Scalable Theory-Driven Regularization of Scene Graph Generation Models

Davide Buffelli[*†1], **Efthymia Tsamoura**[†2]

[1] University of Padova, Via Gradenigo 6/b 35131, Padova, Italy
[2] Samsung AI, 50-60 Station Road CB1 2JH, Cambridge, United Kingdom
davide.buffelli@phd.unipd.it, efi.tsamoura@samsung.com

Abstract

Several techniques have recently aimed to improve the performance of deep learning models for Scene Graph Generation (SGG) by incorporating background knowledge. State-of-the-art techniques can be divided into two families: one where the background knowledge is incorporated into the model in a subsymbolic fashion, and another in which the background knowledge is maintained in symbolic form. Despite promising results, both families of techniques face several shortcomings: the first one requires ad-hoc, more complex neural architectures increasing the training or inference cost; the second one suffers from limited scalability w.r.t. the size of the background knowledge. Our work introduces a regularization technique for injecting symbolic background knowledge into neural SGG models that overcomes the limitations of prior art. Our technique is model-agnostic, does not incur any cost at inference time, and scales to previously unmanageable background knowledge sizes. We demonstrate that our technique can improve the accuracy of state-of-the-art SGG models, by up to 33%.

Introduction

A *scene graph* is a set of *facts* describing the objects occurring in an image and their inter-relationships. *Scene Graph Generation* (SGG) asks to identify all the facts that hold in an image. Using prior knowledge (for instance commonsense knowledge bases and knowledge graphs (Sap et al. 2019)) is particularly appealing in SGG, as relationships in scene graphs naturally adhere to commonsense principles. This intuition has led to the introduction of *neurosymbolic* techniques (d'Avila Garcez, Broda, and Gabbay 2002) that inject background knowledge into a neural model at training-time and/or use it at inference-time (also called *testing-time*) to amend its predictions.

Neurosymbolic SGG techniques are divided into two major families. The first one represents knowledge in a subsymbolic fashion and incorporates it either only at training-time (Xie et al. 2019), at testing-time (Zareian et al. 2020), or both at training- and testing-time (Gu et al. 2019; Zareian, Karaman, and Chang 2020). The second family maintains knowledge in symbolic form and injects it into the model at training-time only (Donadello, Serafini, and d'Avila Garcez 2017; van Krieken, Acar, and van Harmelen 2019). While they have led to promising results, both groups of techniques face several shortcomings. The first one requires introducing ad-hoc, more complex neural architectures, and accessing the background knowledge at inference-time, thus increasing the training or testing cost. More importantly, ad-hoc neural architectures make it difficult to take advantage of state-of-the-art, neural SGG models, such as VCTree (Tang et al. 2019). The second family suffers from limited scalability with respect to the number of formulas considered, making them impractical in real-world scenarios.

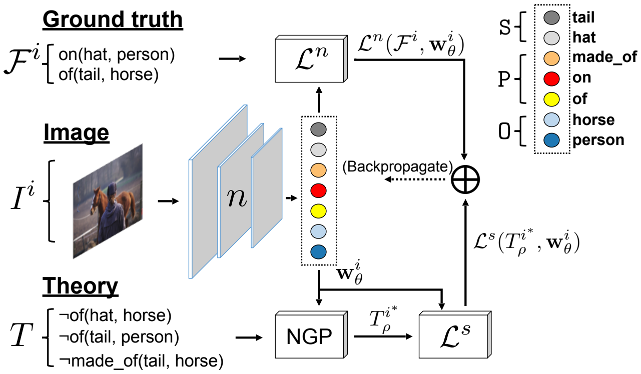

Figure 1: At training-time, background knowledge expressed through negative formulas in first-order logic is injected into a deep model n so that the model's predictions $\mathbf{w}_\theta^i$ for each input image I^i adhere to the background knowledge T. Knowledge injection is performed via a logic-based loss function $\mathcal{L}^s$. To scale to large theories, neural-guided projection (NGP) selects a fixed-size subset $T_\rho^{i^*}$ of the theory to compute the loss for each I^i.

Our work introduces a neurosymbolic regularization technique in which symbolic background knowledge, also referred to as a *theory*, is used as an additional supervision signal for a neural model (see Figure 1). Our objective is to amend the neural network when its predictions do not abide by the background knowledge. The main difference between our proposal and prior art on neurosymbolic SGG is

[*] Work partially done during an internship at Samsung AI.

[†] These authors contributed equally.

that, instead of providing examples of what the neural model should predict (as in (Gu et al. 2019; Zareian, Karaman, and Chang 2020; Zareian et al. 2020)), we provide examples of what the model should *not* predict. This is achieved by enforcing negative *integrity constraints (ICs)*, expressed in the form ¬`predicate(subject, object)`, through a logic-based loss function. The class of *negative* ICs, which is not supported by (Gu et al. 2019; Zareian, Karaman, and Chang 2020; Zareian et al. 2020) provides two benefits. Firstly, unlike any other symbolic-based regularization method, it allows us to design a technique that scales in the presence of hundreds of thousands of ICs. To this extent, instead of using the whole theory for regularizing every training sample, we propose a *neural-guided projection (NGP)* procedure that identifies a small subset of ICs which are maximally logically violated under the neural predictions. The task of amending the neural module towards having its outputs abide by the ICs amounts to solving an optimization problem in which the weights of the neural module are updated to minimize the maximum violation of the ICs. Secondly, it is easy for users to (semi-)automatically create such ICs from existing knowledge bases or even from the training data itself, by creating a negative IC out of each fact *not* in the knowledge base or training data. To assess the robustness of NGP, we ran experiments using two different theories. The first one was created by taking the complement of the commonsense knowledge graph ConceptNet (Speer, Chin, and Havasi 2017), while the second one by taking the complement of the training facts.

Beyond outperforming prior relevant (sub)symbolic regularization techniques, NGP offers multiple other benefits. Firstly, unlike (Gu et al. 2019; Zareian, Karaman, and Chang 2020), NGP is oblivious to the neural models and loss function used. Furthermore, it does not require accessing the background knowledge at inference-time like (Gu et al. 2019; Zareian, Karaman, and Chang 2020). Similarly to (Xie et al. 2019; Donadello, Serafini, and d'Avila Garcez 2017; Gu et al. 2019; Zareian, Karaman, and Chang 2020; Zhu, Fathi, and Fei-Fei 2014), as well as to prior art on knowledge distillation (Dao et al. 2021; Hinton, Vinyals, and Dean 2015), we do *not* question the background knowledge. Our analysis shows that NGP is robust to the theory in use, improving accuracy even when considering *only* the complement of the training facts as negative ICs. Our empirical comparison confirms that NGP:

- improves the accuracy of state-of-the-art SGG models, namely IMP (Xu et al. 2017), MOTIFS (Zellers et al. 2018) and VCTree (Tang et al. 2019), by up to 33%;
- scales to theories including approximately 1M ICs– sizes no prior symbolic-based regularization technique supports (Donadello, Serafini, and d'Avila Garcez 2017);
- is particularly effective when applied in conjunction with TDE (Tang et al. 2020), a technique that tackles the bias in the data, improving the performance of IMP, MOTIFS and VCTree by up to 16 percentile units;
- outperforms GLAT (Zareian et al. 2020) and LENSR (Xie et al. 2019), two state-of-the-art regularization techniques that maintain the knowledge in subsymbolic form, by up to 18% and 15%;
- improves the accuracy of SGG models by up to six times when restricting the availability of ground-truth facts.

Via suitable regularization components, such as TDE (Tang et al. 2020), we outperform in accuracy recently introduced state-of-the-art models (Li et al. 2021) by up to 90% and ad-hoc neurosymbolic SGG architectures leveraging external knowledge bases (Gu et al. 2019) by up to 86%.

An extended version of this paper with additional empirical results and examples is available in (Buffelli and Tsamoura 2022). The sources and the data to reproduce our empirical analysis are in https://github.com/tsamoura/ngp.

Preliminaries

First-order logic is a language of *predicates*, *variables* and *constants*. *Terms* are either variables or constants. An *atom* α is an expression of the form $p(\vec{t})$, where p is a predicate and $\vec{t}$ is a vector of terms. *Formulas* are expressions composed over atoms and the logical connectives, $\wedge$, $\vee$ and $\neg$; a formula is propositional if instead of atoms, it is composed over terms. A formula is *ground* when it includes exclusively constants. We use $t \in \varphi$ to denote that a variable t occurs in a propositional formula φ. A *theory* T is a set of formulas. The set of all possible atoms formed using the predicates and the constants occurring in T is the *universe* U of T. An *interpretation* J of T is a total mapping from the elements in U to a domain. We denote by $J(\varphi)$ the value of φ in J.

Classical semantics Interpretations J in classical Boolean logic map elements in the universe to either true ($\top$) or false ($\bot$). We say that J *satisfies* φ if φ evaluates to true in J, i.e., $J(\varphi) = \top$, and refer to J as a *model* of φ.

Fuzzy logic semantics Interpretations in fuzzy logic map elements in the universe to the interval $[0, 1]$. There are multiple ways[1] to define the logical connectives (see (van Krieken, Acar, and van Harmelen 2020)). We say that J *satisfies* φ if $J(\varphi) = 1$.

Probabilistic semantics In probabilistic logics, similarly to the classical case, statements are either true or false. However, a probability is assigned to these truth values (Hájek, Godo, and Esteva 2013). Consider a propositional formula φ composed over independent Bernoulli random variables, where each variable t is true with probability $p(t)$ and false with probability $1 - p(t)$. Let $\mathbf{p}$ denote the vector of the probabilities so assigned to the variables. The probability $P(J, \mathbf{p})$, of an interpretation J under $\mathbf{p}$ is zero if J is *not* a model of φ; otherwise it is given by:

$$\prod_{\mathtt{t} \in \varphi | J(\mathtt{t}) = \top} p(\mathtt{t}) \cdot \prod_{\mathtt{t} \in \varphi | J(\mathtt{t}) = \bot} 1 - p(\mathtt{t}) \ . \tag{1}$$

Given (1), the probability of formula φ being true under $\mathbf{p}$, denoted as $P(\varphi|\mathbf{p})$, is the sum of the probabilities of all the models of φ under $\mathbf{p}$ ((Chavira and Darwiche 2008)):

$$P(\varphi|\mathbf{p}) = \sum_{J \text{ model of } \varphi} P(J, \mathbf{p}) \ . \tag{2}$$

[1] The truth of ground formula φ is: $J(\neg\varphi) := 1 - J(\varphi)$, $J(\varphi_1 \wedge \varphi_2) := \max\{0, J(\varphi_1) + J(\varphi_2) - 1\}$, $J(\varphi_1 \vee \varphi_2) := \min\{1, J(\varphi_1) + J(\varphi_2)\}$ in Lukasiewicz t-(co)norms.

Example 1 *Consider the formula $\phi = \neg(\texttt{h} \wedge \texttt{d} \wedge \texttt{e})$, where* h *stands for horse,* d *for drinks and* e *for eye. All interpretations of ϕ, apart from the one assigning true to each variable, are models of the formula, i.e., the formula evaluates to true in those interpretations. Assuming that each one of the above terms is assigned a probability $p(\cdot)$, the probability of the interpretation that assigns each variable to false is computed as $(1 - p(\texttt{e})) \times (1 - p(\texttt{d})) \times (1 - p(\texttt{h}))$.*

Proposed Framework

Scene graph generation aims to identify all the predicate(subject,object) facts that hold in an image. Let S, P and O be the sets of possible subject, predicate and object terms, respectively. Let also n be a neural module that takes an input image and outputs the facts that are predicted to hold in that image. Without loss of generality, we assume that the output neurons of n are divided into three mutually disjoint sets so that there is a one-to-one mapping between the neurons within each set and the elements included in sets S, P and O. We use S, P and O to denote both the sets of terms and the sets of neurons mapped to those terms and use t to refer both to a term and to the neuron that maps to t. We denote by $w_\theta(\texttt{t})$ the activation value of output neuron t, where θ denotes the trainable parameters of n, and by $\mathbf{w}_\theta$ the vector of activation values of the output neurons, i.e., the predictions of n.

Facts in a scene graph usually abide by commonsense knowledge. We focus on commonsense knowledge encoded as a theory T in first-order logic and in particular on theories in the form of *integrity constraints* (ICs). Namely, an example of a *negative* IC is the formula φ given by $\neg\texttt{drinks}(\texttt{horse}, \texttt{eye})$, which expresses the restriction that a horse cannot drink an eye. Hereafter, we will consider T to include exclusively *negative*, *atomic* ICs.

Semantics A theory T can be used to penalize a model n. For instance, penalizing n under φ involves adjusting n's weights θ so that the neurons drinks, horse and eye cannot *simultaneously* take high activation values. In the language of logic, the terms in S, P and O form a universe. When adopting a probabilistic logic semantics, the activation values $\mathbf{w}_\theta$ of the output neurons can be seen as the likelihood $\mathbf{p} = \mathbf{w}_\theta$ of those terms. When adopting the semantics of fuzzy logic, instead, the vector $\mathbf{w}_\theta$ can be seen as an interpretation J of the output terms as activation values map terms to the interval $[0, 1]$, see above.

Loss Functions

To inject background knowledge into a neural model, we need to quantify the level to which an IC φ is *consistent* with the neural predictions $\mathbf{w}_\theta$. In the case of probabilistic logic, we denote this level of consistency by $P(\varphi|\mathbf{w}_\theta)$ (see (2)). In fuzzy logic, we denote this level of consistency by $\mathbf{w}_\theta(\varphi)$, as $\mathbf{w}_\theta$ is treated as an interpretation. Our framework is not bound to a specific semantics for interpreting theory T, adopting any semantics. To transparently support semantics that blend classical logic with uncertainty, we assume the existence of a function $SAT : (\varphi, \mathbf{w}_\theta) \rightarrow R^+$ expressing the degree of consistency of φ with $\mathbf{w}_\theta$.

Quantifying the consistency between φ and $\mathbf{w}_\theta$ allows us to define a loss function $\mathcal{L}^s(\varphi, \mathbf{w}_\theta)$ that is inversely proportional to $SAT(\varphi, \mathbf{w}_\theta)$. For instance, to define a loss based on (2), we can employ standard cross entropy (as in (Tsamoura, Hospedales, and Michael 2021)). The cross entropy of (2) is also known as *semantic loss* (SL) (Xu et al. 2018). Again, we do not stick to a specific loss function or semantics as in prior art, e.g., (Donadello, Serafini, and d'Avila Garcez 2017), but rather spell out the properties a loss function should satisfy to be incorporated into our framework: (i) $\mathcal{L}^s(\varphi, \mathbf{w}_\theta) = 0$ if the probability of φ under $\mathbf{w}_\theta$ is one (in the case of probabilistic logic) or $\mathbf{w}_\theta(\varphi) = 1$ (in the case of fuzzy logic); (ii) $\mathcal{L}^s$ is differentiable almost everywhere. The first property is to ensure the soundness of the loss function w.r.t. the logic semantics, while the second one is to ensure the ability to train via backprobagation. We use $\mathcal{L}^s(T, \mathbf{w}_\theta)$ as a shorthand for $\mathcal{L}^s(\bigwedge_{\varphi \in T} \varphi, \mathbf{w}_\theta)$.

Optimization Objective

We are now ready to introduce our technique. Let $I^1, \dots, I^m$ be a sequence of training images. SGG benchmarks such as Visual Genome (VG) (Krishna et al. 2017) include for each image I^i a ground truth set $\mathcal{F}^i$ of predicate(subject,object) facts representing relationships that hold in I^i. State-of-the-art neural modules are trained based on loss functions $\mathcal{L}^n$ that take as arguments the facts in $\mathcal{F}^i$ and the neural predictions for I^i. We denote by $\mathbf{w}^i_\theta$ the predictions of n for I^i. As increasing the level of consistency between the ICs in T and $\mathbf{w}^i_\theta$ reduces to minimizing the loss function $\mathcal{L}^s$, our optimization objective becomes:

$$\theta^* := arg \min_\theta \beta_1 \cdot \sum_{i=1}^{m} \mathcal{L}^n(\mathcal{F}^i, \mathbf{w}^i_\theta) + \beta_2 \cdot \sum_{i=1}^{m} \mathcal{L}^s(T, \mathbf{w}^i_\theta).$$

Above, β_1 and β_2 are hyperparameters setting the importance of each component of the loss. In our empirical evaluation, those hyperparameters are computed in an automated fashion using (Kendall, Gal, and Cipolla 2018). The loss function can be an arbitrary, non-linear function and hence $\mathcal{L}^s(T, \mathbf{w}^i_\theta)$ is not necessarily equal to $\sum\limits_{\varphi \in T} \mathcal{L}^s(\varphi, \mathbf{w}^i_\theta)$.

Neural-Guided Projection

Commonsense knowledge bases can be quite large. Hence, if naively implemented, regularization would be very time consuming if not infeasible. To overcome this limitation in a way that aligns with our optimization objective, for each training image I^i we identify the subset $T^{i^*}_\rho$ of ρ integrity constraints associated with the highest value of $\mathcal{L}^s$ among all possible subsets T^i_ρ of ρ ICs. We call the elements of $T^{i^*}_\rho$ the *maximally non-satisfied ICs*:

$$T^{i^*}_\rho := arg \max_{T^i_\rho \subseteq T} \mathcal{L}^s(T^i_\rho, \mathbf{w}^i_\theta), \tag{3}$$

and regularize the neural module w.r.t. those constraints. Regularizing using the maximally non-satisfied ICs maximizes our chances of providing meaningful feedback to the model. Consider again the IC $\varphi = \neg\texttt{drinks}(\texttt{horse}, \texttt{eye})$. If the likelihood of φ being

Algorithm 1: NGP$(I, \mathcal{F}, T, n_t) \to n_{t+1}$

```
$\mathbf{w} := n_t(I)$
$T^*_\rho := arg \max_{T_\rho \subseteq T} \mathcal{L}^s(T_\rho, \mathbf{w})$
$\ell := \beta_1 \cdot \mathcal{L}^n(\mathcal{F}, \mathbf{w}) + \beta_2 \cdot \mathcal{L}^s(T^*_\rho, \mathbf{w})$
$n_{t+1} :=$ backpropagate$(n_t, \bigtriangledown \ell)$
return $n_{t+1}$
```

Note: β_1, β_2 and ρ are hyperparameters.

Algorithm 2: GREEDY$(I, \rho, T, n_t) \to T^*$

```
$\mathbf{w} := n_t(I)$  $T^* := \emptyset$  $j := 1$
while $|T^*| < \rho$ do
    get the $j$-th p(s,o) prediction maximizing $w(\mathrm{p}) \cdot w(\mathrm{s}) \cdot w(\mathrm{o})$
    if $\neg\mathrm{p}(\mathrm{s}, \mathrm{o})$ is in $T$, then add $\neg\mathrm{p}(\mathrm{s}, \mathrm{o})$ to $T^*$
    $j := j + 1$
end while
return $T^*$
```

true under $\mathbf{w}_\theta$ is close to zero, then we are confident that the prediction needs to be amended; otherwise, we cannot know whether the neural predictions are indeed the correct ones or not and hence, we cannot provide meaningful feedback. In that case, only the ground truth annotations can provide meaningful supervision signal to the neural model.

Our technique, referred to as *neural-guided projection (NGP)*, is summarized in Algorithm 1, which presents the steps taking place on an image-by-image basis. The algorithm denotes by I the input image, by $\mathcal{F}$ the ground truth facts that hold in I, by T the theory, and by n_t the state of the neural module at the t-round of the training process, while ρ defines the number of ICs to choose. An overview of NGP is shown in Figure 1.

Computing $T^{i^*}_\rho$ A greedy strategy for computing the set of maximally non-satisfied ICs is presented in Algorithm 2. The arguments are as in Algorithm 1. Iteratively sampling ρ constraints from the theory and computing $\mathcal{L}^s$ after taking the conjunction of those constraints is also an option.

Proposition 1 summarizes the cases in which the ICs chosen in Algorithm 2 are the ones maximizing (3). Let T^* be the set of ICs returned by Algorithm 2, SL denote the semantic loss and DL2 the fuzzy loss from (Fischer et al. 2019).

Proposition 1 *When $\mathcal{L}^s$=SL, then T^* maximizes* (3) *when the formulas in T^* share no common variables. When $\mathcal{L}^s$=DL2, then T^* always maximizes* (3).

The proof of Proposition 1, a further discussion about $\mathcal{L}^s$ and examples of Algorithm 1 and 2, and the DL2 loss (Fischer et al. 2019) are the in the extended version of our paper.

Experiments

Benchmarks Following previous works, e.g., (Zareian, Karaman, and Chang 2020; Li et al. 2021), we use Visual Genome (**VG**) (Krishna et al. 2017) with the same split adopted by (Tang et al. 2020), and the Open Images v6 (**OIv6**) benchmark (Kuznetsova et al. 2020) with the same split adopted by (Li et al. 2021). We mostly focus on VG, as it is heavily biased (Tang et al. 2020) and more challenging than OIv6 (SGG models have lower performance for VG than for OIv6, as also reported in (Li et al. 2021)).

Theories We used VG$^\neg$ and CNet$^\neg$. VG$^\neg$ was computed by taking the complement of the training facts: we enumerated all combinations of predicates, subjects and objects in VG and for each p(s,o) fact that is *not* in the set of training facts, where p, s and o denotes a predicate, subject and object in the domain of VG, we added to VG$^\neg$ the IC $\neg\mathrm{p}(\mathrm{s}, \mathrm{o})$. We adopted the same approach to create theory CNet$^\neg$ out of ConceptNet's knowledge graph. However, there we considered sparse subgraphs of the entire graph. In particular, we identified subject-object pairs (s, o) having less than ten $\mathrm{p}(\mathrm{s}, \mathrm{o})$ facts in ConceptNet, where p, s and o denotes a predicate, subject and object in the domain of VG or OIv6, respectively. We then repeated the same process for subject-predicate and predicate-object pairs. While the presence, or absence, of a fact in either ConceptNet or the VG training data affects our theory, NGP is not biased by the training facts' frequencies. We did not manually check the resulting theories and hence, they may include constraints that violate commonsense, reflecting real-world noisy settings. Theories CNet$^\neg$ and VG$^\neg$ include approximately 500k and 1M ICs.

Models Similarly to (Tang et al. 2020) and (Suhail et al. 2021), we applied NGP on three state-of-the-art neural SGG models: **IMP** (Xu et al. 2017), **MOTIFS** (Zellers et al. 2018) and **VCTree** (Tang et al. 2019). Prior art (Zareian et al. 2020; Tang et al. 2020; Li et al. 2021) also considers KERN (Chen et al. 2019) and VTransE (Zhang et al. 2017)– we use the more recent model VCTree.

Regularization techniques We considered several recently proposed state-of-the-art regularization techniques:

- **TDE** (Tang et al. 2020), a neural-based technique that operates at inference-time and aims at removing the bias towards more frequently appearing predicates in the data;
- **GLAT** (Zareian et al. 2020), a neural-based technique that amends SGG models at inference-time using patterns captured from the training facts;
- **LENSR** (Xie et al. 2019), a neural-based technique that amends SGG models at training-time after embedding the input symbolic knowledge into a manifold;
- **LTNs** (Donadello, Serafini, and d'Avila Garcez 2017), a symbolic-based technique that injects the input symbolic knowledge to an SGG model at training-time;
- **ITR**, our own symbolic-based technique that returns the most-likely prediction not violating any input IC, where the likelihood of a prediction is the product of the confidences of its predicate, subject and object as assigned by a model. ITR is an inference-time counterpart to NGP.

LTNs is a direct competitor to NGP, while LENSR and GLAT are the neural counterparts to NGP. TDE does not use commonsense knowledge and hence it is orthogonal to all the other regularization techniques.

Additional architectures We consider **KBFN** (Gu et al. 2019), a state-of-the-art ad-hoc, architecture accessing ConceptNet both at training- and at testing-time; and **BGNN** (Li et al. 2021) a recently-introduced confidence-aware bipartite graph neural network with adaptive message propagation mechanism. BGNN cannot be easily integrated with regularization techniques (as also shown in our results), as it makes use of an ad-hoc data sampling procedure at training-time. Indeed, the authors position BGNN as an alternative to models trained with TDE.

Overview of experimental results We considered the standard tasks of *predicate* and *scene graph classification*. Given an input image, and a set of bounding boxes with labels indicating the subjects/objects contained in each bounding box, predicate classification asks to predict the facts that hold in the image. In scene graph classification, the goal is the same, but the bounding boxes are unlabeled. We used the standard measures Mean Recall@k (mR@k) and zero-shot Recall@k (zsR@k) to assess accuracy. mR@k was proposed as a replacement to recall@k to address the data bias issue in SGG benchmarks (Tang et al. 2020, 2019). zsR@k measures recall@k considering only the facts that are in the testing but not the training set (Lu et al. 2016).

We employed NGP with different loss functions. We set the number of constraints (i.e., ρ in Eq. 3) to $\rho = 3$. We found that this value adds minimum computational overhead while improving mR and zsR. We considered the loss functions DL2 (Fischer et al. 2019) (fuzzy logic) and SL (Xu et al. 2018) (probabilistic logic). NGP(X) denotes NGP employed using loss X. All experiments ran using the full theories. LTNs were prohibitively slow for the size of our theory: using the same computational resources we used for NGP, it would have taken 4,000 hours for training for just one epoch. As such, we do not report results for LTNs.

Table 1 shows the impact of NGP, LENSR and ITR on IMP, MOTIFS and TDE for theory CNet$^\neg$. Similarly, Table 2 shows the impact of NGP, GLAT and LENSR for theory VG$^\neg$. NGP and LENSR adopt VG$^\neg$ for a fair comparison against GLAT, as the latter regularizes SGG models using knowledge mined from the training images. Table 3 studies the integration of TDE and NGP on MOTIFS and VCTree (TDE does not support IMP (Tang et al. 2020)). Table 4 focuses on theory VG$^\neg$ and studies the integration of TDE with NGP and GLAT when the baseline model is VCTree.

The above results are on the VG dataset. Table 5 shows the impact of NGP(SL) with CNet$^\neg$ and TDE on MOTIFS for the OIv6 dataset when the models are trained with limited access to the ground truth labels. In particular, we remove 0%, 50% and 75% of the ground truth facts at training-time, while keeping the corresponding images in the training set. As all the baselines we consider require the ground facts to compute a loss $\mathcal{L}^n$, the above setting leads to discarding each sample that misses ground truth facts when training a baseline model (both with and without TDE). In contrast, when applying NGP, we use only $\mathcal{L}^s$ at training-time when the ground-truth facts are not available. The above setting demonstrates the effectiveness of NGP in weak supervision. We report results for MOTIFS, as it was the most challenging to regularize, as discussed below. OIv6 does not provide zero-shot evaluation and, thus, we report only mR@k. Similarly to Table 5, Figure 2 shows the impact of NGP(SL) with CNet$^\neg$ on IMP and VCTree when reducing 10%–50% of the ground-truth facts in VG. The task of interest is predicate classification. Again, when the ground-truth facts of an image are missing, $\mathcal{L}^s$ is used to back propagate through the SGG model when regularizing under NGP; images that miss ground-truth facts are ignored in the absence of NGP. For completeness, Figure 2 also shows mR and zsR when using the full training set (0% reduction). Figure 3 shows R@100 on a per-predicate basis for predicate and scene graph classification, respectively, when the benchmark is VG. In both cases, the baseline model is VCTree regularized under TDE (blue bars); NGP(SL) is applied using CNet$^\neg$ and $\rho = 2$ (orange bars). Figure 4 reports results on VG for the ad-hoc architecture KBFN and the model BGNN. NGP is applied with CNet$^\neg$ and KBFN with ConceptNet– KBFN does not support negative ICs. Further details are in (Buffelli and Tsamoura 2022).

Key Conclusions

NGP can substantially improve the recall of SGG models. Table 1 shows that NGP with theory CNet$^\neg$ improves the relative mR@k of IMP, MOTIFS and VCTree up to 25%, 3% and 4.5% on predicate classification; on scene graph classification, the improvements are up to 33%, 20% and 6.4%. Table 2 shows that when NGP uses VG$^\neg$, the relative improvements over IMP, and VCTree further increase to 34% and 5% on predicate classification, and to 36% and 3% on scene graph classification. Table 5 shows that NGP can improve the performance of MOTIFS by 4% in predicate classification, even with fewer ground-truth facts. The results in Table 1 for zsR@k also show that NGP can improve a model's generalization capabilities of predicting facts that are missing from the training set.

We observe MOTIFS is sensitive to regularization: LENSR always decreases its recall; NGP increases its recall with CNet$^\neg$, but decreases it when adopting either VG$^\neg$, see Table 2, or the semantics of fuzzy logic, see Table 1. We conjecture that the decreases are because MOTIFS favors the most frequent predicate for a given subject-object pair in the ground-truth facts. Hence, adding a regularization term that penalizes predictions outside of the training facts may lead to severe overfitting explaining also the drastic drop in zsR@k. Regarding the fuzzy logic semantics, the decrease stresses the limitations of techniques like LTNs that are bound to fuzzy logic. Given the above, we only consider probabilistic logic for NGP hereafter, without discarding the potential of fuzzy logic in other scenarios.

NGP outperforms prior regularization techniques in most scenarios. NGP is the most effective regularization technique in most cases in Table 1. For instance, regularization of IMP via NGP(SL) leads to up to 25% higher mR@k over ITR on predicate classification. With the exception of MOTIFS, NGP also outperforms GLAT and LENSR in the scenarios in Table 2 leading to up to 20% and 27% higher accuracy in predicate and scene graph classification. The results show that LENSR fails to provide a meaningful loss for training for scene graph classification. Below, we attempt to explain why. In advance of regularization, LENSR learns a manifold $\mathcal{M}$ representing the input theory and a function q mapping embeddings of predictions into the space of $\mathcal{M}$. At regularization-time, LENSR maps via q the embedding of a $\mathtt{p(s,o)}$ prediction into the space of $\mathcal{M}$, where the em-

Model	Theory	Reg.	Predicate Classification						Scene Graph Classification					
			mR@			zsR@			mR@			zsR@		
			20	50	100	20	50	100	20	50	100	20	50	100
I	-	-	9.26	11.43	12.23	12.23	17.28	19.92	5.57	6.31	6.74	2.04	3.47	3.90
I	CNet¬	ITR	9.27	11.44	12.23	12.24	17.30	19.94	5.61	6.35	6.78	2.08	3.50	3.92
I	CNet¬	LENSR	10.56	13.16	14.22	12.78	18.31	21.06	0.01	0.01	0.02	0.01	0.01	0.01
I	CNet¬	NGP(SL)	11.29	14.22	15.30	12.84	**18.75**	21.84	**6.99**	**8.45**	**8.92**	**2.71**	**4.48**	**5.35**
I	CNet¬	NGP(DL2)	**11.62**	**14.73**	**15.92**	**13.13**	18.57	**21.87**	5.58	6.42	6.95	2.17	3.50	3.93
M		-	12.65	16.08	17.35	1.21	3.34	5.57	6.81	8.31	8.85	0.33	0.65	1.13
M	CNet¬	ITR	12.68	16.10	17.39	1.23	3.35	5.57	6.82	8.32	8.85	0.35	0.66	1.13
M	CNet¬	LENSR	12.50	15.90	17.20	1.12	3.26	5.37	0.30	0.34	0.36	0.02	0.02	0.02
M	CNet¬	NGP(SL)	**12.94**	**16.44**	**17.76**	**1.31**	**3.57**	**5.74**	**8.16**	**10.00**	**10.54**	**0.49**	**1.05**	**1.58**
M	CNet¬	NGP(DL2)	7.35	10.52	12.34	0.27	0.67	1.20	4.92	7.99	6.56	0.13	0.24	1.09
V	-	-	13.07	16.75	18.11	1.04	3.28	5.52	9.29	11.42	12.12	0.48	1.37	2.09
V	CNet¬	ITR	13.71	17.27	18.58	**1.37**	3.80	**6.38**	9.36	11.49	12.19	0.51	1.40	2.17
V	CNet¬	LENSR	13.53	16.98	18.27	1.33	3.83	5.88	0.0	0.01	0.01	0.02	0.02	0.02
V	CNet¬	NGP(SL)	13.69	**17.51**	**18.92**	1.29	**3.85**	6.04	**9.89**	**11.75**	**12.35**	**0.67**	**1.56**	**2.44**
V	CNet¬	NGP(DL2)	**13.86**	17.49	18.77	1.16	3.62	5.68	9.41	11.56	12.12	0.49	1.38	2.39

Table 1: Impact of different regularization strategies on IMP (I), MOTIFS (M) and VCTree (V) using CNet¬. Results on VG.

Model	Theory	Reg.	Predicate Classification						Scene Graph Classification					
			mR@			zsR@			mR@			zsR@		
			20	50	100	20	50	100	20	50	100	20	50	100
I	-	-	9.26	11.43	12.23	12.23	17.28	19.92	5.57	6.31	6.74	2.04	3.47	3.90
I	-	GLAT	10.04	12.44	13.30	11.87	17.04	19.72	5.95	6.75	7.17	2.09	3.40	3.82
I	VG¬	LENSR	10.51	13.29	14.33	12.40	18.07	21.22	0.01	0.01	0.02	0.01	0.01	0.01
I	VG¬	NGP(SL)	**11.82**	**15.16**	**16.46**	**12.39**	**18.18**	**21.13**	**7.14**	**8.60**	**9.15**	**2.95**	**4.62**	**5.66**
M	-	-	12.65	16.08	17.35	1.21	3.34	5.57	6.81	8.31	8.85	0.33	0.65	1.13
M	-	GLAT	**12.82**	**16.26**	**17.60**	1.26	**3.49**	**5.79**	**6.84**	**8.34**	**8.89**	**0.32**	**0.63**	**1.12**
M	VG¬	LENSR	12.57	16.09	17.38	**1.37**	3.41	5.65	0.01	0.01	0.01	0.02	0.02	0.02
M	VG¬	NGP(SL)	12.10	15.28	16.54	1.34	3.43	5.47	6.27	7.94	8.42	0.14	0.35	0.55
V	-	-	13.07	16.75	18.11	1.04	3.28	5.52	9.29	11.42	12.12	0.48	1.37	2.09
V	-	GLAT	13.88	17.51	18.90	1.28	3.87	**6.43**	9.39	11.52	12.20	0.51	1.42	2.17
V	VG¬	LENSR	13.46	17.06	18.49	1.27	3.69	5.98	0.0	0.01	0.02	0.01	0.01	0.01
V	VG¬	NGP(SL)	**14.09**	**17.72**	**19.08**	**1.35**	**3.98**	6.36	**9.57**	**11.68**	**12.49**	**0.61**	**1.51**	**2.39**

Table 2: Impact of different regularization strategies on IMP (I), MOTIFS (M) and VCTree (V) using VG¬. Results on VG.

bedding of $\texttt{p(s,o)}$ is the sum of the word embeddings of $\texttt{s}$, $\texttt{p}$ and $\texttt{o}$ weighted by $w(\texttt{p})$, $w(\texttt{s})$ and $w(\texttt{o})$. The L2 distance between the mapped embedding and $\mathcal{M}$ serves as a loss to back-propagate through an SGG model. As the predictions of a model have higher uncertainty in scene graph classification than in predicate classification (not only $\texttt{p}$, but also $\texttt{s}$ and $\texttt{o}$ are now uncertain), the embedding of $\texttt{p(s,o)}$ will be further away from the prediction embeddings that LENSR has seen while learning q in advance of regularization. This discrepancy leads q to transform the prediction embeddings erroneously, leading to a loss function that provides a meaningless training signal. LENSR was not tested on scene graph classification by the authors (Xie et al. 2019).

NGP complements bias reduction techniques. Regarding MOTIFS, the recall improvements brought by TDE are up to 59% and 73% in predicate and scene graph classification and increase to 62% and 89% when NGP is additionally applied using CNet¬. Regarding VCTree, the recall improvements brought by TDE are up to 62% and 38% in predicate and scene graph classification; when NGP is additionally applied, recall increases up to 88% and 63%. Tables 1 and 3 show that the combination of TDE with NGP leads to much higher improvements than the sum of the improvements obtained by applying each technique separately.

We observe similar improvements when NGP is applied using VG¬. On predicate classification, mR@k increases to 24.07%, 31.06% and 34.53%; on scene graph classification, mR@k increases to 11.19%, 15.10% and 17.66%, see Table 4. The adoption of VG¬ allows us to establish a fair comparison against GLAT. Our empirical analysis shows that GLAT cannot be effectively integrated with bias reduction techniques: when GLAT is applied jointly with TDE, mR@k on predicate classification drops from 19.40%, 25.94% and 29.48% to 13.07%, 19.05% and 23.14%, see Table 4. The corresponding decreases in recall are even larger on scene graph classification: mR@k drops from 10.51%, 14.53%

Model	Theory	Reg.	Predicate Classification						Scene Graph Classification					
			mR@			zsR@			mR@			zsR@		
			20	50	100	20	50	100	20	50	100	20	50	100
M	-	-	12.65	16.08	17.35	1.21	3.34	5.57	6.81	8.31	8.85	0.33	0.65	1.13
M	-	TDE	17.18	23.95	27.66	8.10	13.68	17.11	10.11	13.44	15.35	1.85	3.01	3.68
M	CNet¬	N+TDE	**17.99**	**24.50**	**28.16**	**8.51**	**14.00**	**17.80**	**11.80**	**15.11**	**16.77**	**1.92**	**3.05**	**3.74**
V	-	-	13.07	16.75	18.11	1.04	3.28	5.52	9.29	11.42	12.12	0.48	1.37	2.09
V	-	TDE	19.40	25.94	29.48	8.14	12.38	14.07	10.51	14.53	16.73	1.48	2.54	**3.99**
V	CNet¬	N+TDE	**23.91**	**30.78**	**34.19**	**8.15**	**12.47**	**15.41**	**13.60**	**17.69**	**19.85**	**1.57**	**2.63**	3.63

Table 3: Impact of NGP(SL) (abbreviated as "N") on MOTIFS (M) and VCTree (V) with TDE. Results on VG.

Model	Theory	Reg.	Predicate Classification						Scene Graph Classification					
			mR@			zsR@			mR@			zsR@		
			20	50	100	20	50	100	20	50	100	20	50	100
V	-	-	13.07	16.75	18.11	1.04	3.28	5.52	9.29	11.42	12.12	0.48	1.37	2.09
V	-	TDE	19.40	25.94	29.48	**8.14**	**12.38**	**14.07**	10.51	14.53	16.73	1.48	2.54	**3.99**
V	VG¬	N+TDE	**24.07**	**31.06**	**34.53**	6.30	10.46	12.90	**11.19**	**15.10**	**17.66**	**1.66**	**2.64**	3.51
V	-	-	13.07	16.75	18.11	1.04	3.28	5.52	9.29	11.42	12.12	0.48	1.37	2.09
V	-	TDE	**19.40**	**25.94**	**29.48**	**8.14**	**12.38**	**14.07**	**10.51**	**14.53**	**16.73**	**1.48**	**2.54**	**3.99**
V	-	G+TDE	13.07	19.05	23.14	4.99	8.09	11.01	0.90	2.06	3.60	1.30	2.22	3.18

Table 4: Impact of NGP(SL) (abbreviated as "N") and GLAT (abbreviated as "G") on VCTree (V) with TDE. Results on VG.

% Red.	Reg.	Prd Cls mR@		Sg Cls mR@	
		50	100	50	100
-0%	-	46.10	46.15	**28.90**	**28.92**
-0%	TDE	42.00	42.01	12.87	12.87
-0%	NGP(SL)	**48.65**	**48.70**	26.07	26.10
-50%	-	43.12	43.17	25.81	25.83
-50%	TDE	32.07	32.08	19.34	19.36
-50%	NGP(SL)	**46.41**	**46.46**	**26.21**	**26.24**
-75%	-	42.16	42.17	23.27	23.28
-75%	TDE	34.09	34.10	14.73	14.74
-75%	NGP(SL)	**44.90**	**44.94**	**25.23**	**25.26**

Table 5: Impact of NGP(SL) and TDE on MOTIF when reducing the ground-truth facts from the OIv6 dataset.

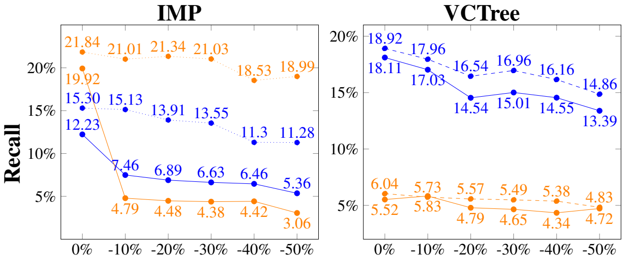

Figure 2: Impact of NGP on IMP and VCTree for predicate classification when reducing VG's ground-truth. Blue lines show mR@100; orange show zR@100. Solid lines show mR and zsR w/o NGP; dotted show mR and zsR w/ NGP.

and 16.73% to 0.90%, 2.06% and 3.60%.

NGP is particularly beneficial when reducing the amount of ground-truth facts. Figure 2 shows that the accuracy of SGG models can substantially decrease when reducing the ground-truth facts. In the case of VG, the most sensitive model is IMP: when reducing the training data by 50%, zsR@100 drops by more than 6.5 times (19.92 % vs. 3.06%), while mR@100 drops by more than two times (12.23 % vs. 5.36%). In the case of OIv6, Table 5, MOTIFS' mR@100 drops from 46.15% to 42.17% in predicate classification when reducing the ground-truth by 75%; in scene graph classification, MOTIFS' mR@100 drops from 28.92% to 23.28%. NGP can lead to drastic accuracy improvements for those cases. Regarding IMP and VG, zsR@100 can increase from 3.06% to 18.99% when reducing the ground-truth by 50%; zsR@100 can similarly increase from 5.36% to 11.28%. Similarly, when reducing by 75% of the ground-truth of OIv6, mR@100 for predicate classification can increase from 42.17% to 44.94% in the case of MOTIFS; mR@100 can increase from 42.17% to 44.94% for scene graph classification, when NGP is applied.

While NGP drops the mR of MOTIFS in scene graph classification when the whole ground-truth is used in OIv6, it is beneficial when reducing the ground-truth by 50% and 75%, Table 5. The high mR for MOTIFS even with significantly fewer ground-truth facts in Table 5 manifests that frequency-based techniques are effective for the OIv6 dataset. Still, the integration with logic-based approaches (NGP) can further improve mR, Table 5. It is also worth noting that while TDE is particularly effective in VG, it decreases the mR of MOTIFS up to 11% in OIv6. This is because OIv6 has a much higher annotation quality, and hence de-biasing is not crucial. Finally, in contrast to NGP, TDE provides no supervi-

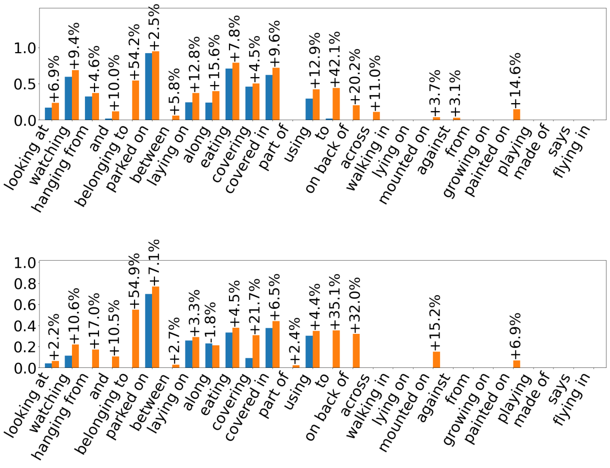

Figure 3: R@100 for the 28 least frequent predicates in VG for predicate and scene graph classification (upper and lower figure, respectively).

sion when reducing the ground-truth facts.

NGP improves recall for less frequent predicates. VG is a highly skewed benchmark (Tang et al. 2020), since 90% of the ground truth facts reference only few predicates (e.g., `looking at` facts are the 0.00263% of the ground truth facts; `flying in` facts are only the 0.00001%). In Figure 3, we plot the R@100 for the 28 least frequent predicates in VG (predicates shown in decreasing order of their occurence frequencies) for predicate classification (upper part) and scene graph classification (lower part). The blue bars show recall of the baseline model (VCTree regularized with TDE); the orange ones show recall after adding NGP(SL) on top. The percentages on the bars show the relative changes in recall due to NGP. Figure 3 shows that NGP brings major improvements in recall for predicates with very few training data, such as `belonging to` and `on back of`, showing it acts as a form of weak supervision.

Regularization can be more effective than sophisticated (neurosymbolic) SGG models. The mR@k of KBFN is 17.01% and 18.43% for predicate classification, see Figure 4. When jointly regularizing VCTree using NGP(SL) and TDE, the mR@k is 30.78% and 34.19%. Similarly, for scene graph generation, the mR@k of KBFN is 15.79% and 17.07%, and 17.69% and 19.85% for the regularized VCTree model. Likewise, the regularized VCTree model reaches up to 90% higher performance than BGNN. These results show that regularizing a standard SGG model like VCTree, can be more effective than ad-hoc, neurosymbolic SGG architectures or more sophisticated models.

Related Work

Regularising neural models using symbolic knowledge has been extensively studied in information and natural language analysis (Wang and Pan 2020; Minervini and Riedel 2018; Rocktäschel, Singh, and Riedel 2015). Unlike the above line of research, NGP focuses on scalable knowledge injection into SGG models under different semantics.

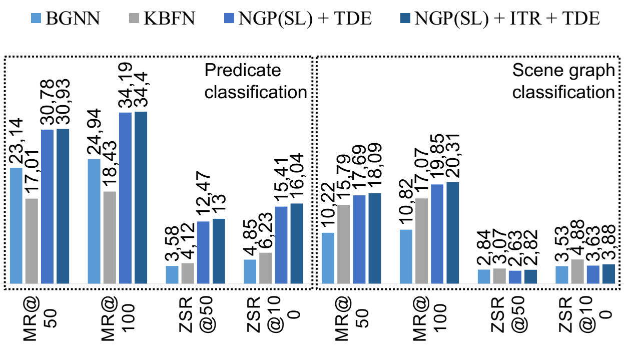

Figure 4: Regularization vs. ad-hoc architectures and sophisticated models. Results on VG.

Differently from contrastive learning (Oord, Li, and Vinyals 2018; Chen and He 2021; Jaiswal et al. 2021) where models are trained in an unsupervised fashion by performing tasks that can be created from the input itself, NGP trains neural models using symbolic domain knowledge. The authors in (Suhail et al. 2021) train a graph neural network to learn the joint conditional density of a scene graph and then use it as a loss function. To deal with the ambiguity in the SGG annotations, the work in (Yang et al. 2021) generates different probabilistic representations of the predicates. In contrast to NGP, the above techniques do not support external knowledge. Finally, the work in (Zhong et al. 2021) generates localized scene graphs from image-text pairs; the technique does not rely on logic, but exclusively on neural models. Integrating logic-based regularization with the above research is an interesting future direction.

Every technique that uses learned or fixed background knowledge as a prior, e.g., (Gu et al. 2019; Zareian et al. 2020), is biased towards that knowledge. Differently from techniques like MOTIFS (Zellers et al. 2018), NGP is not biased by the frequency of the training facts: if the background knowledge is independent of the training facts or their frequencies, then NGP will not be biased toward the training facts or their frequencies. The above holds as both the logic-based losses and NGP's mechanism for choosing the maximally violated ICs are indifferent to any frequencies.

Conclusions

We introduced NGP, the first highly-scalable, symbolic, SGG regularization framework that leads to state-of-the-art accuracy. Future research includes supporting richer formulas and regularizing models under theories mined via knowledge extraction e.g., (Zhu, Fathi, and Fei-Fei 2014)– NGP supports such theories by weighting the ICs. Integrating NGP with neurosymbolic techniques that support indirect supervision like DeepProbLog (Manhaeve et al. 2018), NeuroLog (Tsamoura, Hospedales, and Michael 2021) and ABL (Dai et al. 2019) is another direction for future research.

Acknowledgments

Davide Buffelli was supported, in part, by the Italian Ministry of Education, University and Research (MIUR), under PRIN Project n. 20174LF3T8 "AHeAD" and the initiative "Departments of Excellence" (Law 232/2016), and by University of Padova under project "SID 2020: RATED-X". Part of this work was done while Davide Buffelli was doing an internship at Samsung AI.

References

Buffelli, D.; and Tsamoura, E. 2022. Scalable Regularization of Scene Graph Generation Models using Symbolic Theories. *arXiv:2209.02749*, abs/2209.02749.

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

Chen, T.; Yu, W.; Chen, R.; and Lin, L. 2019. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6163–6171.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Dai, W.-Z.; Xu, Q.; Yu, Y.; and Zhou, Z.-H. 2019. Bridging Machine Learning and Logical Reasoning by Abductive Learning. In *NeurIPS*, 2815–2826.

Dao, T.; Kamath, G. M.; Syrgkanis, V.; and Mackey, L. 2021. Knowledge Distillation as Semiparametric Inference. In *ICLR*.

d'Avila Garcez, A. S.; Broda, K.; and Gabbay, D. M. 2002. *Neural-symbolic learning systems: foundations and applications*. Perspectives in neural computing. Springer.

Donadello, I.; Serafini, L.; and d'Avila Garcez, A. S. 2017. Logic Tensor Networks for Semantic Image Interpretation. In *IJCAI*, 1596–1602.

Fischer, M.; Balunovic, M.; Drachsler-Cohen, D.; Gehr, T.; Zhang, C.; and Vechev, M. T. 2019. DL2: Training and Querying Neural Networks with Logic. In *ICML*, volume 97, 1931–1941.

Gu, J.; Zhao, H.; Lin, Z.; Li, S.; Cai, J.; and Ling, M. 2019. Scene Graph Generation With External Knowledge and Image Reconstruction. In *CVPR*, 1969–1978.

Hájek, P.; Godo, L.; and Esteva, F. 2013. Fuzzy Logic and Probability. *CoRR*, abs/1302.4953.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Jaiswal, A.; Babu, A. R.; Zadeh, M. Z.; Banerjee, D.; and Makedon, F. 2021. A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1).

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.; Shamma, D. A.; Bernstein, M. S.; and Fei-Fei, L. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.*, 123(1): 32–73.

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J. R. R.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; Duerig, T.; and Ferrari, V. 2020. The Open Images Dataset V4. *International Journal of Computer Vision*, 128(7): 1956–1981.

Li, R.; Zhang, S.; Wan, B.; and He, X. 2021. Bipartite Graph Network With Adaptive Message Passing for Unbiased Scene Graph Generation. In *CVPR*, 11109–11119.

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual Relationship Detection with Language Priors. In *ECCV*.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. DeepProbLog: Neural Probabilistic Logic Programming. In *NeurIPS*, 3749–3759.

Minervini, P.; and Riedel, S. 2018. Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge. In *CoNLL*, 65–74.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv:1807.03748*.

Rocktäschel, T.; Singh, S.; and Riedel, S. 2015. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In *ACL*, 1119–1129.

Sap, M.; Bras, R. L.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *AAAI*, 3027–3035.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*, 4444–4451.

Suhail, M.; Mittal, A.; Siddiquie, B.; Broaddus, C.; Eledath, J.; Medioni, G. G.; and Sigal, L. 2021. Energy-Based Learning for Scene Graph Generation. In *CVPR*, 13936–13945.

Tang, K.; Niu, Y.; Huang, J.; Shi, J.; and Zhang, H. 2020. Unbiased Scene Graph Generation From Biased Training. In *CVPR*, 3713–3722.

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to Compose Dynamic Tree Structures for Visual Contexts. In *CVPR*, 6619–6628.

Tsamoura, E.; Hospedales, T.; and Michael, L. 2021. Neural-Symbolic Integration: A Compositional Perspective. In *AAAI*.

van Krieken, E.; Acar, E.; and van Harmelen, F. 2019. Semi-Supervised Learning using Differentiable Reasoning. *IF-CoLog Journal of Logic and its Applications*, 6(4): 633–653.

van Krieken, E.; Acar, E.; and van Harmelen, F. 2020. Analyzing Differentiable Fuzzy Implications. In *KR*, 893–903.

Wang, W.; and Pan, S. J. 2020. Integrating Deep Learning with Logic Fusion for Information Extraction. In *AAAI*, 9225–9232.

Xie, Y.; Xu, Z.; Meel, K. S.; Kankanhalli, M. S.; and Soh, H. 2019. Embedding Symbolic Knowledge into Deep Networks. In *NeurIPS*, 4235–4245.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene Graph Generation by Iterative Message Passing. In *CVPR*, 3097–3106.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and Van den Broeck, G. 2018. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *ICML*, 5502–5511.

Yang, G.; Zhang, J.; Zhang, Y.; Wu, B.; and Yang, Y. 2021. Probabilistic Modeling of Semantic Ambiguity for Scene Graph Generation. In *CVPR*, 12527–12536.

Zareian, A.; Karaman, S.; and Chang, S.-F. 2020. Bridging Knowledge Graphs to Generate Scene Graphs. In *ECCV*.

Zareian, A.; Wang, Z.; You, H.; and Chang, S.-F. 2020. Learning Visual Commonsense for Robust Scene Graph Generation. In *ECCV*, 642–657.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural Motifs: Scene Graph Parsing With Global Context. In *CVPR*, 5831–5840.

Zhang, H.; Kyaw, Z.; Chang, S.; and Chua, T. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *CVPR*, 3107–3115.

Zhong, Y.; Shi, J.; Yang, J.; Xu, C.; and Li, Y. 2021. Learning to Generate Scene Graph from Natural Language Supervision. In *ICCV*, 1803–1814.

Zhu, Y.; Fathi, A.; and Fei-Fei, L. 2014. Reasoning about Object Affordances in a Knowledge Base Representation. In *ECCV*, volume 8690, 408–424.