^{sup}.

$$\ell_{\text{MinEnt}}(\theta) \;=\; \underbrace{-\sum_i \sum_k y^l_{i,k} \log \theta_k^{x^l_i}}_{\ell^{sup}} - \sum_j \sum_k \theta_k^{x^u_j} \log \theta_k^{x^u_j} \tag{2}$$

Mutual exclusivity (Sajjadi et al., 2016a; Xu et al., 2018) assumes no class overlap, i.e. correct predictions form 'one-hot' vectors. Viewed as vectors of logical variables $\boldsymbol{z}$, such outputs exclusively satisfy the logical formula $\bigvee_k (\boldsymbol{z}_k \bigwedge_{j \neq k} \neg \boldsymbol{z}_j)$. A function based on this formula is applied to unlabelled predictions:

$$\ell_{\text{MutExc}}(\theta) \;=\; \ell^{sup} - \sum_j \log \sum_k \theta_k^{x^u_j} \prod_{k' \neq k} (1 - \theta_{k'}^{x^u_j}) \tag{3}$$

Pseudo-labelling (Lee, 2013) assumes that predicted classes $k_j(t) = \arg\max_k \theta_k^{x^u_j}$ for unlabelled data x^u_j at iteration t, are correct (at the time) and treated as labelled data:

$$\ell_{\text{Pseudo}}(\theta, t) \;=\; \ell^{sup} - \sum_j \log \sum_k \mathbb{1}_{k=k_j(t)} \theta_k^{x^u_j} \tag{4}$$

Whilst intuitive, these methods lack a theoretical rationale comparable to that of generative models (Eq. 1). Summing over all labels for unlabelled samples is of little use (Lawrence & Jordan, 2006):

$$p(\boldsymbol{Y}^l|\boldsymbol{X}^l, \boldsymbol{X}^u) = \int_\theta p(\theta) p(\boldsymbol{Y}^l|\boldsymbol{X}^l, \theta) \underbrace{\textstyle\sum_{\boldsymbol{Y}^u} p(\boldsymbol{Y}^u|\boldsymbol{X}^u, \theta)}_{=1} = \int_\theta p(\theta) p(\boldsymbol{Y}^l|\boldsymbol{X}^l, \theta). \tag{5}$$

Indeed, under the previously considered graphical model (Fig. 1, *centre*), parameters θ of $p(\boldsymbol{Y}^l|\boldsymbol{X}^l, \theta)$ are provably independent of $\boldsymbol{X}^u$ (Seeger, 2006; Chapelle et al., 2009). Approaches to breaking the independence include introducing additional variables to Gaussian Processes (Lawrence & Jordan, 2006), or an assumption that parameters of $p(y|x)$ are dependent on those of $p(x)$ (Seeger, 2006). Taking further the (general) assumption of (Seeger, 2006), we provide a probabilistic model for *discriminative* SSL (DSSL), complementing that for generative SSL (Eq. 1).

2.2 Neuro-symbolic learning

Neuro-symbolic learning (NSL) is a broad field with the ambitious aim of bringing together statistical machine learning and logical reasoning (see (Garcez et al., 2019) for a summary). Many approaches either introduce probabilities into a logical framework (De Raedt et al., 2007; Manhaeve et al., 2018; Rocktäschel & Riedel, 2017); or combine logical rules into statistical learning methods such as neural networks (Rocktäschel et al., 2015; Ding et al., 2018; Marra et al., 2019; van Krieken et al., 2019; Wang et al., 2019). A general framework for NSL (Valiant, 2000; Garcez et al., 2019) places statistical methods within a low-level *perceptual* component that processes raw data (e.g. performing pattern recognition), the output of which feeds a *reasoning* module, e.g. performing logical inference (Fig. 2). This structure surfaces in various works (e.g. Marra et al., 2019; Wang et al., 2019; van Krieken et al.,

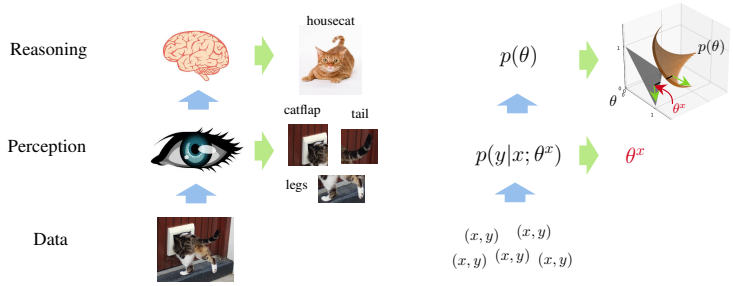

Figure 2: A general framework for neuro-symbolic learning combining statistical learning (perception) and logical rules (reasoning) (Valiant, 2000; Garcez et al., 2019). We draw an analogy to our probabilistic model for discriminative SSL (§3), in which $p(\theta)$ can be defined with logical rules (§5).

2019; Dai et al., 2019), in some cases taking explicit analytical form. Marra et al. (2019) propose a 2-step graphical model (their Fig. 1) comprising a neural network and a "semantic layer", however logical constraints are later introduced as a design choice (their Eq. 2), whereas in our work they are a natural way to parameterise a probability distribution. The graphical model for SSL of van Krieken et al. (2019) (their Fig. 1) includes a neural network component and a logic-based prior. However, knowledge base rules directly influence labels (y) of unlabelled data (only), whereas in our model, rules define a prior over parameters (θ^x) of distributions $p(\mathrm{y}|x), \forall x$. Where (van Krieken et al., 2019) views probabilities as 'continuous relaxations' of logical rules, we show such rules can be used to define the 'structure' of the prior $p(\theta)$ in our hierarchical probabilistic model for DSSL, which extends to provide a theoretical basis for neuro-symbolic semi-supervised learning.

3 A PROBABILISTIC MODEL FOR DISCRIMINATIVE SSL

We consider the labels $y \in \mathcal{Y}$ that occur with a given x as samples from $p(y|x;\theta^x)$, a distribution over the label space with some parameter θ^x, e.g. a multinomial distribution over classes, or a distribution over real values. We assume that every $x \in \mathcal{X}$ corresponds to a single *ground truth* θ^x in some domain Θ, thus there exists a well defined (deterministic) function $f: \mathcal{X} \to \Theta$, $f(x) = \theta^x$. Various predictive models learn to approximate f, e.g. neural networks with an appropriate loss function, taking in x and outputting $\tilde{\theta}^x$, an estimate of θ^x that maximises $p(y|x;\tilde{\theta}^x)$. Note that, given the 1-1 correspondence, sampling x from $p(x)$ induces a specific distribution $p(\theta)$ over corresponding parameters θ^x, and so over the outputs expected from a prediction model. In particular, $p(\theta)$ provides a prior over unlabelled predictions $\tilde{\theta}^u$, potentially enabling the prediction model to be improved, e.g. by back-propagating a gradient to increase $p(\tilde{\theta}^u)$. We will show (Sec. 4) that under certain assumptions the analytical form of $p(\theta)$ is known *a priori*, alternatively the distribution $p(\tilde{\theta}^{x^l})$ over labelled predictions provides an empirical approximation to $p(\theta)$. To formalise, let: $\theta^{\boldsymbol{X}^l} = \{\theta^x\}_{x \in \boldsymbol{X}^l}$ be the set of parameters of $p(\mathrm{y}|x)$ for each $x \in \boldsymbol{X}^l$ (with $\theta^{\boldsymbol{X}^u}$ defined similarly). For example, in K-class classification $p(\mathrm{y}|x)$ is a categorical distribution with parameter vector θ on the simplex $\Delta^K \subseteq \mathbb{R}^K$, where $\theta^x_k = p(y=k|x)$, $k \in \{1 \dots K\}$; and $p(\theta)$ is a distribution over Δ^K. Treating each θ^x as a latent random variable, the conditional distribution of the data factorises comparably to Eq. 5:

$$p(\boldsymbol{Y}^l|\boldsymbol{X}^l,\boldsymbol{X}^u) = \int_{\alpha,\theta^{\boldsymbol{X}},\tilde{\theta}^{\boldsymbol{X}}} p(\alpha)p(\boldsymbol{Y}^l|\tilde{\theta}^{\boldsymbol{X}^l})p(\tilde{\theta}^{\boldsymbol{X}^l}|\theta^{\boldsymbol{X}^l})p(\theta^{\boldsymbol{X}^l}|\alpha)\underbrace{\textstyle\sum_{\boldsymbol{Y}^u} p(\boldsymbol{Y}^u|\tilde{\theta}^{\boldsymbol{X}^u})}_{=1} p(\tilde{\theta}^{\boldsymbol{X}^u}|\theta^{\boldsymbol{X}^u})p(\theta^{\boldsymbol{X}^u}|\alpha)$$

$$\approx \int_{\alpha,\tilde{\theta}^{\boldsymbol{X}^u}} p(\alpha)\, p(\boldsymbol{Y}^l|\tilde{\theta}^{\boldsymbol{X}^l})p(\tilde{\theta}^{\boldsymbol{X}^l}|\alpha)\, p(\tilde{\theta}^{\boldsymbol{X}^u}|\alpha) \tag{6}$$

Auxiliary variables $\tilde{\theta}^x \doteq f_\omega(x)$ are introduced as estimates of ground truth θ^x, for $f_\omega: \mathcal{X} \to \Theta$, a family of functions parameterised by ω, e.g. a neural network. We replace $p(\boldsymbol{Y}|\boldsymbol{X},\theta^{\boldsymbol{X}})$ by $p(\boldsymbol{Y}|\theta^{\boldsymbol{X}})$ as θ^x fully defines $p(y|x)$. The distribution $p(\tilde{\theta}^x|\theta^x)$, over predictions given ground truth parameters, reflects model accuracy and is expected to vary over $x \in \mathcal{X}$. For *labelled data*, on which the model is trained, we assume (in row 2) that $\tilde{\theta}^x$ well approximates θ^x, s.t. $p(\tilde{\theta}^x|\,\theta^x) \approx \delta_{\theta^x - \tilde{\theta}^x}$. For *unlabelled data*, $p(\tilde{\theta}^x|\,\theta^x)$ is unknown but anticipated to be higher (on average) for predictions 'close' to the true parameter. $p(\tilde{\theta}^x|\,\alpha) = \int_{\theta^x} p(\tilde{\theta}^x|\,\theta^x)p(\theta^x|\alpha)$ can be interpreted as a *relaxation* of the prior applied to predictions (a perspective we take going forwards), that matches the prior in the error-free case, $p(\tilde{\theta}^x|\,\theta^x) \approx \delta_{\theta^x - \tilde{\theta}^x}$. Fig. 1 (*right*) shows the corresponding graphical model. Taken together, the relationship $f_\omega(x) = \tilde{\theta}^x$, the prior $p(\theta|\alpha)$ and the assumed closeness between θ^x and $\tilde{\theta}^x$, break the

independence noted previously (Sec. 2). Without f_ω, a sample x^u reveals nothing of $p(y|x^u;\theta^{x^u})$; without $p(\tilde{\theta}^x|\alpha)$, predictions can be made but not evaluated or improved. Interpreting terms of Eq. 6:

- $p(\boldsymbol{Y}^l|\,\tilde{\theta}^{\boldsymbol{X}^l})$ encourages labelled predictions $\tilde{\theta}^{x^l}$ to approximate parameters of $p(y|x^l)$;
- $p(\tilde{\theta}^{\boldsymbol{X}^l}|\,\alpha)$ encourages α to capture the distribution over $\tilde{\theta}^{x^l}$, parameters predicted for x^l; and
- $p(\tilde{\theta}^{\boldsymbol{X}^u}|\,\alpha)$ allows predictions $\tilde{\theta}^{x^u}$ on unlabelled data to be evaluated under prior knowledge of $p(\theta)$ and/or the approximate posterior distribution $p(\tilde{\theta}^{\boldsymbol{X}^l}|\alpha)$ learned from labelled data.

Maximum *a posteriori* estimates of θ^x are learned by optimising Eq. 6, e.g. in K-class classification (classes indexed by k) by minimising the following objective with respect to ω (recall $\tilde{\theta}^x \doteq f_\omega(x)$):

$$\ell_{DSSL}(\theta) \;= -\sum_i \sum_k y^l_{i,k} \log \tilde{\theta}^{x^l_i}_k - \sum_i \log p(\tilde{\theta}^{x^l_i}|\alpha) - \sum_j \log p(\tilde{\theta}^{x^u_j}|\alpha) \tag{7}$$

The distribution $p(\theta|\alpha)$ effectively regularises a supervised learning model, but, in contrast to common techniques (e.g. ℓ_1, ℓ_2), regularisation here is over model *outputs* $\tilde{\theta}$ not its weights ω. Importantly, $p(\theta)$ provides a relationship between data samples required for SSL, as an alternative to $p(x)$.

A natural question is: when to consider SSL methods that depend on $p(y|x)$ rather than $p(x)$, given that SSL has fewer y than x by definition. Importantly, the two options are not mutually exclusive, but rather orthogonal and can be combined, as in recent approaches (Berthelot et al., 2019b;a). Furthermore, the structure of $p(\theta)$ is often far simpler than that of $p(x)$ and may be known *a priori*, thus applying DSSL can be straightforward. We analyse the form of $p(\theta)$ in several cases in Sec. 4.

To relate discriminative and generative SSL, we note the inherent symmetric relationship between them. Rewriting the joint distributions underlying Eq.s 1 and 6 as:

$$p(\boldsymbol{Y}^l, \boldsymbol{X}^l, \boldsymbol{X}^u) = \int_{\pi,\psi^{\boldsymbol{Y}}} p(\pi) p(\boldsymbol{X}^l|\psi^{\boldsymbol{Y}^l}) p(\psi^{\boldsymbol{Y}^l}, \boldsymbol{Y}^l|\pi) \sum_{\boldsymbol{Y}^u} p(\boldsymbol{X}^u|\psi^{\boldsymbol{Y}^u}) p(\psi^{\boldsymbol{Y}^u}, \boldsymbol{Y}^u|\pi) \qquad \text{[Gen.]}$$

$$p(\boldsymbol{Y}^l, \boldsymbol{X}^l, \boldsymbol{X}^u) = \int_{\alpha,\theta^{\boldsymbol{X}}} p(\alpha) p(\boldsymbol{Y}^l|\,\theta^{\boldsymbol{X}^l}) p(\theta^{\boldsymbol{X}^l}, \boldsymbol{X}^l|\alpha) \sum_{\boldsymbol{Y}^u} p(\boldsymbol{Y}^u|\,\theta^{\boldsymbol{X}^u}) p(\theta^{\boldsymbol{X}^u}, \boldsymbol{X}^u|\alpha) \qquad \text{[Disc.]}$$

reflects a similar hierarchical structure in which one element of the data (y in the former, x in the latter) acts to 'label' or 'index' a distribution over the other (i.e. as a superscript to ψ or θ, resp.).

Lastly, we note that $f(x) = \theta^x$ (assumed differentiable) gives a relationship $p(\theta) = |\boldsymbol{J}|p(x)$, for $\boldsymbol{J}$ the Jacobian matrix $\boldsymbol{J}_{i,j} = \frac{\partial x_i}{\partial \theta_j}$. Thus, if $p(x) = \sum_k p(x|y=k)\pi_k$ is a mixture distribution with class probabilities $\pi_k = p(y=k)$, then $p(\theta)$ is also a mixture with identical class weights:

$$p(\theta) = |\boldsymbol{J}|p(x) = |\boldsymbol{J}| \sum_k p(x|y=k)\pi_k = \sum_k |\boldsymbol{J}|p(x|y=k)\pi_k = \sum_k p(\theta|y=k)\pi_k. \qquad \square$$

Thus any cluster/mixture assumption of $p(x)$ applies also to $p(\theta)$; and class conditional distributions $p(\theta|y)$ over ground truth parameters must differ sufficiently for classification to be possible.

4 Applications of discriminative semi-supervised learning

To make use of Eq. 6 requires a description of $p(\theta)$, ideally in analytic form. Such form depends on two key properties of the data/task: (i) the label domain $\mathcal{Y}$, e.g. being continuous or discrete; and (ii) $y|x$ being stochastic or deterministic. In the discrete case, labels may represent (i) one of K distinct classes (indexed k) by 'one-hot' vectors $\boldsymbol{y} \in \{e_k\}_k$, whereby each distribution $p(\mathbf{y}|x)$ is parameterised by some $\theta^x \in \Delta^K$ (the simplex), where $\theta^x_k = p(y=k|x)$; or (ii) K binary (non-exclusive) features with $\boldsymbol{y} \in \{0,1\}^K$ (combinations of which are distinct), whereby $\theta^x \in \Delta^{2^K}$ and $\theta^x_k = p(\boldsymbol{y}_k = 1|x)$. Table 1 shows examples of commonly occurring cases, several of which we discuss in detail.

Table 1: Task and data properties affecting the distribution $p(\theta)$ over parameters of $p(y|x;\theta)$.

Map $\boldsymbol{x} \to \boldsymbol{y}$ \ Domain $\mathcal{Y}$	Discrete (classification)		Continuous (regression)
	Distinct classes	**Non-exclusive features**	
Stochastic	Mix of Gaussians	-	-
Deterministic	MNIST, SVHN CIFAR, Imagenet	Animals w/attributes Sudoku completion Knowledge Base completion	Image-to-image translation

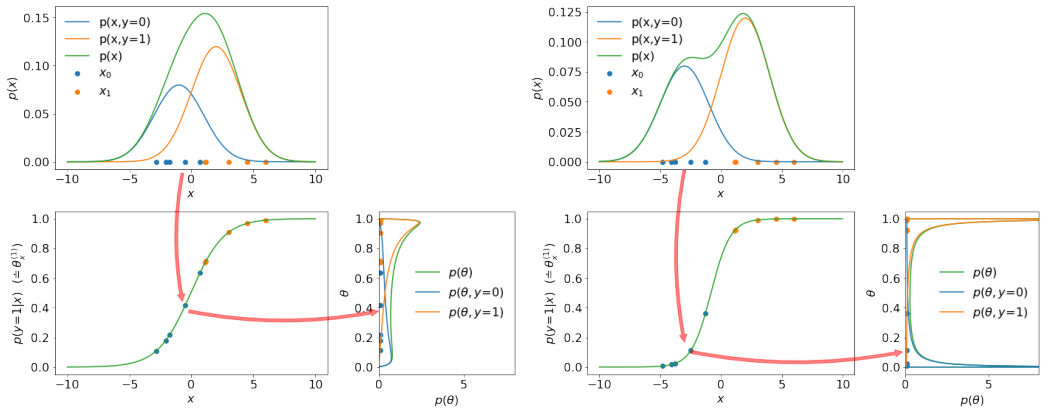

Figure 3: The distribution $p(\theta)$ for a mix of 2 univariate Gaussians (varying class separation).

Stochastic classification, distinct classes ($\boldsymbol{y}\in\{\boldsymbol{e}_k\}, \theta\in\Delta^K$): To exemplify how $p(x)$ and $p(\theta)$ can relate, we consider a mixture of two 1-dimensional equivariant Gaussians: $p(x)=\sum_k \pi_k p(x|y=k)$, $k\in\{0,1\}$, where $x|y=k \sim \mathcal{N}(\mu_k, \Sigma)$. Here, $p(\theta)$ can be derived in closed form (see Appendix):

$$p(\theta) = \sum_{k=0}^{1} \pi_k \underbrace{\sqrt{\frac{\sigma^2}{2\pi}} \frac{1}{|\mu_1-\mu_0|} \exp\{a(\log\tfrac{\theta_1}{\theta_0})^2 + (b_k-1)\log\theta_1 + (-b_k-1)\log\theta_0 + c_k\}}_{p(\theta|y=k)}$$

for coefficients a, b_k, c_k. As expected (Sec. 3), $p(\theta)$ is a mixture with the same class probabilities as $p(x)$. Fig. 3 shows plots of $p(x)$ and $p(\theta)$ varying the separation of class means μ_k, illustrating how the former determines the latter. Just as parameters of $p(x)$ would be revised if a set of new data samples x^u failed to fit, their predictions θ^x and $p(\theta)$ offer a similar opportunity. As the class means are separated, class overlap reduces and $p(\theta)$ tends towards a limiting *discrete* distribution. While here both $p(x)$ and $p(\theta)$ are analytically known, $p(x)$ is often complex and not known. However, subject to the data/task, $p(\theta)$ may yet be simply analytically defined.

Deterministic classification, distinct classes ($\boldsymbol{y}\in\{\boldsymbol{e}_k\}, \theta\in\{\boldsymbol{e}_k\}$): In many classification tasks, each x associates (materially) with only one class label y, i.e. no other label $y'\neq y$ appears with that x; thus classes are mutually exclusive and $y|x$ is *deterministic*. This applies to well-known image classification datasets such as MNIST, CIFAR and ImageNet. As such, $\theta_k^x = p(y=k|x)\in\{0,1\}$ and ground truth parameters, *as well as the labels*, form one-hot vectors $\theta\in\{\boldsymbol{e}_k\}_k\subset\Delta^K$. It follows that each class conditional $p(\theta|y=k)$ approximates a delta function $\delta_{\theta=\boldsymbol{e}_k}$ at a vertex of the simplex Δ^K:

$$p(\theta) \;=\; \sum\nolimits_k p(y=k)\, p(\theta|y=k) \;\approx\; \sum\nolimits_k \pi_k\, \delta_{\theta=\boldsymbol{e}_k}\,. \tag{8}$$

In words, this simply means that the true parameter θ^x for some x must be one-hot (identifying the single true label y), not a parameter allowing stochasticity (not one-hot). Thus, irrespective of complexity in $p(x)$, the structure of $p(\theta)$ is defined simply. This distribution, however, is discontinuous and lacks support over most of Θ, i.e. $p(\tilde{\theta})=0$ for any prediction $\tilde{\theta}\neq\boldsymbol{e}_k$, making it unsuitable for gradient-based learning methods. However, a continuous approximation to $p(\theta)$ is obtained by *relaxing* each delta component to a suitable function over Θ. Such relaxation can be interpreted as approximating a *noise* or *error* model $p(\tilde{\theta}^x|\theta^x)$ of predictions given true parameters (see Sec 3). From Eqs 2, 3 and 4, the unlabelled loss components of SSL methods entropy minimisation (Grandvalet & Bengio, 2005), mutual exclusivity (Sajjadi et al., 2016a; Xu et al., 2018) and pseudo-labelling (Lee, 2013) (plotted in Fig. 4) can be seen as (un-normalised) continuous relaxations $\hat{p}(\theta)$ of the discrete $p(\theta)$. (Note that, $\hat{p}(\theta)$ need not be normalised in practice since a weighting term in the loss function renders any proportionality constant obsolete.) We thus justify these methods theoretically under the discriminative probabilistic model for SSL (Eqs. 6, 7).

Deterministic classification, non-exclusive features ($\boldsymbol{y}\in\{0,1\}^K, \theta\in\{0,1\}^K$): In some classification tasks, label vectors $\boldsymbol{y}\in\{0,1\}^K$ represent multiple (K) binary attributes of x, e.g. features present in an image, a solution to Sudoku, or the relations connecting subject and object entities in a knowledge base. As in those examples, $\boldsymbol{y}|x$ may be deterministic. If so, for a given x_* and its (unique) label $\boldsymbol{y}_*$, the conditional distribution $p(\mathbf{y}|x_*)$ is given by the indicator function $\mathbb{1}_{\boldsymbol{y}-\boldsymbol{y}_*}$, with parameter $\theta^{x_*}=\boldsymbol{y}_*$. Thus, all parameters θ^x are restricted to vertices of the simplex $\{0,1\}^K\subset\Delta^{2^K}$ (analogous to one-hot vectors previously). It follows that $p(\theta|\boldsymbol{y})=\delta_{\theta-\boldsymbol{y}}$ and so $p(\theta)=\sum_{\mathcal{Y}}\pi_{\boldsymbol{y}}\,\delta_{\theta-\boldsymbol{y}}$

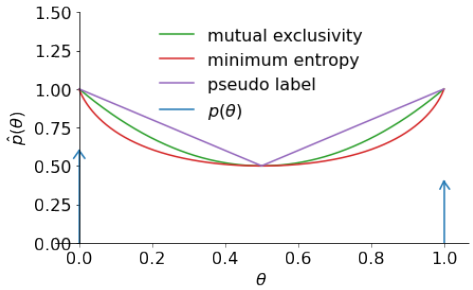

Figure 4: Unsupervised loss components of entropy minimisation (Eq. 2), mutual exclusivity (Eq. 3) and pseudo-labelling (Eq. 4) (exponentiated for comparison to probabilities), seen as continuous relaxations $\hat{p}(\theta)$ of the discrete distribution $p(\theta)$, for deterministic $y|x$ with distinct classes.

is a weighted sum of point probability masses at $\theta \in \{0,1\}^K$. A continuous relaxation of $p(\theta)$ again gives an approximation suitable for gradient based learning. The case becomes more interesting when we consider logical relationships that may exist between attributes (Sec. 5). Note that each $\theta \in \{0,1\}^K \subset \Delta^{2^K}$ (from a continuous space) maps, by identity $\theta = \boldsymbol{y}$, to a label $\boldsymbol{y} \in \{0,1\}^K$ (in a discrete space), thus there exists a one-to-one correspondence between θ in the support of $p(\theta)$ and labels $\boldsymbol{y}$. Consequently, the distribution $p(\theta)$, required for DSSL, can also be learned or improved from *unpaired labels* $\boldsymbol{y} \sim p(\boldsymbol{y})$, a variation of standard SSL (we leave this direction to future work).

5 Neuro-Symbolic Semi-Supervised Learning

In classification with deterministic non-exclusive binary features (Sec. 4), certain configurations of features may be impossible, e.g. an animal having both legs and fins, three kings on a chess board, or two knowledge base entities being related by *capital_city_of* but not *city_in*. Where so, the support of $p(\mathbf{y}|x)$ for any x is confined to a data-specific set of *valid* labels $\mathbb{V}$, a subset of all *plausible* labels $\mathbb{P} = \{0,1\}^K$, such that $p(\boldsymbol{y}|x) = 0, \forall \boldsymbol{y} \in \mathbb{P} \setminus \mathbb{V}$. Given the 1-1 correspondence between $\boldsymbol{y}, \theta \in \{0,1\}^K$ (§4), we let $\mathbb{V}$, $\mathbb{P}$ refer to both labels $\boldsymbol{y}$ and parameters θ that are valid or plausible (resp.). Thus:

$$p(\theta|\alpha) \;=\; \sum_{\boldsymbol{y} \in \mathbb{V}} p(\boldsymbol{y}) p(\theta|\boldsymbol{y}) \;=\; \sum_{\boldsymbol{y} \in \mathbb{V}} \pi_{\boldsymbol{y}} \delta_{\theta - \boldsymbol{y}}, \tag{9}$$

parameterised by $\alpha = \{\mathbb{V}, \Pi_{\mathbb{V}}\}$, where $\Pi_{\mathbb{V}} = \{\pi_{\boldsymbol{y}} = p(\boldsymbol{y})\}_{\boldsymbol{y} \in \mathbb{V}}$ are marginal label probabilities. (Note that Eq, 9 holds for any 'larger' set $\mathbb{V}' \subseteq \mathbb{P}$ containing $\mathbb{V}$.) As in the examples above, valid labels, and thus $\mathbb{V}$ itself, may be defined by the mutual exclusivity of attributes under the laws of nature, the rules of a game, or relationships between entity relations. Such 'universal' rules have certainty (*cf* the uncertain generalisation of statistical models, e.g. neural networks) and their wide reach may allow a large set $\mathbb{V}$ to be defined in relatively few rules between attributes. Loosely speaking, $\mathbb{V}$ defines the *structure* of $p(\theta)$, and $\Pi_{\mathbb{V}}$ its *magnitude*. Focusing on such structure, we drop $\pi_{\boldsymbol{y}}$ and consider probability mass (replacing $\delta_{\theta_k - c}$ with Kronecker delta $\delta_{\theta_k c}$), to define:

$$s(\theta) \;=\; \sum_{\boldsymbol{y} \in \mathbb{V}} \delta_{\theta \boldsymbol{y}} \;=\; \sum_{\boldsymbol{y} \in \mathbb{V}} \prod_{k:\boldsymbol{y}_k = 1} \delta_{\theta_k 1} \prod_{k:\boldsymbol{y}_k = 0} \delta_{\theta_k 0}\,, \tag{10}$$

which (sequentially) evaluates whether a given parameter θ matches a valid label $\boldsymbol{y} \in \mathbb{V}$: $s(\theta) = 1$ if $\theta \in \mathbb{V}$, else $s(\theta) = 0$. Restricted to plausible $\theta \in \mathbb{P}$ and defining logical variables $\boldsymbol{z}_k \iff (\delta_{\theta_k 1} = 1)$, the same test is expressible as a *logical formula* in propositional logic:

$$\bigvee_{\boldsymbol{y} \in \mathbb{V}} \bigwedge_{k:(\boldsymbol{y}_k = 1)} \boldsymbol{z}_k \bigwedge_{k:\neg(\boldsymbol{y}_k = 1)} \neg \boldsymbol{z}_k\,, \tag{11}$$

which evaluates to True $\iff \theta \in \mathbb{V} \iff s(\theta) = 1$. Comparing Eqs. 10 and 11 shows a relationship between logical and mathematical operations familiar in the fuzzy logic and neuro-symbolic literature (e.g. Bergmann, 2008; Serafini & Garcez, 2016; van Krieken et al., 2019). Here, True maps to 1, False to 0, $\wedge$ to multiplication, $\vee$ to addition, and where $\boldsymbol{z}_k$ corresponds to $\delta_{\theta_k 1} = 1$, a function of θ_k, $\neg \boldsymbol{z}_k$ maps to $\delta_{(1-\theta_k)1} = 1$, the function applied to $1 - \theta_k$. In fact, for any (m-ary) propositional logic operator $\circ(X_1 \ldots X_m)$, e.g. $X_1 \Rightarrow X_2$, various *functional representations* $\rho_\circ : [0,1]^m \to [0,1]$ exist, taking binary inputs corresponding to X_i being True/False, and outputting $\rho_\circ = 1$ if $\circ$ evaluates to True, else 0 (Bergmann, 2008; Marra et al., 2019). A logical formula can often be expressed in several equivalent ways, each of which corresponds to an equivalent functional representation (by compounding those of its operators, for which there may be several choices) that evaluates to 1 *iff* the logical formula is True, else 0. Thus, any set of logical rules that define $\mathbb{V}$, are equivalent to Eq. 11, and can be converted to a functional representation equivalent to $s(\theta)$ (restricted to $\theta \in \mathbb{P}$).

To generalise to all $\theta \in \Theta = \Delta^{2^K}$, the $\delta_{\theta_k 1}$ are relaxed to any continuous function $g(\theta_k) : \Theta \to [0, 1]$, $g(1) = 1$, $g(0) = 0$, leaving the resulting function $\hat{p}(\theta)$ unchanged for $\theta \in \mathbb{P}$. By choosing only continuous functional representations throughout, $\hat{p}(\theta)$ is itself continuous over Θ. As such $\hat{p} : \Theta \to [0, 1]$ satisfies $\hat{p}(\theta) = s(\theta) = 1$ for valid $\theta \in \mathbb{V}$, and $\hat{p}(\theta) = s(\theta) = 0$ for invalid $\theta \in \mathbb{P} \backslash \mathbb{V}$, serving as a continuous relaxation of $p(\theta)$, $\forall \theta \in \Theta$, up to probability weights $\pi_{\boldsymbol{y}}$ (see Sec. 6). This proves that the distribution $p(\theta)$ of Eqs. 6, 7, as required for discriminative SSL, can be approximated by a functional representation of logical rules. In practice (e.g. Serafini & Garcez, 2016; van Krieken et al., 2019; Marra et al., 2019), the simple assumption $g(\theta_k) = \theta_k$ is often taken from *fuzzy logic* (Bergmann, 2008). Applying it in Eq. 10 defines *semantic loss* (Xu et al., 2018), which is thus justified theoretically by our probabilistic model for discriminative SSL. The probabilistic model (Eqs. 6, 7) also allows logical rules to be *learned from data*, e.g. using logical techniques such as *abduction* (Wang et al., 2019; Dai et al., 2019) to learn a set of rules consistent with the observed labels and that ideally entail $\mathbb{V}$.

Having established a principled means of introducing logical rules into probabilistic SSL, we briefly consider ways to compute $\hat{p}(\theta)$ efficiently: (A) Substituting $g(\theta) = \theta$ in Eq. 10, as in (Xu et al., 2018), corresponds closely to *weighted model counting* (Sang et al., 2005; Chavira & Darwiche, 2008), which has efficient computational methods if $|\mathbb{V}|$ is not exponential in K, e.g. by multiplying θ by a binary matrix representing $\mathbb{V}$. (B) If all $\boldsymbol{y} \in \mathbb{V}$ satisfy a common fixed set of R logical rules $\mathbb{L} = \{l_r\}_r$, then Eq. 11 is equivalent to the conjunction $\bigwedge_r l_r$, and $\hat{p}(\theta)$ to a product of R terms. Thus, $\log \hat{p}(\theta)$ equates to a summation, and each rule l_r contributes a term independently added to the loss function. As such, a subset of $\mathbb{L}$ might be applied for efficiency, e.g. the most 'restrictive' or 'critical'. (C) Further, if each rule l_r involves up to T_r terms and $T_r R \ll |\mathbb{V}| K$, then $\hat{p}(\theta)$ (in product form) is far more efficient to compute than Eq. 10 (in sum form), which considers every valid label $\theta \in \mathbb{V}$. As an example, in classifying Sudoku solutions, R represents the number of rules, e.g. "only one 5 in row 1" as specified by a disjunction of $T_r = 9$ terms, whereas $|\mathbb{V}|$ is all possible Sudoku configurations.

Many works combine functional representations of logical formulae with statistical learning, but we introduce a principled explanation of where logical rules 'fit' within a probabilistic framework: as the structure of a prior distribution over valid parameters $p(\theta)$ in discriminative semi-supervised learning.

6 Discussion

In Section 5 we focus on the *structure* of a discrete distribution $p(\theta)$ defined by $\mathbb{V}$, rather than $\pi_{\boldsymbol{y}} \in \Pi_{\mathbb{V}}$, the probability mass assigned to each support of $p(\theta)$. Unless attributes occur independently, probabilities $\pi_{\boldsymbol{y}}$ do not factorise analogously to Eq. 10 and so cannot, in general, be manipulated as flexibly as the structural aspect. However, predictions $\tilde{\theta}^x$ are not random and $p(\tilde{\theta}^x | \theta^x)$, the probability of a prediction given the ground truth, is expected to fall as they diverge, as approximated by the loss term of DSSL methods (Table 4). Thus, whether omitting $\pi_{\boldsymbol{y}}$ has a significant effect depends on (i) prediction accuracy, (ii) the assumed noise model $p(\tilde{\theta}^x | \theta^x)$ and (iii) the class imbalance. Thus for reasonably balanced classes and an exponentially decaying noise model (e.g. minimum entropy), ignoring $\pi_{\boldsymbol{y}}$ may have little impact. This rationalises why label probabilities $\pi_{\boldsymbol{y}}$ are omitted in our analysis, and also the performance of discriminative SSL methods that omit them (Eqs 2, 3, 4).

7 Conclusion

We present a hierarchical probabilistic model for *discriminative* semi-supervised learning, complementing the classical generative model. Central to our model are parameters θ^x of the distribution $p(\mathrm{y}|x; \theta^x)$, as often predicted by neural networks. The distribution $p(\theta)$ over those parameters serves as a prior over outputs of a predictive model for unlabelled data. Subject to properties of the data, the analytical form of $p(\theta)$ may be known *a priori*. Alternatively, predictions on labelled data, or even unpaired labels, can provide an empirical estimate of $p(\theta)$. Where labels represent binary attributes, inter-attribute relationships may be expressed by logical rules. We show how such rules fit within the same probabilistic model, providing a principled means of combining logical reasoning and statistical learning. Such rules may be known *a priori* and imposed, or potentially learned from the data. Our model justifies several existing methods for semi-supervised learning and neuro-symbolic learning; and accords with a general architecture proposed for neuro-symbolic computation (Valiant, 2000; Garcez et al., 2019), comprising low level *perception* and high level *reasoning* modules (Fig 2). In future work, we plan to consider the stochastic $y|x$ case (i.e. combining aleatoric uncertainty), to make rigorous the concept of a *confidence* model (i.e. epistemic uncertainty), and to extend the principled combination of statistical learning and logical reasoning to supervised learning.

REFERENCES

Merrie Bergmann. *An introduction to many-valued and fuzzy logic: semantics, algebras, and derivation systems*. Cambridge University Press, 2008.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019b.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

Wang-Zhou Dai, Qiuling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. In *Advances in Neural Information Processing Systems*, 2019.

Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *International Joint Conferences on Artificial Intelligence*, 2007.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. Improving knowledge graph embedding using simple constraints. *arXiv preprint arXiv:1805.02408*, 2018.

Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75, 2002.

Artur d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Serafini, Michael Spranger, and Son N Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*, 2019.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2005.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-Supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems*, 2014.

Samuli Laine and Timo Aila. Temporal Ensembling for Semi-Supervised Learning. In *International Conference on Learning Representations*, 2017.

Neil D Lawrence and Michael I Jordan. Gaussian processes and the null-category noise model. *Semi-Supervised Learning*, pp. 137–150, 2006.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, International Conference on Machine Learning*, 2013.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, 2018.

Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Integrating learning and reasoning with deep logic models. *arXiv preprint arXiv:1901.04195*, 2019.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin 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.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, 2015.

Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems*, 2017.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *IEEE International Conference on Image Processing*, 2016a.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 2016b.

Tian Sang, Paul Beame, and Henry A Kautz. Performing bayesian inference by weighted model counting. In *American Association for Artificial Intelligence*, 2005.

Matthias Seeger. A taxonomy for semi-supervised learning methods. Technical report, MIT Press, 2006.

Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

Antti Tarvainen and Harri Valpola. Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. In *Advances in Neural Information Processing Systems*, 2017.

Leslie G Valiant. A neuroidal architecture for cognitive computation. *Journal of the ACM (JACM)*, 47(5):854–882, 2000.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

Emile van Krieken, Erman Acar, and Frank van Harmelen. Semi-supervised learning using differentiable reasoning. *arXiv preprint arXiv:1908.04700*, 2019.

Po-Wei Wang, Priya L Donti, Bryan Wilder, and Zico Kolter. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, 2019.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International Conference on Machine Learning*, 2018.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

APPENDIX: DERIVATION OF $p(\theta)$ FOR A MIXTURE OF GAUSSIANS

For a general mixture distribution:

$$\theta_k^x \;=\; p(y{=}k|x) \;=\; \sigma\Big(\log \frac{p(x|y{=}k)\pi_k}{\sum_{k'\neq k} p(x|y{=}k')\pi_{k'}}\Big);$$

$$\frac{d\theta_k^x}{dx} \;=\; \theta_k^x(1{-}\theta_k^x)\Big(\tfrac{d}{dx}\log p(x|y{=}k) - \sum_{k'\neq k} \frac{p(x|y{=}k')\pi_k}{\sum_{k''\neq k} p(x|y{=}k'')\pi_{k''}} \tfrac{d}{dx}\log p(x|y{=}k')\Big)$$

which, in our particular case, become:

$$\theta_1^x = \sigma\big(\log\tfrac{\pi_1}{\pi_0} + \tfrac{\mu_1-\mu_0}{\sigma^2}x - \tfrac{1}{2}(\tfrac{\mu_1^2}{\sigma^2} - \tfrac{\mu_0^2}{\sigma^2})\big)\;, \qquad \frac{d\theta_k^x}{dx} = \theta_k^x(1-\theta_k^x)(\tfrac{\mu_1^2}{\sigma^2} - \tfrac{\mu_0^2}{\sigma^2}).$$

Rearranging the former gives x in terms of θ. Substituting into $p(\theta){=}|\boldsymbol{J}|p(x)$ gives:

$$p(\theta) = \sum_{k=0}^{1} \pi_k \underbrace{\sqrt{\tfrac{\sigma^2}{2\pi}}\tfrac{1}{|\mu_1-\mu_0|}\exp\{a(\log\tfrac{\theta_1}{\theta_0})^2 + (b_k-1)\log\theta_1 + (-b_k-1)\log\theta_0 + c_k\}}_{p(\theta|y=k)}$$

with coefficients: $a = \frac{-\sigma^2}{2(\mu_1-\mu_0)^2}$, $b_k = \frac{\mu_k}{\mu_1-\mu_0} + \frac{\sigma^2}{(\mu_1-\mu_0)^2}(\frac{\mu_1^2-\mu_0^2}{\sigma^2} - \log\frac{\pi_1}{\pi_0})$, $c_k = -\frac{(\mu_1-\mu_0)^2 b_k^2}{2\sigma^2}$.