Using Commonsense to Guide Dialog Structure Induction via Neural Probabilistic Soft Logic

Anonymous ACL submission

Abstract

Latent Structure Induction from task-oriented dialogs would be made more robust and data-efficient by injecting symbolic knowledge into a neural learning process. We introduce *Neural Probabilistic Soft Logic Dialogue Structure Induction* (NEUPSL DSI), a general and principled approach that injects the symbolic knowledge into the latent space of a neural generative model via the *Probablistic Soft Logic*(PSL) formalism and allows for end-to-end gradient training. We conduct a thorough empirical investigation on the effect of NEUPSL DSI learning on the representation quality, few-shot learning, and out-of-domain generalization performance of the neural network. Over three simulated and real-world dialog structure induction benchmarks and across both unsupervised and semi-supervised settings for standard and cross-domain generalization, the injection of symbolic knowledge using NEUPSL DSI in unsupervised and semi-supervised settings provides a consistent boost in performance over the canonical baselines.

1 Introduction

The seamless integration of commonsense prior knowledge into the neural learning of language structure has been an open challenge in the machine learning and natural language processing communities. In this work, we inject commonsense symbolic knowledge into the neural learning process of a two-party *dialog structure induction* (DSI) task (Zhai and Williams, 2014; Shi et al., 2019). This tasks aims to learn a graph, known as *dialog structure*, capturing the potential flow of states occurring in a dialog dataset for a specific task-oriented domain, e.g. Figure 1 represents a potential dialog structure for the goal-oriented task of booking a hotel. Nodes in the dialog structure represent conversational topics or *dialog acts* that abstract the intent of individual utterances and edges represent transitions between dialog acts over successive turns of the dialog.

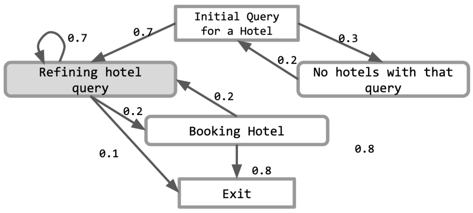

Figure 1: Example dialog structure for the goal-oriented task booking a hotel.

Traditionally, the dialog structure is hand-crafted by human domain experts. This process is both labor-intensive, and in most situations does not generalize easily to new domains. There has been previous work using supervised methods to learn this dialog structure from labeled data, starting from (Jurafsky, 1997). However, since structure annotation is expensive and subject to low-rater agreements, supervised methods are constrained by the small size of training data and the low label quality (Zhai and Williams, 2014). On the other hand, there has been work that attempts to perform DSI in an unsupervised fashion, e.g., *hidden Markov models* (Chotimongkol, 2008; Ian Ritter et al., 2010; Zhai and Williams, 2014) and more recently *Variational Recurrent Neural Networks* (VRNN) (Chung et al., 2015; Shi et al., 2019). However, these approaches are purely data-driven, have difficulty when the amount of data is limited or noisy, and cannot easily exploit both domain-specific and domain-independent dialog rules that are readily available from human experts.

In this work, we propose *Neural Probabilistic Soft Logic Dialogue Structure Induction* (NEUPSL DSI), a practical neuro-symbolic approach that improves the quality of learned dialog structure by infusing commonsense dialog knowledge into the end-to-end, gradient-based learning of a neural model. We leverage *Probabilistic Soft Logic* (PSL), a well-studied soft logic formalism, to express common-sense dialog rules in succinct and interpretable first-order logic statements that can be incoropraated easily into differentiable learning

(Bach et al., 2017; Pryor et al., 2022), leading to a simple method for common-sense knowledge injection with no change to the SGD-based training pipeline of an existing neural generative model.

Our key contributions are: 1) we propose NEUPSL DSI, a general and extendable latent dialog structure learning framework leveraging the probabilistic soft logic (PSL) formalism. NEUPSL DSI comes with novel smooth relaxation of PSL tailored to ensure rich gradient signal during back-propagation, which is important for achieving good empirical performance under SGD-based neuro-symbolic learning; 2) we evaluate NEUPSL DSI over both synthetic and realistic dialog datasets and under three evaluation protocols: standard generalization, domain generalization and domain adaptation, showing quantitatively that injecting common-sense reasoning provides a boost over unsupervised and few-shot methods, and 3) we comprehensively investigate the effect of soft logic-augmented learning on different aspects of the learned neural model, by examining its quality in representation learning, and performances in few-shot learning and structure induction.

2 Related Work

Dialog Structure Induction (DSI) refers to the task of inferring latent states of a dialog without complete supervision of the state labels. Earlier work focus on building advanced clustering methods, e.g., topic models, HMM, GMM (Zhai and Williams, 2014), which are later combined with pre-trained or task-specific neural representations (Nath and Kubba, 2021; Lv et al., 2021; Qiu et al., 2022). Another stream of research focuses on infering latent states using neural generative models, most notably *Direct-Discrete Variational Recurrent Neural Networks* (DD-VRNN) (Shi et al., 2019), with later improvements including BERT encoder (Chen et al., 2021), GNN-based latent-space model (Sun et al., 2021; Xu et al., 2021), structured-attention decoder(Qiu et al., 2020), and database query modeling (Hudeček and Dušek, 2022). Finally, Zhang et al. (2020); Wu et al. (2020) explored DSI in semi-supervised and few-shot learning context. No work to date have explored DSI with common-sense supervision, or conducts a comprehensive evaluation of model performance across different generalization settings (i.e., unsupervised, few-shot, domain generalization and domain adaptation).

A related field of work, Neuro-Symbolic computing (NeSy), is an active area of research that aims to incorporate logic-based reasoning with neural computation. This field contains a plethora of different neural symbolic methods and techniques. The methods that closely relate to our line of work seek to enforce constraints on the output of a neural network (Hu et al., 2016; Donadello et al., 2017; Diligenti et al., 2017; Mehta et al., 2018; Xu et al., 2018; Nandwani et al., 2019). For a more in-depth introduction, we refer the reader to these excellent recent surveys: Besold et al. (2017) and De Raedt et al. (2020). These methods although powerful are either: specific to the domain they work in, do not use the same soft logic formulation, have not been designed for unsupervised systems, or have not been used for dialog structure induction.

Finally, our method is most closely related to the novel NeSy approaches of *Neural Probabilistic Soft Logic* (NeuPSL) (Pryor et al., 2022), *DeepProbLog* (DPL) (Manhaeve et al., 2021), and *Logic Tensor Networks* (LTNs) (Badreddine et al., 2022). LTNs instantiates a model which forwards neural network predictions into functions representing symbolic relations with real-valued or fuzzy logic semantics, while DeepProbLog uses the output of a neural network to specify probabilities of events. The mathematical formulation of LTNs and DPL differ from our underlying soft logic distribution. NeuPSL unites state-of-the-art symbolic reasoning with the low-level perception of deep neural networks through a Probabilistic Soft Logic (PSL). Our method uses a NeuPSL formulation, however, we introduce a novel variation to the soft logic formulation, develop theory for unsupervised tasks, introduce the whole system in Tensorflow, and apply it to dialog structure induction.

3 Background

Our neuro-symbolic approach to dialog structure induction combines the principled formulation of probabilistic soft logic (PSL) rules with a neural generative model. In this work, we take the widely-used Direct-Discrete Variational Recurrent Neural Network (DD-VRNN) as an case study (Shi et al., 2019). We here introduce the necessary syntax and semantics for both the DD-VRNN and PSL.

3.1 Direct Discrete Variational Recurrent Neural Networks

A Direct Discrete Variational Recurrent Neural Networks (DD-VRNN) (Shi et al., 2019) is a proposed expansion to the popular Variational Recurrent Neural Network (VRNN) (Chung et al., 2015), which constucts a sequence of VAEs and associates them with the states of an RNN. The main difference between the DD-VRNN and a traditional VRNN is the priors of the latent states z_t. Here, the prior z_t depends on the previous prior z_{t-1}, which models the transitions between different latent (i.e. dialog) states. Formally, z_t is modeled as:

$$z_t \sim softmax(\phi_\tau^{prior}(z_{t-1})) \tag{1}$$

To fit the prior into the variational inference framework, an approximation of $p(z_t|x_{<t}, z_{<t})$ is made that changes the distribution to $p(z_t|z_{t-1})$ and thus:

$$p(x_{\le T}, z_{\le T}) \approx \prod_{t=1}^{T} p(x_t|z_{\le t}, x_{<t})p(z_t|z_{t-1})$$

Lastly, the objective function used in the DD-VRNN is a timestep-wise variational lower bound (Chung et al., 2015) augmented with a bag-of-word (BOW) loss and Batch Prior Regularization (BPR) (Zhao et al., 2017, 2018), i.e.:

$$\mathcal{L}_{VRNN} = \mathbb{E}_{q(z_{\le T}|x_{\le T})}[\log p(x_t|z_{\le t}, x_{<t}) + \sum_{t=1}^{T} -KL(q(z_t|x_{x \le t}, z_{<t})||p(z_t|x_{<t}, z_{<t}))],$$

so that the full objective function is

$$\mathcal{L}_{DD-VRNN} = \mathcal{L}_{VRNN} + \lambda * \mathcal{L}_{bow} \tag{2}$$

where λ is a tunable weight and $\mathcal{L}_{bow}$ is the BOW loss. For further details on $\mathcal{L}_{bow}$ see Section 4.3 and Shi et al. (2019). Additionally, to expand this to a semi-supervised domain, the objective function is augmented as:

$$\mathcal{L}_{DD-VRNN} = \mathcal{L}_{VRNN} + \lambda * \mathcal{L}_{bow} + \mathcal{L}_{supervised}$$

where $\mathcal{L}_{supervised}$ is the loss between the labels and predictions, e.g., *cross-entropy*.

3.2 Probabilistic Soft Logic

In this work we introduce soft constraints in a declarative fashion, similar to that of Probabilistic Soft Logic (PSL). PSL is a declarative statistical relational learning (SRL) framework for defining a particular graphical model, known as a *hinge-loss Markov random field* (HL-MRF) (Bach et al., 2017). More formally, PSL models relational dependencies and structural constraints using first-order logical rules, referred to as *templates* with arguments known as *atoms*. For example, the statement of "first utterance in a dialog is likely to belong to the `greet` state" can be expressed as:

$$\textsc{FirstUtt}(\mathtt{U}) \rightarrow \textsc{State}(\mathtt{U}, greet) \tag{3}$$

where ($\textsc{FirstUtt}(\mathtt{U})$, $\textsc{State}(\mathtt{U}, greet)$) are the *atoms* (i.e., atomic boolean statements) indicating, respectively, whether an utterance `U` is the first utterance of the dialog, or if it belongs to the state `greet`.

The *Probabilistic Soft Logic* (PSL) formalism (Bach et al., 2017) allows model to learn with soft logic constraints by allowing the originally Boolean-valued atoms to take continuous truth values that lie in the interval $[0, 1]$. Using this relaxation, PSL replaces logical operations with a form of soft logic termed *Lukasiewicz* logic (Klir and Yuan, 1995):

$$A \wedge B = max(0.0, A{+}B - 1.0)$$
$$A \vee B = min(1.0, A{+}B)$$
$$\neg A = 1.0 - A$$

where A and B are either ground atoms or logical expressions over atoms. In either case, they have values between [0,1]. For example, PSL will convert the statement from Equation 3, into the following:

$$min\{1.0,\ (1.0 - \textsc{FirstUtt}(\mathtt{U})) + \textsc{State}(\mathtt{U}, greet))\} \tag{4}$$

since $A \rightarrow B \equiv \neg A \vee B$. In this way, we can create a collection of functions $\{\ell_i\}_{i=1}^m$ that maps data to $[0, 1]$, known as *templates*. Note, this classic Lukasiewicz relaxation in fact leads to issues in gradient-based neural learning, due to its suboptimal gradient behavior. In Section 4.2, we discuss this in detail and propose a novel relaxation that is more suitable for gradient-based neural learning.

Using the templates, PSL defines a conditional probability density function over the unobserved random variables $\mathbf{y}$ given the observed data $\mathbf{x}$ known as the *Hinge-Loss Markov Random Field* (HL-MRF):

$$P(\mathbf{y}|\mathbf{x}) \propto exp(-\sum_{i=1}^{m} w_i * \phi_i(\mathbf{y}, \mathbf{x})) \tag{5}$$

Here w_i a non-negative weight and ϕ_i a *potential function* based on the templates:

$$\phi_i(\mathbf{y}, \mathbf{x}) = max\{0, \ell_i(\mathbf{y}, \mathbf{x})\} \tag{6}$$

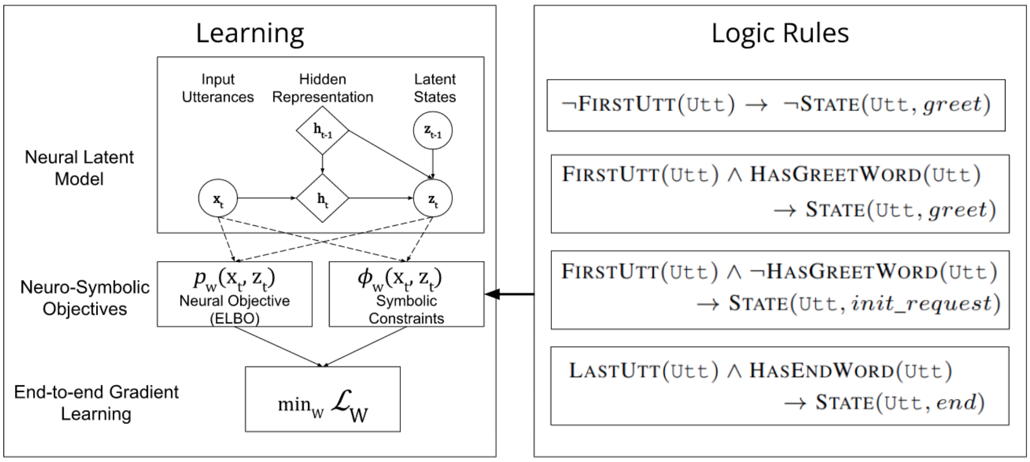

Figure 2: High-level pipeline of the NEUPSL DSI learning procedure.

Then the inference for the model predictions $\mathbf{y}$ coventionally proceeds by *maximum a posterior* (MAP) estimation, i.e., by maximizing the objective function $P(\mathbf{y}|\mathbf{x})$ (eq. 5) with respect to $\mathbf{y}$.

4 Neural Probabilistic Soft Logic Dialogue Structure Induction

In this section, we describe our approach for integrating common sense reasoning and neural network-based dialog structure induction. Our approach integrates an unsupervised neural generative model with commonsense dialog rules using soft constraints. We refer to our approach as *Neural Probabilistic Soft Logic Dialogue Structure Induction* (NEUPSL DSI). In the following, we first define the dialog structure learning problem, describe how to integrate the neural and symbolic losses, and then highlight important model components that are key to address optimization and representation-learning challenges under gradient-based neuro-symbolic learning.

Problem Formulation Given a goal-oriented dialog corpus $\mathcal{U} = \{\mathcal{D}_i\}_{i=1}^N$, we consider the DSI problem of learning a graph G underlying the corpus. More formally, dialog structure is defined as a directed graph $G = (S, P)$, where $S = \{s_1, \ldots, s_m\}$ encodes a set of dialog states, and P a probability distribution $p(s_t|s_{<t})$ representing the likelihood of transition between states (see Figure 1 for an example). Given the underlying dialog structure G, a dialog $d_i = \{x_1, \ldots, x_T\} \in \mathcal{D}$ is a temporally-ordered set of utterances x_t. Here, x_t's are generated according to an utterance distribution conditional on past history $p(x_t|s_{\leq t}, x_{<t})$, and the state s_t is generated according to $p(s_t|s_{<t})$. Given a dialog corpus $\mathcal{D} = \{d_i\}_{i=1}^n$, the task of DSI is to learn a directed graphical model $G = (S, P)$ as close to the underlying graph as possible.

4.1 Integrating Neural and Symbolic Learning under NEUPSL DSI

We now introduce how the NEUPSL DSI approach formally integrates the DD-VRNN with the soft symbolic constraints to allow for end-to-end gradient training. To begin, we define the relaxation of the symbolic constraints to be the same as described in Section 3.2. With this relaxation, we can build upon the foundations developed by Pryor et al. (2022) on Neural Probabilistic Soft Logic (NeuPSL), by augmenting the standard unsupervised DD-VRNN loss with a constraint loss. Figure 2 provides a graphical representation of this integration of the DD-VRNN and the symbolic constraints. Intuitively, NEUPSL DSI can be described in three parts: instantiation, inference, and learning.

In the instantiation process of the NEUPSL DSI model, a set of first-order templates, combined with a set of random variables creates a set of potentials that define a loss used for learning and evaluation. Let $p_{\mathbf{w}}$ be the DD-VRNN's predictive function of latent states with hidden parameters $\mathbf{w}$ and input utterances $\mathbf{x}$. The output of this function, defined as $p_{\mathbf{w}}(\mathbf{x})$, will be the probability distribution representing the likelihood of each latent class for a given utterance (Equation 1). Given a first-order symbolic rule $\ell_i(\mathbf{y}, \mathbf{x})$ where the decision variable $\mathbf{y} = p_{\mathbf{w}}(\mathbf{x})$ is the latent state prediction from the neural model $p_{\mathbf{w}}(\mathbf{x})$, we can instantiate a set of **deep hinge-loss potentials** of the form:

$$\phi_{\mathbf{w},i}(\mathbf{x}) = \max(0, \ell_i(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}))$$

For example, in reference to the example in Equation 4, the decision variable $\mathbf{y} = p_{\mathbf{w}}(\mathbf{x})$ is associated with the STATE($\mathbf{x}$, *greet*) random variables, leading to

$$\ell_i(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}) = \\ min\{1.0, (1.0 - \textsc{FirstUtt}(\mathbf{x})) + p_{\mathbf{w}}(\mathbf{x})\}.$$

With the instantiated model described above, the NeuPSL DSI inference objective is broken into a *neural inference* objective and a *symbolic inference* objective. The neural inference objective is computed by evaluating the the DD-VRNN model predictions with respect to the standard loss function for DSI. Given the deep hinge-loss potentials $\{\phi_{\mathbf{w},i}\}_{i=1}^m$, the symbolic inference objective is the HL-MRF likelihood (Equation 5) evaluated at the decision variables $\mathbf{y} = p_{\mathbf{w}}(x)$:

$$P_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = exp\big(- \sum_{i=1}^{m} w_i * \phi_{\mathbf{w},i}(\mathbf{x})\big) \quad (7)$$

Under the NeuPSL DSI, the decision variables $\mathbf{y} = p_{\mathbf{w}}(x)$ are implicitly controlled by neural network weights $\mathbf{w}$, therefore the conventional MAP inference in symbolic learning for decision variables $\mathbf{y}^* = \arg\min_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ can be done simply via neural weight minimization $\arg\min_{\mathbf{w}} P_{\mathbf{w}}(\mathbf{y}|\mathbf{x})$. As a result, NeuPSL DSI learning minimizes a constrained optimization objective:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \Big[\mathcal{L}_{DD-VRNN} + \lambda * \mathcal{L}_{constraint}\Big]$$

where we define the constraint loss to be the log likelihood of the HL-MRF distribution (7):

$$\mathcal{L}_{Constraint} = -logP_{\mathbf{w}}(\mathbf{y}|\mathbf{x}).$$

4.2 Improving soft logic constraints for gradient learning

The straightforward linear soft constraints used by the classic Lukasiewicz relaxation fails to pass back gradients with a magnitude and instead passes back a direction (e.g. ± 1). Formally, the gradient of a potential $\phi_{\mathbf{w}}(\mathbf{x}) = \max(0, \ell(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}))$ with respect to $\mathbf{w}$ is:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}}\phi_{\mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\ell(p_{\mathbf{w}}, \mathbf{x}) \cdot 1_{\phi_{\mathbf{w}}>0} \\ &= \Big[\frac{\partial}{\partial p_{\mathbf{w}}}\ell(p_{\mathbf{w}}, \mathbf{x})\Big] \cdot \frac{\partial}{\partial \mathbf{w}}p_{\mathbf{w}} \cdot 1_{\phi_{\mathbf{w}}>0}\end{aligned}$$

Here $\ell(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}) = a \cdot p_{\mathbf{w}}(\mathbf{x}) + b$ where $a, b \in \mathbb{R}$ and $p_{\mathbf{w}}(\mathbf{x}) \in [0, 1]$, which leads to the gradient $\frac{\partial}{\partial p_{\mathbf{w}}}\ell(p_{\mathbf{w}}, \mathbf{x}) = a$. Observing the three Lukasiewicz operations described in Section 3.2 it is clear that a will always result in ± 1, unless there are multiple $p_{\mathbf{w}}(\mathbf{x})$ per constraint.

As a result, this classic soft relaxation leads to a naive, non-smooth gradient:

$$\frac{\partial}{\partial \mathbf{w}}\phi_{\mathbf{w}} = \Big[a1_{\phi_{\mathbf{w}}>0}\Big] \cdot \frac{\partial}{\partial \mathbf{w}}p_{\mathbf{w}} \quad (8)$$

that is mostly consists of the predictive probability gradient $\frac{\partial}{\partial \mathbf{w}}p_{\mathbf{w}}$. It barely informs the model of the degree to which $p_{\mathbf{w}}$ satisfies the symbolic constraint $\phi_{\mathbf{w}}$ (other than the non-smooth step function $1_{\phi_{\mathbf{w}}>0}$), thereby creating challenges in gradient-based learning.

In this work, we propose a novel log-based relaxation that provides smoother and more informative gradient information for the symbolic constraints:

$$\psi_{\mathbf{w}}(\mathbf{x}) = \log\big(\phi_{\mathbf{w}}(\mathbf{x})\big) = \log\big(\max(0, \ell(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}))\big).$$

This seemingly simple transformation brings a non-trivial change to the gradient behavior:

$$\frac{\partial}{\partial \mathbf{w}}\psi_{\mathbf{w}} = \frac{1}{\phi_{\mathbf{w}}(\mathbf{x})} \cdot \frac{\partial}{\partial \mathbf{w}}\phi_{\mathbf{w}} = \Big[\frac{a}{\phi_{\mathbf{w}}}1_{\phi_{\mathbf{w}}>0}\Big] \cdot \frac{\partial}{\partial \mathbf{w}}p_{\mathbf{w}},$$

As shown, the gradient from the symbolic constraint now contains a new term $\frac{1}{\phi_{\mathbf{w}}(\mathbf{x})}$. It informs the model of the degree to which the model prediction satisfies the symbolic constraint ℓ, so that it is no longer a discrete step function with respect to $\phi_{\mathbf{w}}$. As a result, when the satisfaction of a rule $\phi_{\mathbf{w}}$ is non-negative but low (i.e., uncertain), the gradient magnitude will be high, and when the satisfaction of the rule is high, the gradient magnitude will be low. In this way, the gradient of the symbolic constraint terms ϕ_i now guides the neural model to more efficiently focus on learning the challenging examples that don't strongly obey the existing symbolic rules. This leads to a more effective collaboration between the neural and the symbolic components during model learning, and empirically leads to improved generalization performance (Section 5).

4.3 Stronger control of posterior collapse via weighted bag of words

It is important to avoid a collapsed VRNN solution, where the model puts all of its predictions in just a handful of states. This problem has been referred to as the vanishing latent variable problem (Zhao et al., 2017). Zhao et al. (2017) address this by introducing a *bag-of-word (BOW) loss* to VRNN modeling which requires the decoder network to predict the bag-of-words in response x. They separate x into two variables: x_o (word order) and x_{bow}

(no word order), with the assumption that they are conditionally independent given z and c:

$$p(x, z|c) = p(x_o|z, c)p(x_{bow}|z, c)p(z|c).$$

Let f be the output of a multilayer perception with parameters z, x, where $f \in \mathbb{R}^V$ with V the vocabulary size. Then the BOW probability is defined as $\log p(x_{bow}|z, c) = \log \prod_{t=1}^{|x|} \frac{e^{f_{x_t}}}{\sum_j^V e^{f_j}}$, where $|x|$ is the length of x and x_t is the word index of the t_{th} word in x.

To impose stronger regularization against the posterior collapse, we make use of a tf-idf-based re-weighting scheme using the tf-idf weights computed from the training corpus. Intuitively, this reweighting scheme helps the model to focus on reconstructing the non-generic terms that are unique to each dialog states, which encourages the model to “pull" the sentences from different dialog states further apart in its representations space in order to better minimize the weighted BOW loss. In comparison, a model under the uniformly-weighted BOW loss may be distracted by reconstructing the high-prevalence common terms (e.g., "what is", "can I", "when") that are shared by all dialog states, and thus less effective in preventing the collapse of the latent representations between the different states. As a result, we specify the tf-idf weighted BOW probability as:

$$\log p(x_{bow}|z, c) = \log \prod_{t=1}^{|x|} \frac{w_{x_i} e^{f_{x_t}}}{\sum_j^V e^{f_j}},$$

where $w_{x_t} = \frac{(1-\alpha)}{N} + \alpha w'_{x_t}$, N is the corpus size, w'_{x_t} is the tf-idf word weight for the x_t index, and α is a hyperparameter. In Section 5 we explore how this alteration affects the performance and observe if the PSL constraints still provide a boost.

5 Experimental Evaluation

In this section, we evaluate the performance of our proposed NeuPSL DSI method over two synthetic and one real-world task-orientated dialog corpus. We evaluate dialog structure induction performance and provide an extensive ablation analysis over all data settings to demonstrate the effectiveness of the NeuPSL DSI method. We explore the following questions: Q1) How does the model performance change in an unsupervised setting when soft constraints are incorporated into the loss? Q2) When introducing few-shot labels to the DD-VRNN for training, do soft constraints provide a boost? Q3) How does the alteration to the soft logic constraints and the re-weighted bag-of-words loss effect performance?

5.1 Dataset, Constraints, and Metrics

We explore these questions over three goal-oriented dialog datasets: MultiWoZ 2.1 synthetic (Campagna et al., 2020), and two versions of the Schema Guided Dialog (SGD) dataset SGD-synthetic (where the utterance is generated by a template-based dialog simulator) and SGD-real (which replaces the machine-generated utterances of SGD-synthetic with its human-paraphrased counterparts) (Rastogi et al., 2020). For the SGD-real dataset, we evaluate over three unique data settings, *standard generalization* (train and test over the same domain), *domain generalization* (train and test over different domains), and *domain adaptation* (model train on (possibly labelled) data from training domain and unlabelled data from test domain, and tests on the evaluation data from test domain.) Exact details on how each synthetic dataset is created can be found in the Appendix.

In the synthetic MultiWoZ setting, we introduce a set of 11 structural domain agnostic dialog rules. An example of one of these rules can be seen in Equation 3. These rules are introduced to represent general facts about dialogs and show how a few domain agnostic rules designed by a human expert can drastically improve performance. For all other settings we introduce a single token-based dialog rule. This constraint incorporates the idea that states are likely to contain utterances with known tokens, e.g., utterances containing 'hello' are likely to belong to the greet state. This rule was designed to show the potential boost in performance a model can achieve from a singular source of simple prior information. It is important to note that these constraints, in terms of the optimization problem, are not required to be satisfied. This means the model can learn to harmonize conflicts between data and the constraints during the learning process (e.g., in semi-supervised settings). Appendix C contains further details.

We explore an experimental evaluation in both an unsupervised and highly constrained semi-supervised setting. For both the overall results and the ablation analysis, we use class balanced accuracy and adjusted mutual information (AMI) (see Appendix D.1 for detail).

Method	SGD			SGD Synthetic	MultiWoZ
	Standard Generalization	Domain Generalization	Domain Adaptation	Standard Generalization	Standard Generalization
DD-VRNN	0.448 ± 0.019	0.476 ± 0.029	0.514 ± 0.028	0.553 ± 0.017	0.451 ± 0.042
NeuPSL DSI	**0.539 ± 0.048**	**0.541 ± 0.036**	**0.559 ± 0.045**	**0.811 ± 0.005**	**0.618 ± 0.028**

Table 1: Test set performance on MultiWoZ Synthetic, SGD, and SGD Synthetic. All reported results are averaged over 10 splits. Highlighted in bold are the highest performing methods.

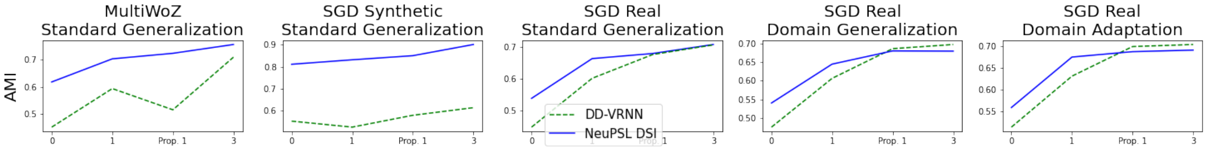

Figure 3: Average AMI for varying amount of supervision for MultiWoZ, SGD Synthetic, and SGD Real; Standard Generalization; Domain Generalization; Domain Adaptation.

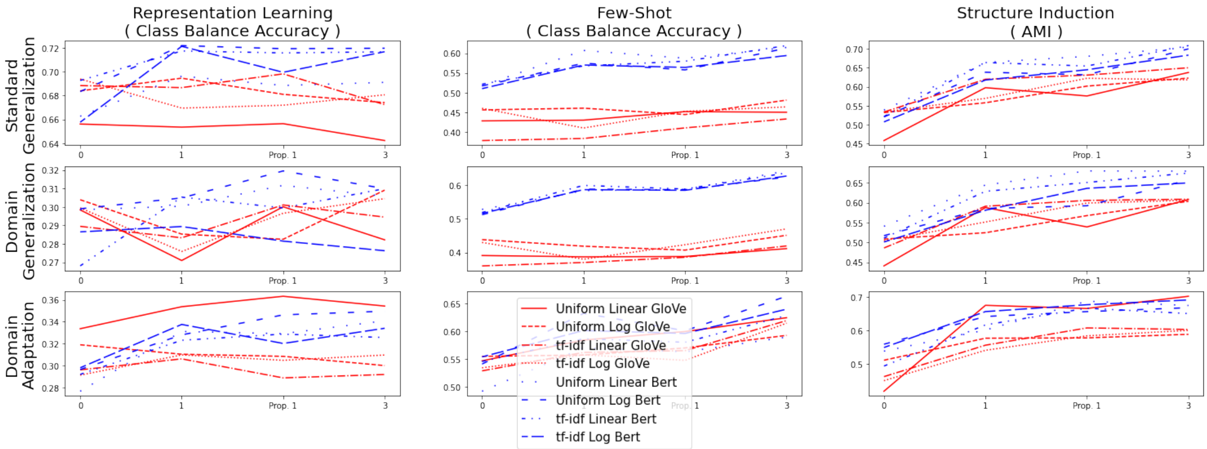

Figure 4: Average performance for representation learning, few-shot learning, and structure induction performance for the SGD dataset with varying amount of supervision.

5.2 Main results

Table 1 summarizes the main results of the NeuPSL DSI model compared to the *DD-VRNN* baseline, in a strictly unsupervised setting across all 5 dialog structure induction datasets. In comparison to the purely data driven DD-VRNN method, the NeuPSL DSI method outperforms all settings by over 4.0% in AMI. To reiterate, this performance improvement does not require additional supervision in the form of labels, but rather a few selected structural constraints. Additionally, comparing the NeuPSL DSI performance in the SGD standard generalization against the SGD domain generalization and SGD domain adaptation we see the AMI maintains its performance or improves. This trend indicates that the constraints do not hurt the generalizability of the neural model.

To further understand how these constraints affect the model we examine three highly constrained few shot settings: 1 shot, proportional 1 shot, and 3 shot. Both the 1 shot and 3 shot settings are randomly given one or three labels per class, while proportional 1 shot is given the same number of labels as the 1 shot setting but the distribution of labels are proportional to the class size. Any class below 1% will not be provided a label. Figure 3 summarizes the few shot results. In all settings the introduction of labels improves performance. This means the constraints do not overpower learning, rather it is a trade off between generalizing to these priors and learning over the labels. In the SGD settings, as the number of labels increase, the pure data driven approach is able to perform as well or better then NeuPSL DSI. This indicates that the token constraint hits a limit and the small decrease in performance is a notion of the bias-variance trade-off. However, the in the MultiWoZ setting, the domain agnostic dialog rules are able to maintain a performance improvement showing the simple constraints can boost a models performance without additional labeled data.

5.3 Ablation Study

In this section we provide an extensive ablation analysis over the SGD dataset where we examine when soft constraints provide a boost in per-

Bag-of-Words Weights	Constraint Loss	Embedding	Representation Learning (Class Balanced Accuracy)	Few-Shot Learning (Class Balanced Accuracy)	Structure Induction (AMI)
Uniform	Linear	Bert	0.588 ± 0.016	**0.517 ± 0.021**	**0.539 ± 0.048**
Uniform	Linear	GloVe	0.620 ± 0.023	0.428 ± 0.021	0.458 ± 0.024
Uniform	Log	Bert	0.600 ± 0.022	**0.517 ± 0.023**	0.520 ± 0.033
Uniform	Log	GloVe	**0.650 ± 0.011**	0.456 ± 0.014	0.532 ± 0.009
tf-idf	Linear	Bert	0.573 ± 0.022	**0.521 ± 0.018**	0.522 ± 0.024
tf-idf	Linear	GloVe	0.595 ± 0.014	0.379 ± 0.015	0.533 ± 0.048
tf-idf	Log	Bert	0.578 ± 0.021	**0.510 ± 0.022**	0.507 ± 0.060
tf-idf	Log	GloVe	**0.653 ± 0.014**	0.460 ± 0.009	0.534 ± 0.033

Table 2: Test set performance on SGD standard generalization data setting.

Bag-of-Words Weights	Constraint Loss	Embedding	Representation Learning (Class Balanced Accuracy)	Few-Shot Learning (Class Balanced Accuracy)	Structure Induction (AMI)
Uniform	Linear	Bert	**0.597 ± 0.018**	**0.528 ± 0.026**	**0.541 ± 0.036**
Uniform	Linear	GloVe	**0.597 ± 0.012**	0.391 ± 0.018	0.441 ± 0.030
Uniform	Log	Bert	**0.598 ± 0.032**	**0.512 ± 0.021**	**0.517 ± 0.036**
Uniform	Log	GloVe	**0.608 ± 0.014**	0.438 ± 0.017	**0.508 ± 0.006**
tf-idf	Linear	Bert	0.536 ± 0.026	**0.518 ± 0.034**	**0.511 ± 0.018**
tf-idf	Linear	GloVe	0.579 ± 0.033	0.360 ± 0.016	0.486 ± 0.057
tf-idf	Log	Bert	0.573 ± 0.018	**0.516 ± 0.035**	0.501 ± 0.064
tf-idf	Log	GloVe	**0.599 ± 0.025**	0.430 ± 0.020	**0.505 ± 0.005**

Table 3: Test set performance on SGD domain generalization data setting.

Bag-of-Words Weights	Constraint Loss	Embedding	Representation Learning (Class Balanced Accuracy)	Few-Shot Learning (Class Balanced Accuracy)	Structure Induction (AMI)
Uniform	Linear	Bert	0.554 ± 0.135	0.492 ± 0.124	**0.538 ± 0.107**
Uniform	Linear	GloVe	**0.667 ± 0.022**	**0.547 ± 0.025**	0.419 ± 0.073
Uniform	Log	Bert	0.593 ± 0.049	**0.541 ± 0.023**	**0.559 ± 0.045**
Uniform	Log	GloVe	0.638 ± 0.024	**0.555 ± 0.022**	0.511 ± 0.045
tf-idf	Linear	Bert	0.584 ± 0.035	**0.546 ± 0.023**	0.494 ± 0.033
tf-idf	Linear	GloVe	0.593 ± 0.039	0.529 ± 0.022	0.463 ± 0.041
tf-idf	Log	Bert	0.597 ± 0.034	**0.554 ± 0.025**	**0.549 ± 0.038**
tf-idf	Log	GloVe	0.583 ± 0.029	**0.534 ± 0.027**	0.451 ± 0.044

Table 4: Test set AMI and standard deviation on SGD domain adaptation data setting.

formance. An ablation analysis for MultiWoZ and SGD Synthetic is provided in the Appendix. Throughout this section we evaluate how each variation of the model performs over three aspects: 1) representation learning, 2) few-shot learning, and 3) structure induction. To evaluate the representation learning that the NeuPSL DSI method learns, we take the hidden representation of the learned model and train a fully supervised linear classifier to predict dialog acts. After training this linear classifier, we evaluate the averaged class balanced accuracy label performance. To evaluate the few-shot learning that the NeuPSL DSI method learns, we take the hidden representation of the learned model and train a semi-supervised linear classifier to predict dialog acts. We average the class-balanced accuracy of three few-shot settings: 1 shot, 5 shot, and 10 shot. Finally, structure induction performance is evaluated using AMI.

Table 2 (SGD standard), Table 3 (SGD domain generalization), and Table 4 (SGD domain adaptation) summarize the results for the SGD data setting for the unsupervised learning. Each of the tables report the three aspects for evaluation over eight different model settings; uniform / tf-idf bag-of-words weights, linear / log constraint loss, and BERT (Devlin et al., 2018) / GloVe (Pennington et al., 2014) embedding. All reported results are averaged over 10 splits. Highlighted in bold are the highest performing methods, or methods within the the standard deviation of the highest performing methods. In the unsupervised setting no method outshines all others completely. In general the GloVe embedding outperforms Bert in the representation learning, however, for structure induction and few-shot learning Bert typically outperforms its GloVe counterpart.

Figure 4 summarizes the few-shot training results for the SGD data settings when training with 1 shot, proportional 1 shot, and 3 shots. Interestingly we see three methods generally on top in performance: uniform-log-bert, tf-idf-linear-bert, and uniform-linear-bert. There seems to be no clear winner between uniform/tf-idf and linear/log, however, all three of these settings use BERT.

6 Conclusion

We study NeuPSL DSI, a principled learning framework to guide the neural dialog structure learning via symbolic knowledge. Thorough empirical investigation illustrates the concrete benefit of NeuPSL DSI learning on the representation quality, few-shot learning, and out-of-domain generalization performance of the neural network.

References

Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-loss Markov random fields and probabilistic soft logic. *JMLR*, 18(1):1–67.

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic tensor networks. *AI*, 303(4):103649.

Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv*.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Association for Computational Linguistics (ACL)*.

Bingkun Chen, Shaobing Dai, Shenghua Zheng, Lei Liao, and Yang Li. 2021. Dsbert: Unsupervised dialogue structure learning with bert. *arXiv preprint arXiv:2111.04933*.

Ananlada Chotimongkol. 2008. *Learning the Structure of Task-Oriented Conversations from the Corpus of In-Domain Dialogs*. Ph.D. thesis, Carnegie Mellon University, Institute for Language Technologies.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Neural Information Processing Systems (NeurIPS)*.

Luc De Raedt, Sebastijan Dumančić, Robin Manhaeve, and Giuseppe Marra. 2020. From statistical relational to neuro-symbolic artificial intelligence. In *IJCAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Michelangelo Diligenti, Soumali Roychowdhury, and Marco Gori. 2017. Integrating prior knowledge into deep learning. In *ICMLA*.

Ivan Donadello, Luciano Serafini, and Artur S. d'Avila Garcez. 2017. Logic tensor networks for semantic image interpretation. In *IJCAI*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Language Resources and Evaluation Conference (LREC)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 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.

Vojtěch Hudeček and Ondřej Dušek. 2022. Learning interpretable latent dialogue actions with less supervision. *arXiv preprint arXiv:2209.11128*.

Dan Jurafsky. 1997. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *Institute of Cognitive Science Technical Report*.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

George J. Klir and Bo Yuan. 1995. *Fuzzy Sets and Fuzzy Logic - Theory and Applications*. Prentice Hall.

lan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Association for Computational Linguistics (ACL)*.

Chenxu Lv, Hengtong Lu, Shuyu Lei, Huixing Jiang, Wei Wu, Caixia Yuan, and Xiaojie Wang. 2021. Task-oriented clustering for dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4338–4347.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2021. Neural probabilistic logic programming in DeepProbLog. *AI*, 298:103504.

Sanket Vaibhav Mehta, Jay Yoon Lee, and Jaime Carbonell. 2018. Towards semi-supervised learning for deep semantic role labeling. In *EMNLP*.

Yatin Nandwani, Abhishek Pathak, and Parag Singla. 2019. A primal dual formulation for deep learning with constraints. In *NeurIPS*.

Apurba Nath and Aayush Kubba. 2021. Tscan: Dialog structure discovery using scan, adaptation of scan to text data. *Engineering and Applied Sciences*, 6(5):82.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Connor Pryor, Charles Dickens, Eriq Augustine, Alon Albalak, William Wang, and Lise Getoor. 2022. Neupsl: Neural probabilistic soft logic.

Liang Qiu, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Structure extraction in task-oriented dialogues with slot clustering. *arXiv preprint arXiv:2203.00073*.

Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-chun Zhu. 2020. Structured attention for unsupervised dialogue structure induction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1889–1899.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Weiyan Shi, Tiancheng Zhao, and Zhou Yu. 2019. Unsupervised dialog structure learning. In *Association for Computational Linguistics (ACL)*.

Yajing Sun, Yong Shan, Chengguang Tang, Yue Hu, Yinpei Dai, Jing Yu, Jian Sun, Fei Huang, and Luo Si. 2021. Unsupervised learning of deterministic dialogue structure with edge-enhanced graph auto-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13869–13877.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research (JMLR)*, 11(95):2837–2854.

Chien-Sheng Wu, Steven CH Hoi, and Caiming Xiong. 2020. Improving limited labeled dialogue state tracking with self-supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4462–4472.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. 2018. A semantic loss function for deep learning with symbolic knowledge. In *ICML*.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. Discovering dialog structure graph for coherent dialog generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1726–1739.

Ke Zhai and Jason D. Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Association for Computational Linguistics (ACL)*.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskénazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Association for Computational Linguistics (ACL)*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Association for Computational Linguistics (ACL)*.

Token Constraint
w_1 : HASWORD(Utt, Class) $\rightarrow$ STATE(Utt, Class)

Figure 5: SGD Structure Induction Constraint Model

A Model Details

In this section we provide additional details on the NEUPSL DSI models for the Multi-WoZ and SGD settings. Throughout these subsections, we cover the symbolic constraints and the hyperparameters used. All unspecified values for either the constraints or the DD-VRNN model were left at their default values. Code will be released upon acceptance and is under the Apache 2.0 license.

A.1 SGD Constraints

The NEUPSL DSI model for all SGD settings (synthetic, standard, domain generalization, domain adaptation) uses a single constraint. Figure 5 provides an overview of the constraint which contains the following two predicates:

1. **STATE(`Utt`, `Class`)**
   The STATE continuous valued predicate is the probability that an utterance, identified by the argument `Utt`, belongs to a dialog state, identified by the argument `Class`. For instance the utterance *hello world !* for the *greet* dialog state would create a predicate with value between zero and one, i.e. STATE($hello\ world\ !, greet$) $= 0.7$.

2. **HASWORD(`Utt`, `Class`)**
   The HASWORD binary predicate indicates if an utterance, identified by the argument `Utt`, contains a known token for a particular class, identified by the argument `Class`. For instance if a known token associated with the *greet* class is *hello*, then the utterance *hello world !* would create a predicate with value one, i.e. HASWORD($hello\ world\ !, greet$) $= 1$.

This token constraint encodes the prior knowledge that utterances' are likely to belong to dialog states when an utterance contains tokens that represent that state. For example, if a known token associated with the *greet* class is *hello*, then the utterance *hello world !* is likely to belong to the *greet* state. The major purpose of incorporating this constraint into the model is to show how even a small amount of prior knowledge can aid predictions. To get the set of tokens associated with each state, we trained a supervised linear classifier where the input is an utterance and label is the class. After training, every token is then individually run through the trained model in order to get a set of logits over each class. These logits represent the relative importance that each token has over every class. Sparsity is introduced to this set of logits, leaving only the top 0.1% of values and replacing the others with zeros. This sparsity reduces the set of 261,651 logits to 262 non-zero logits.

A.2 Multi-WoZ Constraints

The NEUPSL DSI model for the Multi-WoZ setting uses a set of dialog constraints, which can be broken into dialog start, dialog middle, and dialog end. Figure 6 provides an overview of the constraints which contains the following 11 predicates:

1. **STATE(`Utt`, `Class`)**
   The STATE continuous valued predicate is the probability that an utterance, identified by the argument `Utt`, belongs to a dialog state, identified by the argument `Class`. For instance the utterance *hello world !* for the *greet* dialog state would create a predicate with value between zero and one, i.e. STATE($hello\ world\ !, greet$) $= 0.7$.

2. **FIRSTUTT(`Utt`)**
   The FIRSTUTT binary predicate indicates if an utterance, identified by the argument `Utt`, is the first utterance in a dialog.

3. **LASTUTT(`Utt`)**
   The LASTUTT binary predicate indicates if an utterance, identified by the argument `Utt`, is the last utterance in a dialog.

4. **PREVUTT(`Utt`)**
   The PREVUTT binary predicate indicates if an utterance, identified by the argument `Utt2`, is the previous utterance in a dialog of another utterance, identified by the argument `U1`.

Dialog Start
$w_1 : \neg\textsc{FirstUtt}(\texttt{Utt}) \rightarrow \neg\textsc{State}(\texttt{Utt}, \textit{greet})$
$w_2 : \textsc{FirstUtt}(\texttt{Utt}) \wedge \textsc{HasGreetWord}(\texttt{Utt}) \rightarrow \textsc{State}(\texttt{Utt}, \textit{greet})$
$w_3 : \textsc{FirstUtt}(\texttt{Utt}) \wedge \neg\textsc{HasGreetWord}(\texttt{Utt}) \rightarrow \textsc{State}(\texttt{Utt}, \textit{init\_request})$

Dialog Middle
$w_4 : \textsc{PrevUtt}(\texttt{Utt1}, \texttt{Utt2}) \wedge \textsc{State}(\texttt{Utt2}, \textit{greet}) \rightarrow \textsc{State}(\texttt{Utt1}, \textit{init\_request})$
$w_5 : \textsc{PrevUtt}(\texttt{Utt1}, \texttt{Utt2}) \wedge \neg\textsc{State}(\texttt{Utt2}, \textit{greet}) \rightarrow \neg\textsc{State}(\texttt{Utt1}, \textit{init\_request})$
$w_6 : \textsc{PrevUtt}(\texttt{Utt1}, \texttt{Utt2}) \wedge \textsc{State}(\texttt{Utt2}, \textit{init\_request}) \rightarrow \textsc{State}(\texttt{Utt1}, \textit{second\_request})$
$w_7 : \textsc{PrevUtt}(\texttt{Utt1}, \texttt{Utt2}) \wedge \textsc{State}(\texttt{Utt2}, \textit{second\_request}) \wedge \textsc{HasInfoQuestionWord}(\texttt{Utt1}) \rightarrow \textsc{State}(\texttt{Utt1}, \textit{info\_question})$
$w_8 : \textsc{PrevUtt}(\texttt{Utt1}, \texttt{Utt2}) \wedge \textsc{State}(\texttt{Utt2}, \textit{second\_request}) \wedge \textsc{HasSlotQuestionWord}(\texttt{Utt1}) \rightarrow \textsc{State}(\texttt{Utt1}, \textit{slot\_question})$
$w_9 : \textsc{PrevUtt}(\texttt{Utt1}, \texttt{Utt2}) \wedge \textsc{State}(\texttt{Utt2}, \textit{end}) \wedge \textsc{HasCancelWord}(\texttt{Utt1}) \rightarrow \textsc{State}(\texttt{Utt1}, \textit{cancel})$

Dialog End
$w_{10} : \textsc{LastUtt}(\texttt{Utt}) \wedge \textsc{HasEndWord}(\texttt{Utt}) \rightarrow \textsc{State}(\texttt{Utt}, \textit{end})$
$w_{11} : \textsc{LastUtt}(\texttt{Utt}) \wedge \textsc{HasAcceptWord}(\texttt{Utt}) \rightarrow \textsc{State}(\texttt{Utt}, \textit{accept})$
$w_{12} : \textsc{LastUtt}(\texttt{Utt}) \wedge \textsc{HasInsistWord}(\texttt{Utt}) \rightarrow \textsc{State}(\texttt{Utt}, \textit{insist})$

Figure 6: MultiWoZ Structure Induction Constraint Model

5. **HASGREETWORD(Utt)**
   The HASGREETWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the greet class. The list of known greet words are $['hello', 'hi']$.

6. **HASINFOQUESTIONWORD(Utt)**
   The HASINFOQUESTIONWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the info question class. The list of known info question words are $['address', 'phone']$.

7. **HASSLOTQUESTIONWORD(Utt)**
   The HASSLOTQUESTIONWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the slot question class. The list of known slot question words are $['what', '?']$.

8. **HASINSISTWORD(Utt)**
   The HASINSISTWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the insist class. The list of known insist words are $['sure', 'no']$.

9. **HASCANCELWORD(Utt)**
   The HASCANCELWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the cancel class. The list of known cancel words are $['no']$.

10. **HASACCEPTWORD(Utt)**
    The HASACCEPTWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the accept class. The list of known accept words are $['yes', 'great']$.

11. **HASENDWORD(Utt)**
    The HASENDWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the end class. The list of known end words are $['thank', 'thanks']$.

The dialog start constraints take advantage of the inherent structure built into the beginning of task-oriented dialogs. In the same order as the dialog start rules in Figure 6: 1) If the first turn utterance does not contain a known greet word, then it does not belong to the *greet* state. 2) If the first turn utterance contains a known greet word, then it belong to the *greet* state. 3) If the first turn utterance does not contain a known greet word, then it belongs to the *initial request* state.

The dialog middle constraints exploit the temporal dependencies within the middle of a dialog. In the same order as the dialog middle rules in Figure 6: 1) If the previous utterance belongs to the *greet* state, then the current utterance belongs to the *initial request* state. 2) If the previous utterance does not belong to the *greet* state, then the current utterance does not belong to the *initial request* state. 3) If the previous utterance belongs to the *initial request* state, then the current utterance belongs to the $second\ request$ state. 4) If the previous utterance belongs to the $second\ request$ state and it has a known info question token, then the current utterance belongs to the $info\ question$ state. 5) If the previous utterance belongs to the $second\ request$ state and it has a known slot question token, then the current utterance belongs to the $slot\ question$ state. 4) If the previous utterance belongs to the end state and it has a known cancel token, then the current utterance belongs to the $cancel$ state.

The dialog end constraints take advantage of the inherent structure built into the end of task-oriented dialogs. In the same order as the dialog end rules in Figure 6: 1) If the last turn utterance contains a known end word, then it belongs to the end state. 2) If the last turn utterance contains a known accept word, then it belong to the $accept$ state. 3) If the last turn utterance contains a known insist word, then it belong to the $insist$ state.

B Additional Model Details

B.1 Symbolic-rule Normalization in the Multi-class Setting

In the multi-class setting (e.g., multiple latent states), some soft logic operation on the model probability $p_{\mathbf{w}}$ will lead a probability that no longer normalize to 1. For example, the negation operation on the probability vector $p_{\mathbf{w}}$ will lead to $!p_{\mathbf{w}} = 1 - p_{\mathbf{w}}$; then in the multi-class setting, the norm of $!p_{\mathbf{w}}$ is $\sum_i^{|C|}(1-p_i) = |C|-1 > 1$, where $|C|$ is the number of classes. To address the above concern, we re-normalize after every soft logic operation:

$$f_{\mathbf{w}}(\mathbf{y}, \mathbf{x}) = f_{\mathbf{w}}(\mathbf{y}, \mathbf{x})/||f_{\mathbf{w}}(\mathbf{y}, \mathbf{x})||,$$

where $f_{\mathbf{w}}(\mathbf{y}, \mathbf{x})$ is the output of a soft logical operation.

B.2 Model Hyperparameters

The *DD-VRNN* uses an LSTM (Hochreiter and Schmidhuber, 1997) with 200-400 units for the RNNs, and fully-connected highly flexible feature extraction functions with a dropout of 0.4 for the input x, the latent vector z, the prior, the encoder and the decoder. The input to the *DD-VRNN* is the utterances with a 300-dimension word embedding created using a GloVe embedding (Pennington et al., 2014) and a Bert embedding (Devlin et al., 2019). The maximum utterance word length was set to 40, the maximum length of a dialog was set to 10, and the tunable weight, γ (Equation 2), was set to 0.1. The total number of parameters are 26,033,659 for the model with GloVe embedding and 135,368,227 with Bert embedding.

The experiments are run in Google TPU V4, and the total GPU hours for all finetuning are 326 GPU hours.

C Datasets

In this section we provide additional information on the SGD, SGD synthetic, and MultiWoZ 2.1 synthetic datasets.

C.1 SGD

The Schema-Guided Dialog (SGD) (Rastogi et al., 2020) is a task-oriented conversation dataset involving interactions with services and APIs covering 20 domains. There are overlapping functionalities over many of different APIs, but their interfaces are different. One conversion may involve multiple domains. Train set contains conversions from 16 domains, and 4 other domains are only present in dev or test sets.

In the experiment, we split the test set based on whether the example is from the 4 domains not present in the train set or not. This gives us 34,308 in-domain 5,441 out-of-domain test examples. To evaluate the generalization of the model, we evaluate the model performance on both test sets. In specific, we establish three different evaluation protocols.

- **SGD Standard Generalization** We train the model using SGD train set, evaluate on the in-domain test set.
- **SGD Domain Generalization** We train the model using SGD train set, evaluate on the out-of-domain test set.
- **SGD Domain Adaptation** We train the model using SGD train set and label-wiped in-domain and out-of-domain test sets, evaluate on out-of-domain test set.

C.2 SGD Synthetic

Using the template-based generator from the SGD developers Kale and Rastogi (2020), we generate 10,800 synthetic dialogs using the same APIs and dialog states as the official SGD data. We split the examples with 75% train and 25% test. The schema-guided generator code is under Apache 2.0

license: https://github.com/google-research/task-oriented-dialogue/blob/main/LICENSE.

C.3 MulitWoZ 2.1 Synthetic

MultiWoZ 2.1 synthetic (Campagna et al., 2020) is a multi-domain goal-oriented dataset covering five domains (Attraction, Hotel, Restaurant, Taxi, and Train) and nine dialog acts ($greet$, $initial\ request$, $second\ request$, $insist$, $info\ question$, $slot\ question$, $accept$, $cancel$, and end). Following Campagna et al. (2020), we generate 10^4 synthetic dialogs from a known ground-truth dialog structure. Figure 7 provides an overview of the ground truth dialog structure, which is based on the original MultiWoz 2.1 dataset (Eric et al., 2019), used through the generative process. These 10^4 synthetic dialogs are randomly sampled without replacement to create 10 splits with 80% train, 10% test, and 10% validation. The MultiWoZ 2.1 synthetic code is under the MIT License: https://github.com/stanford-oval/zero-shot-multiwoz-acl2020. The MultiWoZ 2.1 code uses genie which is under the MIT License: https://github.com/stanford-oval/genie-k8s/blob/master/LICENSE.

D Extended Experimental Evaluation

In this section we provide additional experimental results on the NEUPSL DSI models for all settings. We split the extended evaluation into additional main results, ablation results, and additional experiments. Details describing changes to the models are provided in each subsection.

D.1 Evaluation Metrics

Adjusted Mutual Information (AMI) - AMI evaluates dialog structure prediction by evaluating the correctness of the dialog state assignments. Let $U^* = \{U_1^*, \ldots, U_{C^*}^*\}$ be the ground-truth assignment of dialog states for all utterances in the corpus, and $U = \{U_1, \ldots, U_C\}$ be the predicted assignment of dialog states based on the learned dialog structure model. U^* and U are not directly comparable because they draw from different base sets of states ($U*$ from the ground truth set of states and U from the set of states induced by the DD-VRNN), that may even have different cardinalities. We address this problem by using Adjusted Mutual Information (AMI), a metric originally developed to compare unsupervised clustering algorithms. Intuitively, AMI treats each assignment as a probability distribution over states, and uses Mutual Information to measure their similarity, adjusting for the fact that larger clusters tend to have higher MI. AMI is defined as follows:

$$AMI(U, U^*) = \frac{MI(U, U^*) - \mathbb{E}(MI(U, U^*))}{Avg(H(U), H(U^*)) - \mathbb{E}(MI(U, U^*))}$$

where $MI(U, U^*)$ is the mutual information score, $\mathbb{E}(MI(U, U^*))$ is the expected mutual information over all possible assignments, and $Avg(H(U), H(U^*))$ is the average entropy of the two clusters (Vinh et al., 2010).

Purity . Let $U^* = \{U_1^*, \ldots, U_{C^*}^*\}$ be the ground-truth assignment of dialog states for all utterances in the corpus, and $U = \{U_1, \ldots, U_C\}$ be the predicted assignment of dialog states based on the learned dialog structure model. Each cluster is assigned to the class which is most frequent in the cluster. This assignment then calculates an accuracy summing together the total correct of each cluster and dividing by the total number of clusters. Purity is defined as follows:

$$Purity(U, U^*) = \frac{1}{N} \sum_{k=1}^{K} Count(U, U^*, A_k)$$

where K is the number of unique clusters predicted, N is the total number of predicted utterances, A_k is the most frequent underlying ground truth in cluster k, and $Count(U, U^*, A_k)$ is the total number of correctly labeled utterances within that assigned cluster.

D.2 Main Results

In this section we provide addition experimental results for the structure induction performance. To further understand how accurate the generated dialog structure is, we evaluate the NEUPSL DSI model and the *DD-VRNN* baselines on two additional evaluation metrics, class-balanced accuracy and purity.

Table 5 summarizes extended evalution of the main results for the NEUPSL DSI model and *DD-VRNN* baseline in a strictly unsupervised setting across all 5 dialog structure induction dataset. Note, these values correlate with the reported results in Table 1, i.e., these are not the best performing results but are other metrics for the same runs. The extended results follow a similar trend to the AMI

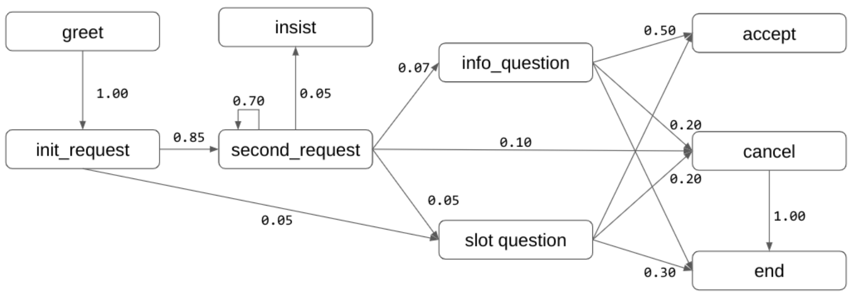

Figure 7: Ground truth dialog structure used to generate the MultiWoZ 2.1 dataset. Transition graph shows transitions over 0.05%.

Metric	Method	SGD Standard	SGD Domain Generalization	SGD Domain Adaptation	SGD Synthetic	MultiWoZ
Purity	DD-VRNN	0.341 ± 0.019	0.425 ± 0.016	**0.443 ± 0.015**	0.447 ± 0.024	0.701 ± 0.042
	NeuPSL DSI	**0.463 ± 0.039**	**0.468 ± 0.039**	0.425 ± 0.056	**0.810 ± 0.005**	**0.762 ± 0.015**
Class Balanced Accuracy	DD-VRNN	0.016 ± 0.012	0.018 ± 0.016	0.009 ± 0.009	0.020 ± 0.015	0.104 ± 0.076
	NeuPSL DSI	**0.125 ± 0.018**	**0.159 ± 0.021**	**0.146 ± 0.036**	**0.474 ± 0.005**	**0.625 ± 0.008**

Table 5: Test set performance on MultiWoZ Synthetic, SGD, and SGD Synthetic. These values correlate with the results reported in Table 1.

results. Surprisingly, we get over 60% class balanced accuracy in the MultiWoZ setting. This indicates that designing a set of domain agnostic common-sense structural rules can provide massive improvements to the models trained over purely token level prior information.

Additionally, we examine three highly constrained few shot settings: 1 shot, proportional 1 shot, and 3 shot. Both the 1 shot and 3 shot settings are randomly given one or three labels per class, while proportional 1 shot is given the same number of labels as the 1 shot setting but the distribution of labels are proportional to the class size. Anything below 1% will not be provided a label. Figure 8 summarizes the few shot results. Similar to the AMI, in all settings the introduction of labels improves performance. In the SGD real setting, we are seeing comparable performance, while the SGD synthetic and MulitWoZ settings see drastic improvements.

D.3 Ablation Analysis

In this section we provide an extensive ablation analysis over the SGD synthetic and MultiWoZ datasets, in which we examine when the constraints provide a boost in performance. Throughout this section, we evaluate how each variation performs over three aspects: 1) representation learning, 2) few-shot learning, and 3)structure induction. To evaluate the representation learning that the NeuPSL DSI method learns, we take the hidden representation of the learned model and train a fully supervised linear classifier with this representation. After training this linear classifier, we evaluate the averaged class balanced accuracy label performance. To evaluate the few-shot learning that the NeuPSL DSI method learns, we take the hidden representation of the learned model and train a semi-supervised linear classifier with this representation. We average the class-balanced accuracy of three few-shot settings: 1 shot, 5 shot, and 10 shot. Finally, to evaluate the structure induction performance, we evaluate the model's AMI.

Table 6 summarizes the unsupervised results for the MulitWoZ data setting. The results are reported over the three aspects for sixteen different model settings; uniform/tf-idf bag-of-words weights, linear/log constraint loss, standard/normalized constraints, and Bert/GloVe embedding. All reported results are averaged over 10 splits. Highlighted in bold are the highest performing methods, or methods within the standard deviation of the highest performing method.

Table 7 summarizes the unsupervised results for the SGD synthetic data setting. The results are reported over the three aspects for four different model settings; uniform/supervised bag-of-words weights, and linear/log constraint loss. Supervised bag-of-words weights use the weights of a fully trained linear classifier, as described in Appendix

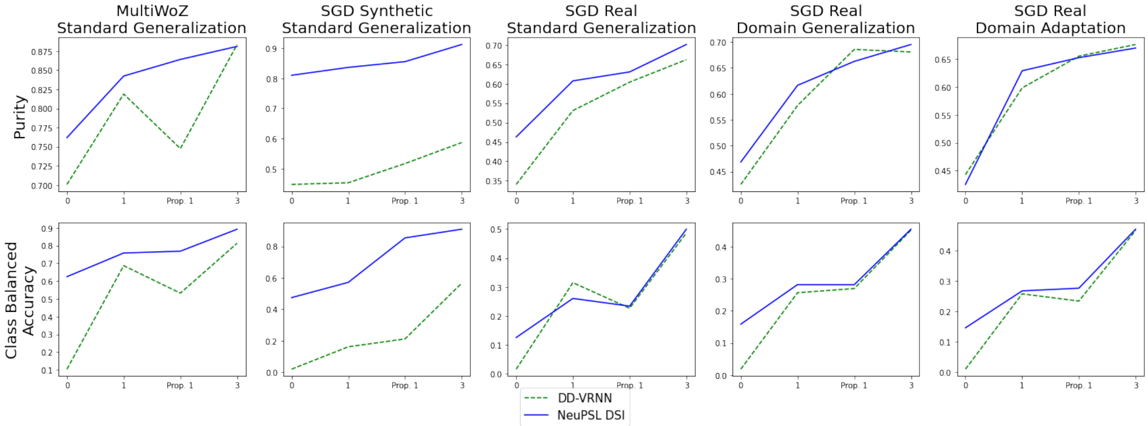

Figure 8: Average Purity and Class Balanced Accuracy on MultiWoZ Synthetic, SGD, and SGD Synthetic for varying amount of supervision. These values correlate with the results reported in Figure 3.

Bag-of-Words Weights	Constraint Loss	Constraints Normalized	Embedding	Representation Learning (Class Balanced Acc.)	Few-Shot Learning (Class Balanced Acc.)	Structure Induction (AMI)
Uniform	Linear	Standard	Bert	**0.941 ± 0.010**	0.667 ± 0.030	0.529 ± 0.040
Uniform	Linear	Standard	GloVe	0.919 ± 0.015	0.672 ± 0.060	0.589 ± 0.050
Uniform	Linear	Normalized	Bert	**0.949 ± 0.008**	0.645 ± 0.028	0.550 ± 0.018
Uniform	Linear	Normalized	GloVe	0.934 ± 0.009	**0.748 ± 0.057**	0.516 ± 0.010
Uniform	Log	Standard	Bert	**0.944 ± 0.005**	0.624 ± 0.039	0.586 ± 0.038
Uniform	Log	Standard	GloVe	0.906 ± 0.008	**0.711 ± 0.050**	0.571 ± 0.011
Uniform	Log	Normalized	Bert	**0.944 ± 0.006**	**0.695 ± 0.027**	0.505 ± 0.029
Uniform	Log	Normalized	GloVe	0.918 ± 0.023	0.680 ± 0.057	**0.612 ± 0.081**
tf-idf	Linear	Standard	Bert	**0.943 ± 0.010**	0.675 ± 0.035	0.574 ± 0.064
tf-idf	Linear	Standard	GloVe	0.881 ± 0.016	**0.744 ± 0.052**	**0.607 ± 0.061**
tf-idf	Linear	Normalized	Bert	**0.947 ± 0.021**	**0.705 ± 0.021**	0.511 ± 0.027
tf-idf	Linear	Normalized	GloVe	0.925 ± 0.013	**0.721 ± 0.051**	0.544 ± 0.039
tf-idf	Log	Standard	Bert	**0.943 ± 0.007**	**0.705 ± 0.030**	0.587 ± 0.027
tf-idf	Log	Standard	GloVe	0.921 ± 0.016	**0.747 ± 0.042**	**0.604 ± 0.012**
tf-idf	Log	Normalized	Bert	**0.943 ± 0.005**	0.689 ± 0.038	**0.618 ± 0.028**
tf-idf	Log	Normalized	GloVe	0.913 ± 0.015	**0.762 ± 0.070**	0.545 ± 0.053

Table 6: Test set performance on MultiWoZ Synthetic data setting.

A.1, and the embedding used is GloVe. All reported results are averaged over 10 splits. Highlighted in bold are the highest performing methods, or methods within the standard deviation of the highest performing method.

Figure 9 and Figure 10 summarize the few-shot training results for the MultiWoZ and SGD synthetic data settings when training with 1 shot, proportional 1 shot, and 3 shots.

D.4 Additional Experiments

Throughout this section, we provides additional dialog structure experiments to further understand when the injection of common-sense knowledge as structural constraints is beneficial. The additional experiments are broken into the following: 1) A study of the sparsity introduced into the tokens in the SGD synthetic setting, and 2) An exploration of an alternative principled soft logic formulation in the MultiWoZ setting.

D.4.1 Sparsity

In this experiment we explore varying the sparsity that was introduced to the token weights, as described in Appendix A.1. Table 8 shows the performance over the three aspects: 1) representation learning, 2) few-shot learning, and 3) structure induction. When the percent of non-zero word weights is 100.00%, this implies the model is trained on full supervision, while the non-zero word weights at 0.00% represents the unsupervised DD-VRNN results. Surprisingly, we find that in all data settings we see substantial improvement to all aspects across the board. Even when the non-zero word weight percentage is 0.02%, resulting in 54 non-zero weights, we still see approximately a 20% improvement to the AMI. Note, 54 non-zero weights is equivalent to about two identifiable

Bag-of-Words Weights	Constraint Loss	Representation Learning (Class Balanced Acc.)	Few-Shot Learning (Class Balanced Acc.)	Structure Induction (AMI)
Uniform	Linear	0.983 ± 0.003	0.717 ± 0.021	0.754 ± 0.032
Uniform	Log	**0.992 ± 0.003**	**0.758 ± 0.015**	0.811 ± 0.005
Supervised	Linear	0.988 ± 0.004	0.714 ± 0.021	0.746 ± 0.035
Supervised	Log	**0.993 ± 0.004**	0.741 ± 0.019	**0.820 ± 0.005**

Table 7: Test set performance on SGD Synthetic data setting.

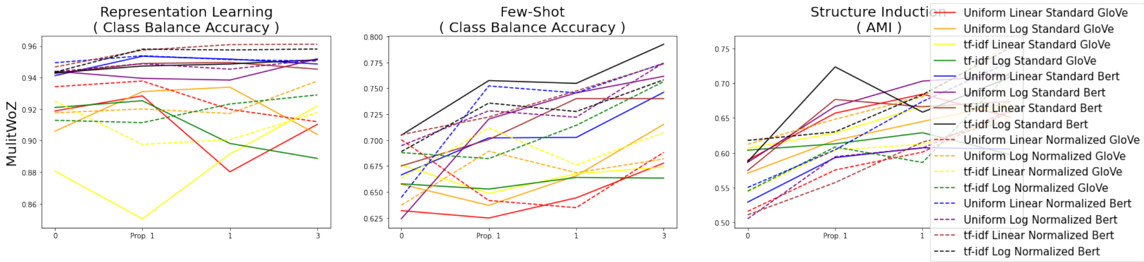

Figure 9: Average performance for Representation Learning, Few-Shot, and Structure Induction for the MulitWoZ dataset with varying amount of supervision.

Non-Zero Word Weights Percentage	Count	Representation Learning (Class Balanced Acc.)	Few-Shot Learning (Class Balanced Acc.)	Structure Induction (AMI)
100.00%	261651	0.9997 ± 0.0006	0.9527 ± 0.0083	0.9999 ± 0.0001
3.25%	8499	0.9995 ± 0.0005	0.9636 ± 0.0028	0.9962 ± 0.0006
0.92%	2418	0.9995 ± 0.0002	0.9475 ± 0.0074	0.9616 ± 0.0010
0.42%	1111	0.9955 ± 0.0010	0.9213 ± 0.0053	0.9450 ± 0.0020
0.19%	504	0.9954 ± 0.0016	0.8591 ± 0.0082	0.7954 ± 0.0018
0.10%	262	0.9904 ± 0.0025	0.8241 ± 0.0243	0.8071 ± 0.0056
0.02%	54	0.9848 ± 0.0019	0.8193 ± 0.0111	0.6607 ± 0.0014
0.00%	0	0.9443 ± 0.0107	0.7283 ± 0.0127	0.5527 ± 0.0171

Table 8: Test set performance on the SGD Synthetic data setting over varying sparsity in the token weights.

Soft Logic	Bag-of-Words Weights	Constraint Loss	Representation Learning (Class Balanced Accuracy)	Few-Shot Learning (Class Balanced Accuracy)	Structure Induction (AMI)
Lukasiewicz	Uniform	Linear	0.9188 ± 0.0150	0.6320 ± 0.0290	0.5892 ± 0.0496
	Uniform	Log	0.9060 ± 0.0083	0.6574 ± 0.0184	0.5707 ± 0.0105
	tf-idf	Linear	0.8807 ± 0.0164	0.6761 ± 0.0289	0.6066 ± 0.0605
	tf-idf	Log	0.9210 ± 0.0160	0.6579 ± 0.0204	0.6037 ± 0.0120
Product Real	Uniform	Linear	0.9151 ± 0.0566	0.6194 ± 0.0529	0.3928 ± 0.1881
	Uniform	Log	0.8807 ± 0.0502	0.6174 ± 0.0525	0.4579 ± 0.1897
	tf-idf	Linear	0.9176 ± 0.0369	0.6741 ± 0.0411	0.4392 ± 0.1903
	tf-idf	Log	0.9232 ± 0.0147	0.6479 ± 0.0367	0.5202 ± 0.0455

Table 9: Test set AMI and standard deviation on MulitWoZ data setting on two soft logic relaxations.

tokens per class.

D.4.2 Alternative Soft Logic Approximation

In this experiment we explore an alternative soft logic formulation, *Product Real* logic, which is used in another principled NeSy framework called *Logic Tensor Networks* (Badreddine et al., 2022). Similar to the *Lukasiewicz* logic, Product Real logic approximates logical clauses with linear inequali-

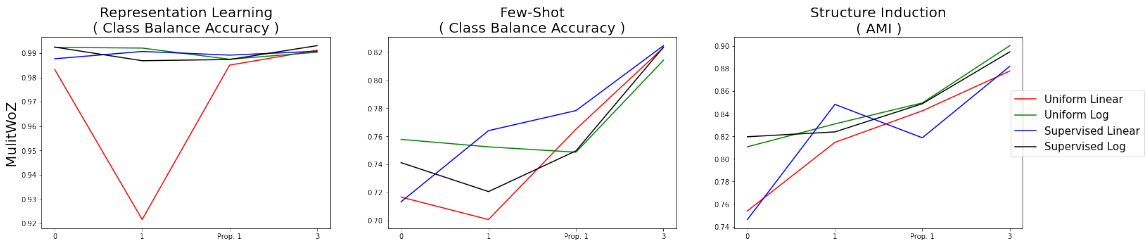

Figure 10: Average performance for Representation Learning, Few-Shot, and Structure Induction for the SGD synthetic dataset with varying amount of supervision.

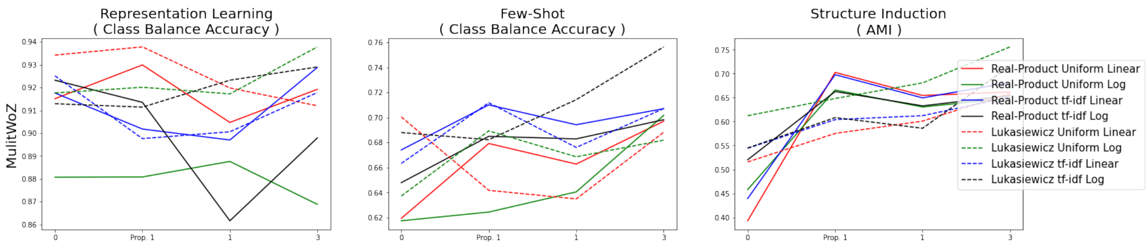

Figure 11: Average performance for Representation Learning, Few-Shot, and Structure Induction for the MultiWoZ dataset with varying amount of supervision on two soft logic relaxations.

ties:

$$A \wedge B = A * B$$
$$A \vee B = A + B - A * B$$
$$\neg A = 1.0 - A$$

where A and B are either ground atoms or logical expressions over atoms. In either case, they have values between [0,1].

Table 9 summarizes the unsupervised results for the MultiWoZ data setting over both the Product Real and Lukasiewicz logics. The results are reported over the three aspects for four different model settings; uniform/supervised bag-of-words weights, and linear/log constraint loss. All reported results are averaged over 10 splits using a GloVe embedding. Surprisingly, in the Structure Induction aspect, Lukasiewicz logic out performs Product Real logic by over 15% in all settings. This result is interesting, as the performance for the representation learning and few-shot learning aspects are roughly equivalent. As both of these aspects use the learned hidden representation, these values suggest that the Lukasiewicz results are aiding the dialog structure induction task without overfitting the hidden representation.

Figure 11 summarizes the few-shot training results for the MultiWoZ synthetic data settings when training with 1 shot, proportional 1 shot, and 3 shots. Noticeably, with the introduction of labels, the Product Real logic closes the gap in all three aspects. However, when observing the largest semi-supervised setting, Lukasiewicz logic still has an edge.