# A Semantic Loss Function for Deep Learning Under Weak Supervision[*]

**Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang & Guy Van den Broeck**
Computer Science Department
University of California, Los Angeles

## Abstract

This paper develops a novel methodology for using symbolic knowledge in deep learning. We define a semantic loss function that bridges between neural output vectors and logical constraints. This loss function captures how close the neural network is to satisfying the constraints on its output. An experimental evaluation shows that our semantic loss function effectively guides the learner to achieve (near-)state-of-the-art results on semi-supervised multi-class classification. Moreover, it significantly increases the ability of the neural network to predict structured objects under weak supervision, such as rankings and shortest paths.

## 1 Introduction

The widespread success of representation learning raises the question of which AI tasks are amenable to deep learning, which require classical model-based symbolic reasoning, and whether we can benefit from an integration of both. In recent years, significant effort has gone towards various ways of using representation learning to solve tasks that were previously tackled by symbolic methods. Such efforts include neural computers, Turing machines, and differentiable programming (e.g., [15, 31, 32, 37]), relational embeddings, deep learning for graph data, and neural theorem proving (e.g., [1, 12, 25, 26]), and many more. Other work has sought to augment deep learning with symbolic knowledge through logical or arithmetic constraints (e.g., [9–11, 16, 19, 21, 24, 28, 33, 35, 36]).

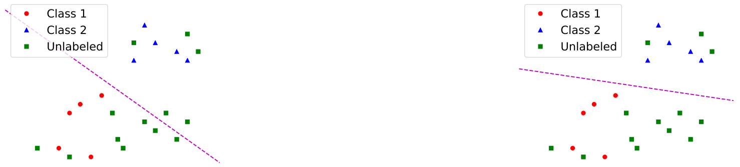
We consider learning tasks where symbolic knowledge is provided to connect the different outputs of a neural network. This knowledge takes the form of a constraint (or sentence) in Boolean logic. It can be as simple as an exactly-one constraint for one-hot output encodings, or as complex as a structured output prediction constraint for intricate combinatorial objects such as rankings, subgraphs, and paths. Our goal is to augment neural network with the ability to learn how to make predictions subject to these constraints, and use the symbolic knowledge to improve its performance.

Most neuro-symbolic approaches aim to simulate or learn symbolic reasoning in an end-to-end deep neural network, or capture symbolic knowledge in a vector-space embedding. This choice is partly motivated by the need for smooth *differentiable* models; adding symbolic reasoning code (e.g., SAT solvers) to a deep learning pipeline destroys this property. Unfortunately, while making reasoning differentiable, the precise logical meaning of the knowledge is often lost. In this paper, we take a distinctly different approach. We first define a differentiable *semantic loss* function that captures how well the outputs of a neural network match a given constraint. This function precisely captures the *meaning* of the constraint, and is independent of its *syntax*.

Next, we show how this semantic loss gives *significant practical improvements* in semi-supervised classification. The semantic loss defined over the exactly-one constraint in this setting permits us to obtain a learning signal from vast amounts of unlabeled data. The key idea is that the semantic loss

[*] The long version of this paper [39] (available at `https://arxiv.org/abs/1711.11157`) derives the semantic loss function from first principles, and discusses more experimental details and related work.

NIPS 2017 Workshop on Learning with Limited Labeled Data: Weak Supervision and Beyond.



(a) Trained without semantic loss

(b) Trained with semantic loss

Figure 1: Binary classification toy example: a linear classifier without and with semantic loss.

helps us improve how consistently we are able to classify the unlabeled data. This simple addition to the loss function of standard deep learning architectures yields (near-)state-of-the-art performance in semi-supervised classification on MNIST, FASHION and CIFAR-10 datasets. Our final set of experiments study the benefits of the semantic loss function for complex structured output learning tasks, such as preference learning and path prediction in a graph [2, 4, 8, 15] to show how it can be generalized onto more difficult problems. By capturing the structure of the output space with logical constraints, and minimizing the semantic loss for this constraint during learning, we are able to learn networks that are much more likely to correctly predict structured objects.

## 2 Semantic Loss and its Application in Semi-Supervised Classification

The goal of a semantic loss function is to bridge the gap between the continuous world of neural networks, and the symbolic world of propositional logic. The semantic loss $\mathrm{L}^{\mathrm{s}}(\alpha, \mathsf{p})$ is a function of a sentence $\alpha$ in propositional logic, defined over variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, and a vector of probabilities $\mathsf{p}$ for the same variables $\mathbf{X}$. Element $\mathsf{p}_i$ denotes the predicted probability of variable $X_i$, and corresponds to a single output of the neural net. For example, the semantic loss between an exactly-one constraint $\alpha$ and a neural net output vector $\mathsf{p}$ captures how close the prediction $\mathsf{p}$ is to having exactly one output set to true (1), and all false (0), regardless of which output is correct.

We desire our semantic loss $\mathrm{L}^{\mathrm{s}}(\alpha, \mathsf{p})$ to be proportional to the negative log-probability of $\alpha$ being satisfied when sampling values according to $\mathsf{p}$. More formally,

$$\mathrm{L}^{\mathrm{s}}(\alpha, \mathsf{p}) \propto -\log \sum_{\mathbf{x} \models \alpha} \prod_{i:\mathbf{x} \models X_i} \mathsf{p}_i \prod_{i:\mathbf{x} \models \neg X_i} (1 - \mathsf{p}_i). \tag{1}$$

Here, $\mathbf{x} \models \alpha$ means that assignment $\mathbf{x}$ to variables $\mathbf{X}$ satisfies sentence $\alpha$, and $\mathbf{x} \models X_i$ means that $X$ is set to true in world $\mathbf{x}$. Intuitively, this is the self-information of obtaining an assignment that satisfies the constraint. This equation is derived from first principles in the long version of this paper.

**Illustrative Example** To illustrate the benefit of semantic loss in the semi-supervised setting, we begin our discussion with a small toy example. Consider a binary classification task as depicted in Figure 1. Ignoring the unlabeled examples, a simple linear classifier learns to distinguish the two classes by separating the labeled examples in Figure 1a. However, the unlabeled examples are also informative, as they must carry some properties that give them a particular label. This is the crux of semantic loss: a model must confidently assign a consistent class even to unlabeled data. Encouraging the model to do so results in a more accurate decision boundary, as illustrated in Figure 1b. Next, we further explore this idea and apply it to real-world image classification tasks.

### 2.1 Algorithm

Our proposed method intends to be generally applicable and compatible with any feedforward neural network. The semantic loss is simply another regularization term that can directly be plugged into an existing loss function. More specifically, for some weight $w$, the new overall loss becomes

$$\text{existing loss} + w \cdot \text{semantic loss}.$$

When the constraint over the output space is simple (for example, there is a small number of solutions $\mathbf{x} \models \alpha$), the semantic loss can be directly computed with Equation 1. Concretely, for the

Table 1: MNIST. Previously reported test accuracies followed by semantic loss results (± stddev)

| Accuracy % with # of used labels | 100 | 1000 | ALL |
|---|---|---|---|
| AtlasRBF [29] | 91.9 (± 0.95) | 96.32 (± 0.12) | 98.69 |
| Deep Generative [17] | 96.67(± 0.14) | 97.60(± 0.02) | 99.04 |
| Virtual Adversarial [22] | 97.67 | 98.64 | 99.36 |
| Ladder Net [30] | **98.94** (±0.37 ) | **99.16** (±0.08) | **99.43** (± 0.02) |
| Baseline: MLP, Gaussian Noise | 78.46 (±1.94) | 94.26 (±0.31) | 99.34 (±0.08) |
| Baseline: Self-Training | 72.55 (±4.21) | 87.43 (±3.07 ) | 99.34 (±0.08) |
| MLP with Semantic Loss | 98.38 (±0.51) | 98.78 (±0.17) | 99.36 (±0.02) |

exactly-one constraint used in $n$-class classification, the semantic loss reduces to

$$\mathrm{L}^{\mathrm{s}}(\text{exactly-one}, \mathsf{p}) \propto -\log \sum_{i=0}^{n-1} \mathsf{p}_i \prod_{j=0, j\neq i}^{n-1} (1 - \mathsf{p}_j),$$

where he values $\mathsf{p}_i$ denote the probability of class $i$ as predicted by the neural net. The semantic loss for the exactly-one constraint is efficient and causes no noticeable overhead in our experiments.

In general, for arbitrary constraints $\alpha$, the semantic loss is not efficient to compute using Equation 1, and more advanced automated reasoning is required. Section 3 discusses this issue in more detail.

### 2.2 Experimental Evaluation

We evaluate semantic loss in the semi-supervised setting by comparing it with several competitive models. We add semantic loss to the same base models used in ladder nets [30], which currently achieve state-of-the-art results on semi-supervised MNIST and CIFAR-10 [18]. Specifically, the MNIST base model is a fully-connected multilayer perceptron (MLP), with layers of size 784-1000-500-250-250-250-10. On CIFAR-10, it is a 10-layer convolutional neural network (CNN) with 3-by-3 padded filters. After every 3 layers, features are subject to a 2-by-2 max-pool layer with strides of 2. We refer to the long version of this paper for further details.

For all semi-supervised experiments, we use the standard 10,000 held-out test examples provided in the original datasets and randomly pick 10,000 training examples as validation set. For values of $N$ that depend on the experiment, we retain $N$ randomly chosen labeled examples from the training set, and remove labels from the rest. We balance classes in the labeled samples to ensure no particular class is over-represented.

**MNIST** The permutation invariant MNIST classification task is commonly used as a test-bed for general semi-supervised learning algorithms. We run experiments for 10 epochs, with a batch size of 10 labeled and 10 unlabeled examples. Experiments are repeated 10 times with different random seeds. Table 1 compares semantic loss to two baselines and state-of-the-art results from the literature. The first baseline is a purely supervised MLP, which makes no use of unlabeled data. The second is the classic self-training method for semi-supervised learning, which operates as follows. After every 1000 iterations, the unlabeled examples that are predicted by the MLP to have more than $95\%$ probability of belonging to a single class, are assigned a psuedo-label and become labeled data.

When given 100 labeled examples ($N = 100$), MLP with semantic loss gains around $20\%$ improvement over the purely supervised baseline. The improvement is even larger ($25\%$) compared to self-training. Considering *the only change is an additional loss term*, this result is very encouraging. Compared to the state of the art, ladder nets slightly outperform semantic loss by $0.5\%$ accuracy. This difference may be an artifact of the excessive tuning of architectures, hyper-parameters and learning rates that the MNIST dataset has been subject to.

**FASHION** The FASHION [38] dataset consists of Zalando's article images, aiming to serve as a more challenging drop-in replacement for MNIST. Arguably, it has not been overused and requires more advanced techniques to achieve good performance. As in the previous experiment, we run our method for 10 epochs, whereas ladder nets need 100 epochs to converge. Again, experiments are repeated 10 times and Table 2 reports the classification accuracy and its standard deviation (except for $N =$ all where it is close to 0 and omitted for space).

Table 2: FASHION. Test accuracy comparison between MLP with semantic loss and ladder nets.

| Accuracy % with # of used labels | 100 | 500 | 1000 | ALL |
|---|---|---|---|---|
| Ladder Net [30] | 81.46 (±0.64 ) | 85.18 (±0.27) | 86.48 (± 0.15) | **90.46** |
| Baseline: MLP, Gaussian Noise | 69.45 (±2.03) | 78.12 (±1.41) | 80.94 (±0.84) | 89.87 |
| MLP with Semantic Loss | **86.74** (±0.71) | **89.49** (±0.24) | **89.67** (±0.09) | 89.81 |

Table 3: CIFAR. Test accuracy comparison between CNN with semantic loss and ladder nets.

| Accuracy % with # of used labels | 4000 | ALL |
|---|---|---|
| CNN Baseline in Ladder Net | 76.67 (± 0.61) | 90.73 |
| Ladder Net [30] | 79.60 (±0.47) | |
| Baseline: CNN, Whitening, Cropping | 77.13 | **90.96** |
| CNN with Semantic Loss | **81.79** | 90.92 |

Experiments show that utilizing semantic loss results in a very large $17\%$ improvement over the baseline when only 100 labels are provided. Moreover, our method compares favorably to ladder nets, except when the setting degrades to be fully supervised. Note that our method already nearly reaches its maximum accuracy with 500 labeled examples, which is only $1\%$ of the training dataset.

**CIFAR-10** To show the general applicability of semantic loss, we evaluate it on CIFAR-10. This dataset consisting of 32-by-32 RGB images in 10 classes. A simple MLP would not have enough representation power to capture the huge variance across objects within the same class. To cope with this spike in difficulty, we switch our underlying model to a 10-layer CNN as described earlier. We use a batch size of 100 samples of which half are unlabeled. Experiments are run for 100 epochs. However, due to our limited computational resources, we report on a single trial. Note that we make slight modifications to the underlying model used in ladder nets to reproduce similar baseline performance. Please refer to the long version of this paper for the details of this experimental setup.

As shown in Table 3, our method compares favorably to ladder nets. However, due to the slight difference in performance between the supervised base models, a direct comparison would be methodologically flawed. Instead, we compare the net improvements over baselines. In terms of this measure, our method scores a gain of $4.66\%$ whereas ladder nets gain $2.93\%$.

### 2.3 Discussion

Overall, the experiments so far have demonstrated the competitiveness and general applicability of our proposed method on semi-supervised learning tasks. It surpassed the previous state of the art (i.e. ladder nets ) on FASHION and CIFAR-10, while being close on MNIST. Considering the simplicity of our method, such results are encouraging. Indeed, a key advantage of semantic loss is that it only requires a simple additional loss term. Without changing the network architecture itself, we incur almost no computational overhead. Conversely, this property makes our method sensitive to the underlying model's performance. Without the underlying predictive power of a strong supervised learning model, we do not expect to see the same benefits we observe here. Recently, we became aware that [22] extended their work to CIFAR-10 and achieved state-of-the-art results [23], surpassing our performance by $5\%$. In future work, we plan to investigate whether applying semantic loss on their architecture would yield an even stronger performance. The objective of semantic loss to increase the confidence of predictions on unlabeled data is in common with information-theoretic approaches to semi-supervised learning [13, 14, 22]. A key difference between semantic loss and information-theoretic losses is that semantic loss generalizes to arbitrary output constraints [39].

## 3 Learning with Complex Constraints

While much of current machine learning research is focused on problems such as multi-class classification, there remain a multitude of difficult problems involving highly constrained output domains. Because semantic loss is defined by a Boolean formula, it can be used on any output domain that can be fully described in this manner. Here, we develop a framework for tractable semantic loss on highly complex constraints, and evaluate it on some difficult examples.

Table 4: Grid shortest path test accuracy.

| Test accuracy % | Coherent | Incoherent |
|---|---|---|
| 5-layer MLP | 5.62 | **85.91** |
| With semantic loss | **28.51** | 83.14 |

Table 5: Peference prediction test accuracy.

| Test accuracy % | Coherent | Incoherent |
|---|---|---|
| 3-layer MLP | 0.01 | **75.74** |
| With semantic loss | **13.59** | 72.43 |

### 3.1 A Tractable Semantic Loss

Our goal here is to develop a method for computing both the semantic loss and its gradient in a tractable manner. Examining Equation 1, we see that the right-hand side is a well-known automated reasoning task called weighted model counting (WMC) [3, 34]. A key property of WMC is that its partial derivatives can be computed in terms of other, slightly modified WMCs. Furthermore, we know of circuit languages that compute weighted model counts, and that are amenable to back-propagation [5]. We use the language and circuit compilation techniques described in [6] to build a Boolean circuit representing our semantic loss. Due to certain properties of this circuit form, we can use it to compute both the values and the gradients of the semantic loss in time linear in the size of the circuit [7]. Once we have constructed this function, we can add it to our standard loss function as described in Section 2.1.

### 3.2 Experimental Evaluation: Grids & Preference Learning

Our ambition when evaluating semantic loss' performance on complex constraints is not to achieve state-of-the-art performance on any particular problem, but rather to highlight its effect. To this end, we evaluate our method on problems with a difficult output space, where the model could no longer fit directly from data, and purposefully use simple MLPs for evaluation.

We begin with the problem of finding the shortest path in a graph. Specifically, we use a 4-by-4 grid $G = (V, E)$, and randomly remove some edges for each example to increase difficulty. Formally, our input is a binary vector of length $|V| + |E|$, with the first $|V|$ variables indicating sources and destinations, and the next $|E|$ which edges are removed. Similarly, the label for a given example is a binary vector of length $|E|$ indicating which edges are in the shortest path. Finally, we require through our semantic loss that the output form a valid simple path between the desired source and destination. This is achieved by using the appropriate logical constraint, as specified in [27].

We also examine the problem of predicting a complete order of user preferences. That is, for a given set of user features, we would like to predict how the user would rank their preference over a fixed set of items. We encode a preference ordering over $n$ items as a flattened binary matrix $\{X_{ij}\}$, where for each $i, j \in \{1, \dots, n\}$, it denotes that item $i$ is at position $j$ [4]. Clearly most outputs do not represent a total ordering, so we use semantic loss to enforce this. We predict rankings over 4 items, with data taken from PrefLib's sushi dataset [20].

Tables 4 and 5 report two different accuracies that illustrate the effect of semantic loss: "Coherent" indicates the percentage of examples for which the classifier gets the entire configuration right, while "Incoherent" measures the percentage of individually correct binary labels, which as a whole may not constitute a valid path/ranking at all. In the case of incoherent accuracy, semantic loss has little effect, and in fact slightly reduces the accuracy as it combats the standard sigmoid cross entropy. In regard to coherent accuracy, however, the semantic loss has a very large effect in guiding the network to jointly output structured objects, rather than optimizing each binary output individually. Remarkably, without semantic loss, the network is only able to output a valid preference ordering on $0.01\%$ of the test examples.

## 4 Conclusions

Both reasoning and semi-supervised learning are often identified as key challenges for deep learning going forward. In this paper, we developed a principled way of combining automated reasoning for propositional logic with existing deep learning architectures. Moreover, we showed that our semantic loss function provides significant benefits during semi-supervised classification, as well as deep structured prediction for highly complex output spaces.

**Acknowledgements**

This research was conducted while Zilu Zhang was a visiting student from Peking University. The authors thank Arthur Choi and Yujia Shen for helpful discussions. This work is partially supported by NSF grants #IIS-1657613, #IIS-1633857 and DARPA XAI grant #N66001-17-2-4032.

## References

[1] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[2] M.-W. Chang, D. Goldwasser, D. Roth, and Y. Tu. Unsupervised constraint driven learning for transliteration discovery. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 299–307. Association for Computational Linguistics, 2009.

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

[4] A. Choi, G. Van den Broeck, and A. Darwiche. Tractable learning for structured probability spaces: A case study in learning preference distributions. In *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[5] A. Darwiche. A differential approach to inference in bayesian networks. *J. ACM*, 50(3):280–305, May 2003.

[6] A. Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 819, 2011.

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

[8] H. Daumé, J. Langford, and D. Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.

[9] T. Demeester, T. Rocktäschel, and S. Riedel. Lifted rule injection for relation embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1389–1399, 2016.

[10] M. Diligenti, M. Gori, and C. Sacca. Semantic-based regularization for learning and inference. *Artificial Intelligence*, 244:143–165, 2017.

[11] I. Donadello, L. Serafini, and A. d. Garcez. Logic tensor networks for semantic image interpretation. 2017.

[12] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[13] A. Erkan and Y. Altun. Semi-supervised learning via generalized maximum entropy. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, volume PMLR, pages 209–216, 2010.

[14] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[15] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[16] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing. Harnessing deep neural networks with logic rules. In *ACL*, 2016.

[17] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014.

[18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[19] P. Márquez-Neila, M. Salzmann, and P. Fua. Imposing hard constraints on deep networks: Promises and limitations. *arXiv preprint arXiv:1706.02025*, 2017.

[20] N. Mattei and T. Walsh. Preflib: A library of preference data HTTP://PREFLIB.ORG. In *Proceedings of ADT*, 2013.

[21] P. Minervini, T. Demeester, T. Rocktäschel, and S. Riedel. Adversarial sets for regularising neural link predictors. *arXiv preprint arXiv:1707.07596*, 2017.

[22] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. In *ICLR*, 2016.

[23] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *ArXiv e-prints*, 2017.

[24] J. Naradowsky and S. Riedel. Modeling exclusion with a differentiable factor graph constraint. 2017.

[25] A. Neelakantan, B. Roth, and A. McCallum. Compositional vector space models for knowledge base inference. 2015.

[26] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pages 2014–2023, 2016.

[27] M. Nishino, N. Yasuda, S. Minato, and M. Nagata. Compiling graph substructures into sentential decision diagrams. In *Proceedings of the Thirty-First Conference on Artificial Intelligence (AAAI)*, 2017.

[28] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1796–1804, 2015.

[29] N. Pitelis, C. Russell, and L. Agapito. Semi-supervised learning using an unsupervised atlas. In *Proceedings of ECML-PKDD*, pages 565–580. Springer, 2014.

[30] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[31] S. Reed and N. De Freitas. Neural programmer-interpreters. *arXiv preprint arXiv:1511.06279*, 2015.

[32] S. Riedel, M. Bosnjak, and T. Rocktäschel. Programming with a differentiable forth interpreter. *CoRR, abs/1605.06640*, 2016.

[33] T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *HLT-NAACL*, pages 1119–1129, 2015.

[34] T. Sang, P. Beame, and H. A. Kautz. Performing bayesian inference by weighted model counting. In *AAAI*, volume 5, pages 475–481, 2005.

[35] R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, pages 2576–2582, 2017.

[36] M. Wang, Y. Tang, J. Wang, and J. Deng. Premise selection for theorem proving by deep graph embedding. *arXiv preprint arXiv:1709.09994*, 2017.

[37] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[38] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[39] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. *CoRR*, abs/1711.11157, Nov. 2017.