Semantic Objective Functions: A Distribution-Aware Method for Adding Logical Constraints in Deep Learning

Miguel Angel Mendez-Lucero[1][a], Enrique Bojorquez Gallardo[2][b] and Vaishak Belle[1][c]

[1]*School of Informatics, University of Edinburgh, Edinburgh, U.K.*

[2]*Institute of Philosophy (HIW), KU Leuven, Leuven, Belgium*

Keywords: Semantic Objective Functions, Probability Distributions, Logic and Deep Learning, Semantic Regularization, Knowledge Distillation, Constraint Learning, Applied Information Geometry.

Abstract: Issues of safety, explainability, and efficiency are of increasing concern in learning systems deployed with hard and soft constraints. Loss-function based techniques have shown promising results in this area, by embedding logical constraints during neural network training. Through an integration of logic and information geometry, we provide a construction and theoretical framework for these tasks that generalize many approaches. We propose a loss-based method that embeds knowledge—enforces logical constraints—into a machine learning model that outputs probability distributions. This is done by constructing a distribution from the logical formula, and constructing a loss function as a linear combination of the original loss function with the Fisher-Rao distance or Kullback-Leibler divergence to the constraint distribution. This construction is primarily for logical constraints in the form of propositional formulas (Boolean variables), but can be extended to formulas of a first-order language with finite variables over a model with compact domain (categorical and continuous variables), and others statistical models that is to be trained with semantic information. We evaluate our method on a variety of learning tasks, including classification tasks with logic constraints, transferring knowledge from logic formulas, and knowledge distillation.

1 INTRODUCTION

Neuro symbolic artificial intelligence (NeSyAI) has emerged as a powerful tool for representing and reasoning about structured logical knowledge in deep learning (Belle, 2020). Within this area, loss-based methods are a set of techniques that provide an integration of logical constraints into the learning process of neural architectures (Giunchiglia et al., 2022). Such constraints are formulas that represent knowledge about the problem domain. They may be used to improve the accuracy, data, parametric efficiency, interpretability, and safety of deep learning models (Belle, 2020). For example, in robotics, logic constraints can be used to represent safety conditions (Kwiatkowska, 2020; Amodei et al., 2016). This information can help the model make more accurate predictions and provide interpretability, as we are embedding human expert knowledge into the network (Gunning, 2017; Rudin, 2018). Additionally, such expert knowledge reduces the problem space as the agent does not need to learn how to avoid harmful states and/or explore suboptimal policies, reducing the amount of data required to train a deep learning model (Hoernle et al., 2022). Broadly speaking, these neurosymbolic techniques solve the problem of obtaining distributions that *satisfy the constraints* while solving a particular task. Our approach is different, we aim to obtain a model that can potentially sample *all* the instances of the constraint–for which satisfying the constraint becomes a consequence. Finally, this framework can also be used for Knowledge Distillation (KD) (Gou et al., 2020), which consists in using an expert model with a complex architecture and transferring its information into a model with a simpler architecture, reducing size and complexity.

1.1 Problem Description

Suppose we have a task that consists in training a statistical model to be able to describe a set of objects X. A description consists on assigning to each vari-

[a] https://orcid.org/0000-0001-8349-8606
[b] https://orcid.org/0009-0000-3321-7685
[c] https://orcid.org/0000-0001-5573-8465

Mendez-Lucero, M. A., Gallardo, E. B. and Belle, V.
Semantic Objective Functions: A Distribution-Aware Method for Adding Logical Constraints in Deep Learning.
DOI: 10.5220/0013229200003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 909-917
ISBN: 978-989-758-737-5; ISSN: 2184-433X

able from a finite set $\mathcal{V} = \{x_1, ..., x_n\}$ a value in a set A. Therefore, the possible descriptions are contained in the set $A^{\mathcal{V}}$. The descriptive instances $a \in A^{\mathcal{V}}$ are the samples, $A^{\mathcal{V}}$ is the sample space, the variables the *features* and the set of values A is the *domain*. Let us also assume that a single description may not capture completely what the object $x \in X$ is, maybe there are some descriptions that are equally good, or others that are better or worse. This can be accounted for if our statistical model assigns to each object $x \in X$ a probability distribution over $A^{\mathcal{V}}$. Asking the model what the object x is, consists in sampling a description from this distribution, the probability of the sample shows how adequate it is as a description. It is the distribution that holds the information about the object—which is limited by the expressiveness of the sample space. Let us also assume that the model can only associate states from a family of distributions $\mathcal{F}$. Therefore, our statistical model consists of a function,

$$F : X \to \mathcal{F}. \tag{1}$$

This model will be trained by minimizing a loss function $L : \mathcal{F} \to \mathbb{R}^{\geq 0}$.

What if there is external information that we want our model to learn as well? This extra information may come in the form of a formula that expresses the relationships between the variables. To exemplify this, if the variables represent features that can be true or false, then the domain can be seen as the set $\{0,1\}$ (e.g. the Boolean space) and the constraint has the form of a propositional formula generated by a subset of the variables, such as $x_1 \wedge \neg x_3$; if the domain is $\mathbb{R}$, then the formula may determine relationships expressed as inequalities that the outcomes of the variables must hold, such as $x_1^2 + x_2^2 + x_3^2 \leq 1$. These formulas act as constraints over the sample space, they represent knowledge that we want to embed into the system. Not only can we have extra information on the form of a specific region of the sample space, it can be a distribution over a region, allowing more complexity on the constraint. We will refer to the distribution that encodes this additional information as the *constraint distribution*. The problem we want to solve now is: How can we train a statistical model for a given task in a way that it also learns this additional information?

In this paper we provide a general framework for solving the problem of transferring information from a given distribution that may come from a logic constraint or a pretrained neural network. Our contributions can be summarized as follows: we provide a canonical way to transform a constraint—in the form of propositional formulas with finite variables, or First Order Logic (FOL) formulas of a language with finite vocabulary and either with finite or real domain—into a distribution, and then construct a loss function out of it. By minimizing over this loss function we can obtain a model that *learns all possible instances of the constraint*, and as a consequence, *satisfies the constraint*. We can also use this method to distill the information from a pretrained model allowing for more parameter efficient networks that also respect the logic constraints.

The paper is divided into six sections. Section 2 contains related work on the problem of finding deep learning methods with logical constraints and KD. Section 3 presents the main concepts from information geometry and mathematical logic that will be used for the construction of the Semantic Objective Functions (SOFs). Section 4 describes the construction of SOFs given a constraint distribution. Section 5 has experiments that show that these methods solve the problem of learning the constraint in the case of propositional formulas, an experiment with the MNIST and Fashion MNIST databases, another using a FOL formula, and a KD problem. Section 6 states the limitations of our approach, as well as some avenues for future work. Finally, Section 7 concludes the paper.

2 RELATED WORK

There are multiple approaches for enforcing logic constraints into the training of deep learning models (Belle, 2020; Giunchiglia et al., 2022). In this section, we will provide a general overview of the approaches that relate to our contributions and were used as basis of our work.

There are several techniques proposed to incorporate logical constraints into deep learning models. For propositional formulas there are semantic loss function (Xu et al., 2017), and LENSR (Xie et al., 2019). For FOL formulas there are: DL2 (Fischer et al., 2019), for formulas of a relational language, MultiplexNet (Hoernle et al., 2022) for formulas in disjunctive normal form and be quantifier-free linear arithmetic predicates, and DeepProbLog (Manhaeve et al., 2018), which encapsulates the outputs of the neural network in the form of neural predicates and learns a distribution to increase the probability of satisfaction of the constraints.

In addition to regularization methods, Knowledge Distillation (KD) offers another approach for transferring structured knowledge from logic rules to neural networks. Examples include methods proposed in (Hu et al., 2016) and (Roychowdhury et al., 2021). These works build a rule-regularized neural network as a teacher and train a student network to mimic the teacher's predictions. A different approach is Concordia (Feldstein et al., 2023), an innovative system that combines probabilistic logical theories with deep neural networks for the teacher and student, respectively.

Our framework differentiates itself from previous methods in certain key aspects. First, unlike (Hu et al., 2016) and (Roychowdhury et al., 2021), it does not require the additional step of training the teacher network at each iteration. Second, compared to (Hu et al., 2016) and (Roychowdhury et al., 2021) and (Feldstein et al., 2023), it only requires a single loss function for training. This loss function can be used for training the network, regardless of whether the teacher is a logic constraint, a pre-trained network, or another kind of statistical model. This flexibility makes the loss function suitable for both semantic and deep learning KD scenarios. In the case of the loss-based methods, they build a regularizing loss function such that its value is zero whenever the model—or in the case of distributions, the support— satisfies the formula—or is contained in the samples that satisfy the constraint. This makes sense, as we do not want the regularizing term to penalize the distributions that satisfy the constraints. Given that the regularizer is a positive function, then all the elements that have zero value are local minimal, meaning that these approaches—as well as the other approaches—solve the problem of *satisfying the constraint*. whereas our aim is to build a loss function that can *learn a distribution consistent with the models of the formula* i.e. has as its *unique minimal value* the uniform distribution over the set of models of the constraint.

Another limitation is that their solution requires the constraint to be the same for all object $x \in X$, whereas in our approach we can have a constraint that depends on the object x. The only restrictions for being able to use our proposed methodology is that the constraint has to be a formula of a language generated by a *finite* set of variables, and the set of instances that satisfy the formula has non-zero finite measure.

3 BACKGROUND THEORY AND NOTATION

The description of this section is compressed for reasons of space, please refer to (Lee, 2022; Gallot et al., 2004; Bell and Slomson, 2006; Enderton, 2001; Cover and Thomas, 2006; Amari, 2005; Amari and Nagaoka, 2000; Csiszár and Shields, 2004; Watanabe, 2009) for a more comprehensive exposition.

3.1 Information Geometry

In this paper we will take $\mathcal{F}$ to be a family of distributions over a space $A^{\mathcal{V}}$ which is parametrized by a chart $\theta \in \Omega \mapsto p(x;\theta) \in \mathcal{F}$, where Ω is an open subset of $\mathbb{R}^m$. It has the structure of a convex Riemannian manifold where the metric is defined through the Fisher Information metric(Amari and Nagaoka, 2000). The distance in the manifold derived by this metric is known as the Fisher-Rao distance, which is the one we will be using, and for distributions p and q it will be denoted as $D_{\mathcal{F}}(p,q)$. Another important "measure" between probability distributions that we are going to use is the Kullback-Leibler Divergence(Cover and Thomas, 2006). This measure, denoted for distributions p, q as $D_{KL}(p||q)$, can be interpreted as "the inefficiency of assuming that the distribution is q when the true distribution is p"(Cover and Thomas, 2006), and is defined as, $\sum_{x \in A^{\mathcal{V}}} p(x) log\left(\frac{p(x)}{q(x)}\right)$ or $\int_{x \in A^{\mathcal{V}}} p(x) log\left(\frac{p(x)}{q(x)}\right) dx$, for the finite and continuous case respectively.

3.2 Mathematical Logic

The logic constraints we will be using are expressed as formulas of a formal—propositional or FOL—language generated by a *finite* set of variables $\mathcal{V} = \{x_1, ..., x_n\}$. The models of a propositional language can be seen as the assignments of a truth value to each variable, i.e., $Mod(\mathcal{L}_{\mathcal{V}}) = 2^{\mathcal{V}} = \{s : \mathcal{V} \to 2\}$ where $2 = \{0, 1\}$. On the other hand, a FOL formal language, $\mathcal{L}^1_{\mathcal{V}}$, is specified by a type τ, which is a set of relations, functions and constant symbols. The terms are generated by the set of variables $\mathcal{V}$, and the formulas are recursively constructed by taking as atomic formulas the relations applied to terms, and then recursively constructing the rest with the logical operations $\wedge, \vee, \neg$, and quantifiers $\forall$ and $\exists$. A model $\mathfrak{A}$ of $\mathcal{L}^1_{\mathcal{V}}$ consists of a set A–the domain of the model–and an interpretation of each symbol in τ in A, and it will associate each formula φ to a set $M_\varphi \subseteq A^n$. This determines a notion of satisfaction within the model that is recursively defined over the set of formulas— $\mathfrak{A} \models \varphi(a_1, ..., a_n)$ if and only if $(a_1, ..., a_n) \in M_\varphi$[1].

4 SEMANTIC OBJECTIVE FUNCTIONS

Gradient optimization algorithms are commonly used in machine learning to optimize the parameters of a model by minimizing a loss function (Goodfellow et al., 2016). These algorithms work by iteratively updating the model parameters in the direction of the negative gradient of the loss function, which results in a sequence of model parameter values that minimize the loss function.

[1]For more details see (Enderton, 2001)

To address this issue, regularizers are often used to constrain the model's parameters during training, preventing them from becoming too large or too complex. They are added to the loss function as penalty terms, and their effect is to add a bias to the gradient update of the model parameters. Given a loss function L and a regularizing term L', the regularized loss function is a linear combination $\alpha L + \beta L'$, where $\alpha, \beta \in \mathbb{R}$. Our construction also includes the case in which the constraint depends on the object $x \in X$, where the training dataset are of the form $\{(X_i, \rho_i, \varphi_i)\}_{i \in I}$ or $\{(X_i, \varphi_i)\}_{i \in I}$ for the supervised and unsupervised cases respectively. In this section we will introduce the Semantic Objective Functions (SOFs) for each case and write down the explicit form in some important finite and continuous cases.

4.1 Constraint Distribution of a Formula

An important part of the construction of the loss terms associated to a formula φ is to construct a probability distribution that is a associated to the formula in a canonical way. What we want is a distribution that samples all the models of the formula with equal probability. For each formula φ, the size of the region that it constraints is denoted as A_φ–which is $|M_\varphi|$ in the propositional case, or A is finite, and $\int_{M_\varphi} 1dx$ when $A = \mathbb{R}$, or any measurable space.

The propositional case has a quite simple representation. Given that the set of models is finite (2^n if n is the amount of atomic propositions), we can order the set of models as $M_1, ..., M_{2^n}$ and represent the constraint distribution of the formula φ as a tuple

$$\rho_\varphi = (p_1, ..., p_{2^n}) \tag{2}$$

where $p_i = 1/A_\varphi$ if $p_i \models \varphi$ and 0 otherwise. An example of this is the *XOR* formula. If we order the models of a propositional language with two variables x_1 and x_2 as $M_1 = (x_1 = True, x_2 = True), M_2 = (x_1 = True, x_2 = False), M_3 = (x_1 = False, x_2 = True)$ and $M_4 = (x_1 = False, x_2 = False)$, then its associated constraint distribution is $\rho_{XOR} = (0, 0.5, 0.5, 0)$.

The FOL case is defined as,

$$\rho_\varphi(a_1, ..., a_n) = \begin{cases} 1/A_\varphi & \text{if } (a_1, ..., a_n) \in M_\varphi \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

These distributions are only well-defined whenever $0 < A_\varphi < \infty$, meaning that M_φ has finite non-zero measure. In the propositional case, this is satisfied whenever the formula is not equivalent to a contradiction (has at least one model). In the FOL this may be interpreted as having that the probability of sampling a model of the formula is not zero (non-zero measure), or that it is not the case there are so many models that the probability of sampling any model is very close to zero, because the larger the amount of models, the less probability of sampling them (finite measure).

4.2 Propositional and Finite Domain Constraints

Given a constraint distribution ρ that is propositional or is interpreted in a model with finite domain, then the Fisher distance has a closed form(Belousov, 2017) and can be used to construct the semantic regularizer as $L_\rho(f) = D_{\mathcal{F}}(\rho, f)$. For these cases the family $\mathcal{F}$ is the $n-1$-simplex $\{f : A^n \to [0,1] | \sum_{s \in 2^n} f(s) = 1\}$. For the propositional case $A = \{0,1\}$, and for the finite categorical case we are working on a model $\mathfrak{A}$ of a FOL language of type τ with domain $A = \{a_1, ..., a_n\}$. To each formula φ, its associated constraint distribution is the uniform distribution $\rho_\varphi \in \mathcal{F}$ defined in 2. In this case the Fisher distance is defined as,

$$D_{\mathcal{F}}(p,q) = arccos\left(\sum_{s \in A^n} \sqrt{p(s)}\sqrt{q(s)}\right). \tag{4}$$

Therefore, the semantic regularizer can be defined as,

$$L_\varphi(f) = arccos\left(\sum_{s \in M_\varphi} \frac{\sqrt{f(s)}}{\sqrt{|M_\varphi|}}\right). \tag{5}$$

4.2.1 Continuous Domain

This is the case where the constraint is given as a formula of a FOL language of type τ in a model $\mathfrak{A}$ with continuous domain A. The terms are generated by the finite set $\mathcal{V}$, therefore the sample space is A^n. For this case we will use the Kullback-Leibler Divergence. There is a restriction on the formulas that we can apply this construction to. They have to be formulas $\varphi \in \mathcal{L}^1_{\mathcal{V}}$ such that M_φ is of finite non-zero measure, so that ρ_φ, as defined in 3, is well defined. The semantic regularizer associated to φ is, $L_\varphi(f) = D_{KL}(\rho_\varphi \| f)$. This function can be rewritten as a sum—or integral—over the set M_φ, which we can obtain through knowledge compilation techniques(Darwiche and Marquis, 2002).

4.3 Extending Weighted Model Counting

As mentioned in Section 3, there are limitations on some semantic regularization techniques that use a function that assigns weights to each atomic formula. This is the case in (Xu et al., 2017), where the logic constraint regularization is defined as logarithm the weighted model counting (WMC) (Chavira and Darwiche, 2008) which is defined as:

$$WMC(\varphi,\omega) = \sum_{M\models\varphi} \prod_{l\in Lit(M)} \omega(l).^2 \qquad (6)$$

Weighted Model Counting defines a weight function $\omega : Lit(M) \to \mathbb{R}^{\geq 0}$ over the literals, so that we can then calculate the probability of satisfaction of each models. Instead, from Equation 6, we can generalize the weight function ω to a weight function for the models, which is of the form $\omega : M_\varphi \to \mathbb{R}$ subject to $\sum_{x\in Mod(\mathcal{L}^0_{\mathcal{V}})} \omega(x) = 1$. That is, $WMC(\varphi,\omega) = \sum_{x\in M_\varphi} \omega(x)$. Therefore, a natural extension of Equation 6 is to extend it to FOL logic constraints with a finite amount of variables as,

$$W(\varphi,\omega) = \int_{M_\varphi} \omega(x_1,...,x_n)dx_1...dx_n. \qquad (7)$$

There is another generalization of WMC known as Weighted Model Integration (WMI) (Belle et al., 2015). That generalization extends WMC to hybrid domains that can take into account both Boolean and continuous variables simultaneously, and the constraints are formulas of propositional logic and a first order language for linear rational arithmetic. In that sense, 7 is the particular case of WMI whenever there are only first order formulas of linear arithmetic logic. It does not generalize to hybrid domains, but deals with measurable spaces more straightforwardly.

5 EXPERIMENTS

This section showcases the capabilities of the SOFs framework through focused experiments. We aim to demonstrate its advantages and potential use cases, rather than achieve state-of-the-art performance (left for future work). We employ propositional logic formulas in experiments from Section 5.1 for clear demonstration. Experiment in Section 5.2 explores Knowledge Distillation over pretrained models. Finally, experiment of Section 5.1 illustrates how to use SOFs for first-order logic (FOL) formulas with a finite amount of variables.

5.1 Classification Tasks with Logic Constraints

We evaluate the effectiveness of Semantic Objective Functions (SOFs) as a regularizer for image classification. We compare SOFs with other regularization techniques on two popular benchmark datasets: MNIST(LeCun et al., 2010), containing handwritten digits 0-9, and Fashion MNIST(Xiao et al., 2017), with images of 10 clothing categories. Both datasets use grayscale images. To enforce a constraint that each image belongs to exactly one class, we utilize a logical formula $\varphi = \bigvee_{i=1}^{n}((\bigwedge_{j\neq i,j=1}^{n} \neg x_j) \wedge x_i)$ representing one-hot encoding.

Our classification model is a simple four-layer Multilayer Perceptron (MLP) with a structure of [784,512,512,10] units per layer. We employ ReLU activation functions in all hidden layers for efficient learning. In the final layer, however, we deviate from the typical SoftMax activation used in single-class classification. Instead, we adopt a sigmoid function. This aligns with the work of (Xu et al., 2017), where the network's output represents the satisfaction probability for each class belonging to the image. To ensure a fair comparison and maintain consistency with their approach, we calculate the probability distribution over the models using the weighing function defined in (Xu et al., 2017), rather than the more general form presented in Equation 7. We incorporated the semantic regularizers (Semloss, WMC, L^2-norm, Fisher-Rao distance and KL-Divergence) as an additional loss term in an MLP with MSE as the main loss. Experiments tested different regularization weights $\lambda \in \{1, 0.1, 0.01, 0.001, 0.0001\}$ during batched training (128 images, Adam optimizer, learning rate 0.001, 10 epochs). We defer a wider method comparison to future work.

Table 1 shows the results of training the MLP with same initial parameters, using different regularizers. We performed a grid search over the different λ's and displayed the best results. Semantic regularizers provide an advantage of improving the accuracy by 1% using the One-Hot-Encoding formula. While it is not shown on the table, the regularizer provides faster convergence than without regularization. Another important result is that the accuracy does not vary much for the different λ in the case of the KL-Div, Fisher distance and L^2-norm, their results never went lower than 94%. This was not the case with the other regularizers, both Semloss and WMC went lower than 10% in both datasets when λ is large enough. We believe this has to do with the fact that their functions have a lot of minimal elements, whereas the rest only have one.

5.2 Constraint Learning Through Knowledge Distillation on Classification Tasks

In this experiment we want to demonstrate the KD capabilities of our SOFs. As explained in Section 2,

[2] $Lit(M)$ is the set of atomic propositions or their negations that are satisfied at M.

Table 1: Learning Classification tasks with semantic regularizers (SR). The results displayed are the best average accuracy after ten epochs. The best accuracy from all the regularizers is show in bold.

	Fashion-MNIST		MNIST	
SR	λ	Acc%	λ	Acc%
WMC	0.01	87.74± .64	0.01	97.99 ± .27
Semloss	0.0001	87.81 ± .64	0.0001	97.78 ± .28
Fisher	0.1	88.23 ± .63	0.1	**98.44** ± .24
KL-Div	0.01	**88.24** ± .63	0.001	98.35± .24
L^2 Norm	0.01	88.22 ± .63	0.1	98.27 ± .25
NoReg	0	75.00	0	97.60

SOFs can enforce the knowledge obtained by a pre-trained statistical model, not just those defined by formulas. In essence, if a neural network successfully learns a distribution through an SOF, the knowledge can be transferred to a new network using the same SOF.

The expert model we take is the one that was trained during the experiments 5.1 and showed best performance per dataset. It is important to notice that this model not only knows how to classify, but it also satisfies the constraint. We trained a smaller MLP with layers [784,256,256,10] to learn how to solve the MNIST and Fashion-MNIST in two ways: using the SOFs as a regularizer, as in Experiment 5.1, or as the main loss function, as a process of KD. In the case of the regularizer, we tested for hyperparameters $\lambda \in \{0, 0.0001, 0.001, 0.01, 0.1, 1.0\}$. and displayed the best results for each SOF. For the case of KD, given that we take the SOF as the total loss function, the only information we use to learn the task is the one provided by the expert model. The labels are not used in training, just for measuring the accuracy. In both cases we take an adam optimizer, batch size of 128, learning rate of 0.001, and train for 10 epochs.

Table 2 shows the results of this experiment. The results show a small reduction on the accuracy of the model compared against the expert models used in training. In the case when the loss function was used as a regularizer, the accuracy was reduced by 0.3% and .24%, for Fashion-MNIST and MNIST respectively. Using the KD method the loss is 0.67%, 0.24% for Fashion-MNIST and MNIST respectively. *It is important to point out that this loss on accuracy comes with a reduction in the amount of required parameters, which is half of the expert model.* Another observation is that these models still remain more accurate than some models—Semloss and WMC—that were trained on experiments 2, see table 1.

5.3 Preliminary Results for First Order Logic Formula

This experiment uses Semantic Objective Functions to learn a probability density function (pdf) that closely matches assignments satisfying a finite-variable FOL formula. To use the Fisher-Rao distance we have to approximate a uniform distribution over these assignments with a distribution from the family of distributions that we are using. If the formula's true pdf does not belong to this family (which will be the case of any family of continuous distributions), the learned distribution will only be an approximation. This lets us leverage the formulas defined in the external case from Section 4.2.1.

To show that our framework is independent of the type of FOL formula, we will be using the following formula over real variables:

$$\begin{aligned} \varphi(x_1, x_2) = (x_1^2 + x_2^2 \leq 1) \wedge \\ \neg \exists z ((z \neq 0) \wedge ((x + z^2 = 0) \vee (y + z^2 = 0))) \end{aligned} \tag{8}$$

Like in Sections 4.3, in order to define our learning method we first need to calculate the set of valid assignments M_φ. For this, a common practice is to rely on SMT solvers or knowledge compilation techniques to compute the set of assignments in which the formula is valid (Darwiche and Marquis, 2002). For this experiment however, we can observe through a quick inspection that the formula defines geometrically the set of all points on the first quadrant of the unitary circle with center in (0,0), its area is $\frac{\pi}{4}$. With the valid ranges defined above and a change of variables to polar coordinates, were $x = rcos\vartheta$ and $y = rsen\vartheta$, we can define a "semantic loss"-like function as (Xu et al., 2017) and Equation 7 for continuous domain as:

$$W(\varphi, \theta) = \int_0^1 \int_0^{\pi/2} f(rcos(\vartheta), rsin(\vartheta); \theta) r d\vartheta dr, \tag{9}$$

where $f(x, y; \theta)$ is the ω function from Equation 7, which will be replaced by the definition of the bi-

Table 2: Regularizers vs Knowledge Distillation. The tables show the results when using the expert network as regularizer or directly for knowledge distillation. The results displayed are the best average accuracy after ten epochs. The best accuracies for both experiments are shown in bold.

	Regularizer				Knowledge Distillation	
	Fahion-MNIST		MNIST		Fahion-MNIST	MNIST
Name	λ	Acc%	λ	Acc%	Acc%	Acc%
KL-Div	0.001	**87.94**±.63	1.0	**98.20**±.26	**87.57**±.64	**98.12**±.26
L^2-Norm	1.0	87.75±.64	0.1	98.02±.27	87.26±.65	98.11±.26
Fisher.D.	1.0	87.60±.63	0.01	98.01±.27	87.38±.65	97.98±.27

variate normal distribution and we will define our learnable parameter $\theta = \mu$. That is, our neural architecture will be a simple bivariate normal distribution and our learnable parameter will be the mean vector μ. Now for our SOF, since we need to only integrate over the valid assignments, and the uniform distribution will always give the same value over those assignments of the formula we can replace the value $\rho_\varphi(x,y)$ to $\frac{1}{\frac{\pi}{4}} = \frac{4}{\pi}$. We also change the variables to polar coordinates, obtaining the expression

$$D_{KL}(\rho_\varphi || f_\theta) = -\frac{4}{\pi}\int_0^1\int_0^{\frac{\pi}{2}} log(f_\theta(r\cos\vartheta, r\sin\vartheta))rd\vartheta dr \tag{10}$$

With Equation 7, we can define a FOL version of the WMC and Semantic Loss which should maximize the probability of satisfaction of the formula. We conducted the experiment over 10 different seeds, over 2000 epochs and by using Adam and stochastic gradient descent optimizer. The results of the experiment can be found on Table 3, where we display the best results from a grid search of learning rate $\alpha \in \{1, 0.1, 0.01, 0.001, 0.0001\}$. To measure the difference between the target and learned density function we used the Total Variation Distance which is defined as, $\delta(p,q) = \frac{1}{2}\int |p(x) - q(x)|dx$.

As expected our SOF was the one to attain the lowest error in comparison with the W and Wlog functions. It is important to mention that none of the loss functions could arrive to the correct solution. The reason is due to the fact that the bi-variate normal distribution does not have a shape that could realistically cover the area defined by the formula.

6 LIMITATIONS AND FUTURE WORK

This work has several limitations that motivate future research directions. First, we assume that for each formula φ, the set M_φ is readily available. This assumption necessitates solving the propositional satisfiability problem (SAT) or the counting satisfiability problem (#SAT) for propositional logic, and solving systems of inequalities for general functions for first-order logic. As mentioned in Section 2, other approaches share this assumption and leverage knowledge compilation techniques to address it (Darwiche and Marquis, 2002; Barrett et al., 2009). An important avenue for future work is to develop approximation methods for the set of satisfying assignments that circumvent the need for explicitly computing M_φ. Second, our current approach is limited to cases where M_φ has non-zero finite measure. Third, the Fisher-Rao distance may not always have a closed-form solution, which can lead to computational challenges. Further research is needed to determine when the Fisher-Rao distance is preferable to the KL-divergence, considering their computational complexity and effectiveness in various settings. Fourth, a comprehensive evaluation comparing our method with existing approaches and testing a wider range of logic constraints is necessary. The experiments of Sections 5.1 and 5.2 were used mainly to show that these methodologies work, but it is hard to see how much better they are because the other architectures perform well without the constraint, so it is also necessary to perform the experiments in tasks where there is a lower performance for state-of-the-art methods. Finally, it would be interesting to investigate the effectiveness of our method for a combination of SOFs obtained from a set of constraints $\{\rho_i\}_{i\in I}$ as well as further experimental analysis on constraint satisfaction using KD methods as in Section 5.2

7 CONCLUSION

In this paper, we proposed a framework and general construction of loss functions out of constraints and expert models, which can be used to transfer knowledge and enforce constraints into learned models. We divided the SOF constructions into two cases, internal and external, and for each case, we looked at closed forms for both finite and continuous domains. We ran experiments for classification tasks and both SOFs, KL-Divergence and Fisher-Rao distance, showed bet-

Table 3: Total Variation Distance for different optimizers and loss functions of φ.

Optimizer	Loss	Learning Rate α	Avg. Total Variation Distance
ADAM	W	0.0001	$0.37111863 \pm 0.03492265$
ADAM	logW	0.0001	$0.36836797 \pm 0.043502506$
ADAM	KLDiv	0.01	**0.2161428 ± 7.386059e-06**
SGD	W	0.0001	$0.39597395 \pm 0.049845863$
SGD	logW	0.0001	$0.37211606 \pm 0.04976438$
SGD	KLDiv	0.001	**0.212498 ±0.0048501617**

ter accuracy than the other proposals (Xu et al., 2017; Chavira and Darwiche, 2008). We also illustrated the use of SOFs for finite variable first-order logic formulas showing promising results. Finally, we conducted KD experiments for two scenarios: when the set of satisfying assignments (SOFs) was used as a regularizer and when it served as the main loss function. These experiments are promising, demonstrating similar accuracy compared to their teacher models while requiring only half the number of parameters.

ACKNOWLEDGEMENTS

Vaishak Belle was supported by a Royal Society Research Fellowship. Miguel Angel Mendez Lucero was supported by CONACYT Mexico.

REFERENCES

Amari, S. and Nagaoka, H. (2000). *Methods of Information Geometry*. Translations of mathematical monographs. American Mathematical Society.

Amari, S.-i. (2005). Information geometry and its applications. *Journal of Mathematical Psychology*, 49:101–102.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Barrett, C., Sebastiani, R., Seshia, S. A., Tinelli, C., Biere, A., Heule, M., van Maaren, H., and Walsh, T. (2009). Handbook of satisfiability. *Satisfiability modulo theories*, 185:825–885.

Bell, J. and Slomson, A. (2006). *Models and Ultraproducts: An Introduction*. Dover Books on Mathematics Series. Dover Publications.

Belle, V. (2020). Symbolic logic meets machine learning: A brief survey in infinite domains. In Davis, J. and Tabia, K., editors, *Scalable Uncertainty Management - 14th International Conference, SUM 2020, Bozen-Bolzano, Italy, September 23-25, 2020, Proceedings*, volume 12322 of *Lecture Notes in Computer Science*, pages 3–16. Springer.

Belle, V., Passerini, A., and Van den Broeck, G. (2015). Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2015, pages 2770–2776.

Belousov, B. (2017). Geodesic distance between probability distributions is not the kl divergence. 11 July, 2017.

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

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

Csiszár, I. and Shields, P. (2004). Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528.

Darwiche, A. and Marquis, P. (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264.

Enderton, H. (2001). *A Mathematical Introduction to Logic*. Elsevier Science.

Feldstein, J., Jurčius, M., and Tsamoura, E. (2023). Parallel neurosymbolic integration with concordia. *arXiv preprint arXiv:2306.00480*.

Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., and Vechev, M. T. (2019). Dl2: Training and querying neural networks with logic. In *International Conference on Machine Learning*.

Gallot, S., Hulin, D., and Lafontaine, J. (2004). *Riemannian Geometry*. Universitext. Springer Berlin Heidelberg.

Giunchiglia, E., Stoian, M. C., and Lukasiewicz, T. (2022). Deep learning with logical constraints. In Raedt, L. D., editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5478–5485. ijcai.org.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2020). Knowledge distillation: A survey. *CoRR*, abs/2006.05525.

Gunning, D. (2017). Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2.

Hoernle, N., Karampatsis, R., Belle, V., and Gal, K. (2022). Multiplexnet: Towards fully satisfied logical constraints in neural networks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence,* *EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 5700–5709. AAAI Press.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. (2016). Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420.

Kwiatkowska, M. (2020). Safety and robustness for deep learning with provable guarantees. In *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1–3. IEEE.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lee, J. (2022). *Manifolds and Differential Geometry*. Graduate Studies in Mathematics. American Mathematical Society.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and Raedt, L. D. (2018). Deepproblog: Neural probabilistic logic programming. *CoRR*, abs/1805.10872.

Roychowdhury, S., Diligenti, M., and Gori, M. (2021). Regularizing deep networks with prior knowledge: A constraint-based approach. *Knowledge-Based Systems*, 222:106989.

Rudin, C. (2018). Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*.

Watanabe, S. (2009). *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, Y., Xu, Z., Meel, K. S., Kankanhalli, M. S., and Soh, H. (2019). Embedding symbolic knowledge into deep networks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4235–4245.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and den Broeck, G. V. (2017). A semantic loss function for deep learning with symbolic knowledge. *CoRR*, abs/1711.11157.