Consistent Scene Graph Generation by Constraint Optimization

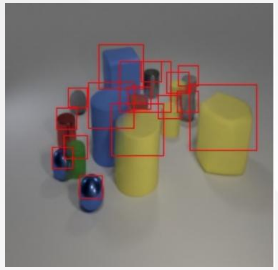
Boqi Chen[1], Kristóf Marussy[2], Sebastian Pilarski[1], Oszkár Semeráth[2], Daniel Varro[1]

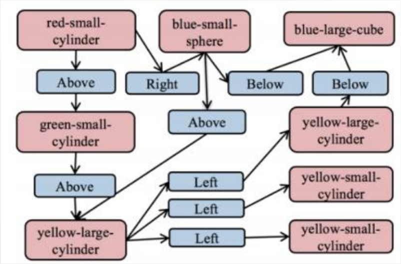
[1]*Department of Electrical & Computer Engineering, McGill University*
[2]*Department of Measurement and Information Systems, Budapest University of Technology and Economics*

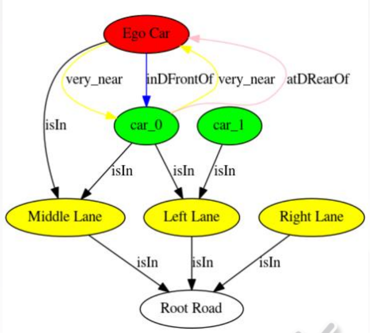
boqi.chen@mail.mcgill.ca, marussy@mit.bme.hu, sebastian.pilarski@mail.mcgill.ca, semerath@mit.bme.hu, daniel.varro@mcgill.ca

Scene Graph Generation (SGG)

- **Scene graphs** represent objects and their relations in an image
- **Scene graph generation** produces a graph from a given image containing ***focused objects*** and their ***relationships***
- Scene graph generation is a common challenge in computer vision







Examples taken from:
Herzig, Roei, et al. "Learning canonical representations for scene graph to image generation." *European Conference on Computer Vision*. Springer, Cham, 2020.
Yu, Shih-Yuan, et al. "Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions." *IEEE Transactions on Intelligent Transportation Systems* (2021).

Motivation: Safety Critical Application

- **Scene graphs** can be used in safety critical fields such as autonomous driving and robotics


US & Canadian companies
+ Add to myFT

Uber self-driving car crashes during US tests

Collision threatens plans to bring high levels of autonomy into commercial use

The Uber SUV on its side following the accident in Tempe, Arizona © Reuters

Richard Waters in San Francisco MARCH 25 2017 63


Waymo van involved in serious collision in Arizona

Devin Coldewey @techcrunch / 7:26 PM EDT • May 4, 2018 Comment


NEWS

Robot kills worker at Volkswagen plant in Germany

Volkswagen has disclosed that a robot has killed a contractor involved in its installation. The fatal accident happened at VW's Baunatal plant, north of Frankfurt on Monday.

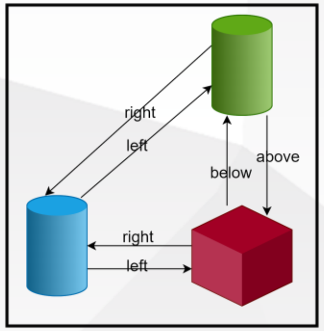
- In such applications, it is important to provide safety guarantee on the produced scenes under consistent situations
  - Law of physics: Car cannot (yet) fly
  - Traffic rules: No contradictory traffic signs

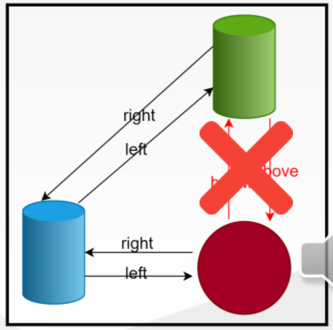
Motivation: Constraints Formulation

- **What is safety?** One aspect of safety is *consistency*: The system should comply with a set of *consistency constraints* Φ
- *Consistency constraints* can be expressed with logic or constraint languages (**FOL**, OLC and VIATRA-Query etc.)

Rule: Nothing can be above a sphere —FOL→ $\forall a, b: \mathrm{Above}(a, b) \Rightarrow \neg \mathrm{Shape}(b, \text{'Sphere'})$

Assumption: ground truth scenes are consistent





Problem Statement

- To *guarantee consistency* in such systems, we define and tackle the problem of *consistent scene graph generation*
- Given a set of constraints Φ, an image I with underlying ground truth scene graph SG_{gt}, find a model $\mathcal{M}$ for scene graph SG such that

1. The generated SG is close to the ground truth SG_{gt} (accurate):

$$P(SG \cong SG_{gt} | \mathcal{M})$$

2. The generated SG satisfy all consistency constraints Φ (consistent)

$$SG \vDash \Phi$$

Problem Statement

accurate consistent

$$P(SG \cong SG_{gt} | \mathcal{M}) \wedge SG \vDash \Phi$$

Many existing deep learning approaches

No explicit guarantees

Assumption: SG_{gt} satisfies Φ: $SG_{gt} \vDash \Phi$

How can we guarantee constraints are always satisfied?

Can help?

$$P(SG \cong SG_{gt} | \mathcal{M}) \qquad SG \vDash \Phi$$

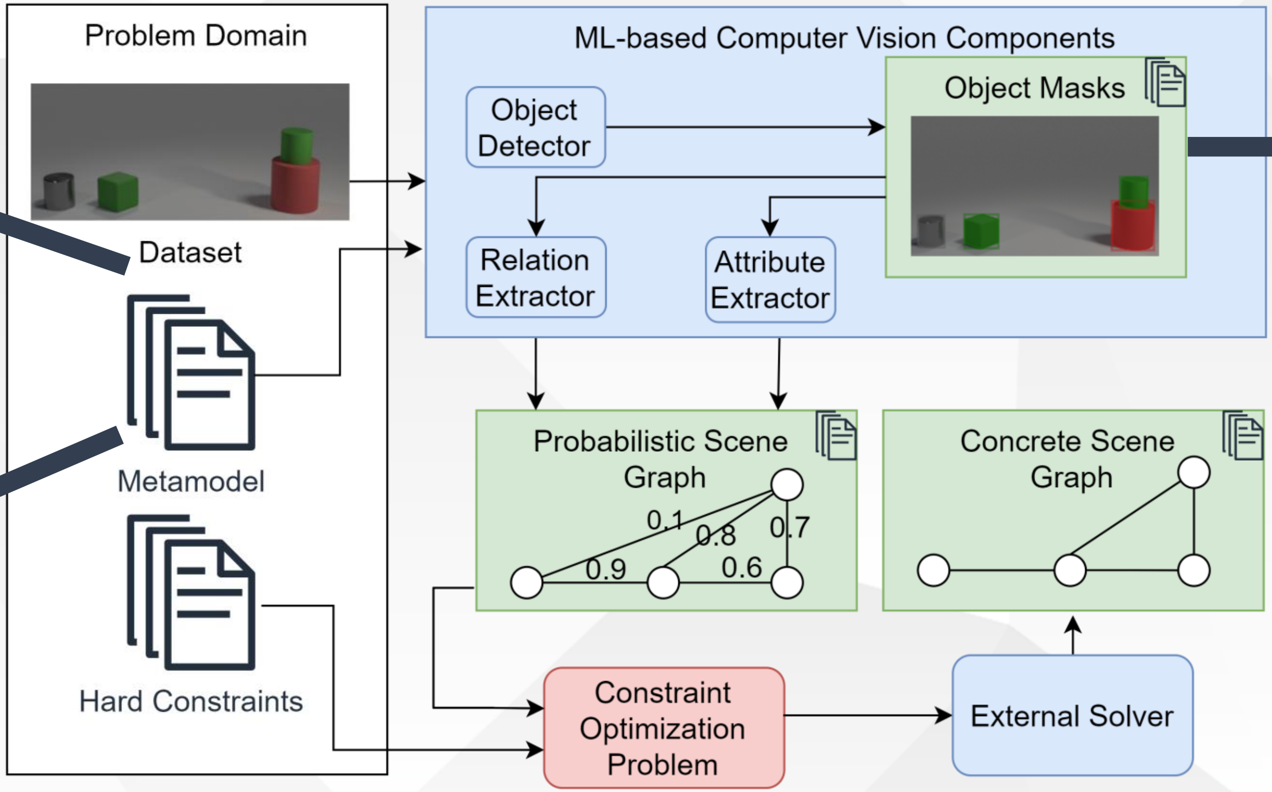
help

Approach Overview

Core Idea: use existing DL methods to optimize for $P(SG \cong SG_{gt}|\mathcal{M})$, and handle $SG \vDash \Phi$ later with *constraint optimization*

Relation types R

Attribute types A and possible values



Localize objects

Approach: ML-based Component

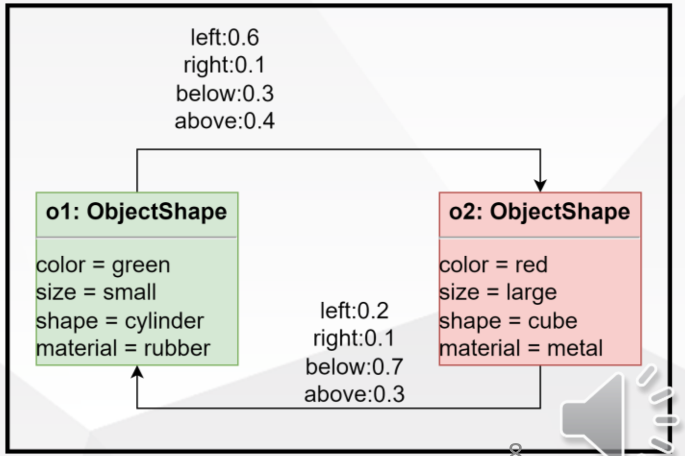
A ML-based vision model outputs two types of independent probabilities for an input image to form a *probabilistic graph*

1. The probability of a pair of objects and a relation type

$$P_R: N \times \mathbb{R} \times N \to [0,1],: P_R(\overrightarrow{n_1 n_2}^r)$$

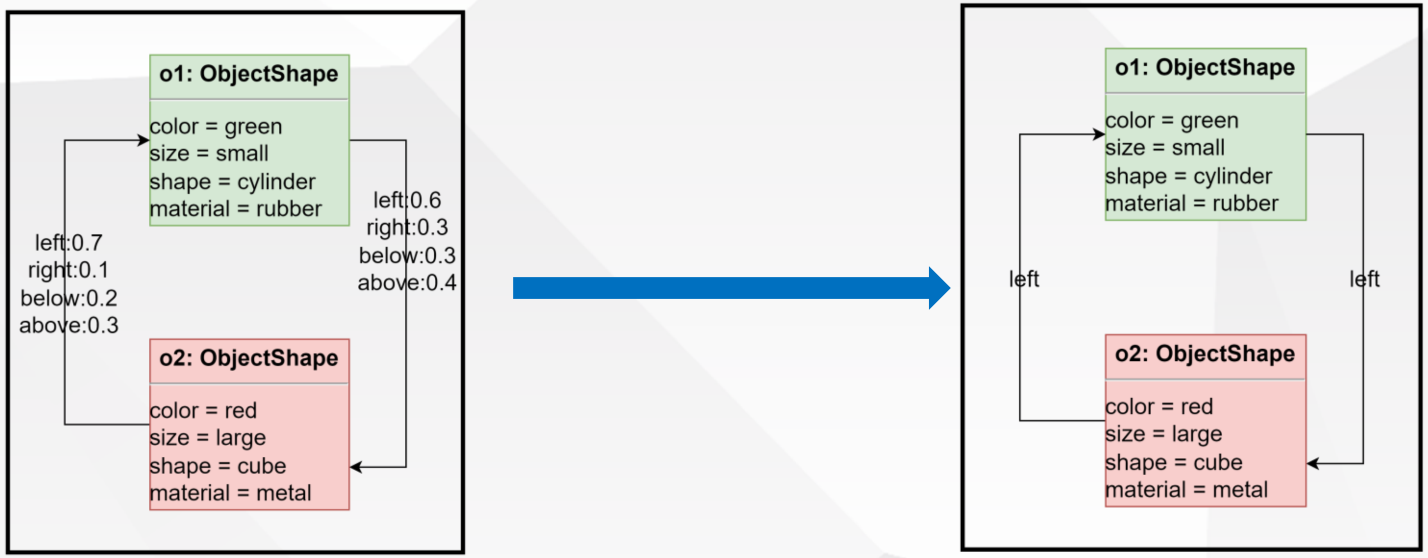
2. The probability of an object, an attribute type and an attribute value

$$P_A: N \times A \times V_a \to [0,1]: P_A(\overrightarrow{n\, v}^a)$$



Approach: Baseline

Commonly, we can choose a *concrete scene graph* from the probabilistic SG by selecting the *most probable relations and attributes* individually



(a) Probabilistic graph

(b) Concrete graph

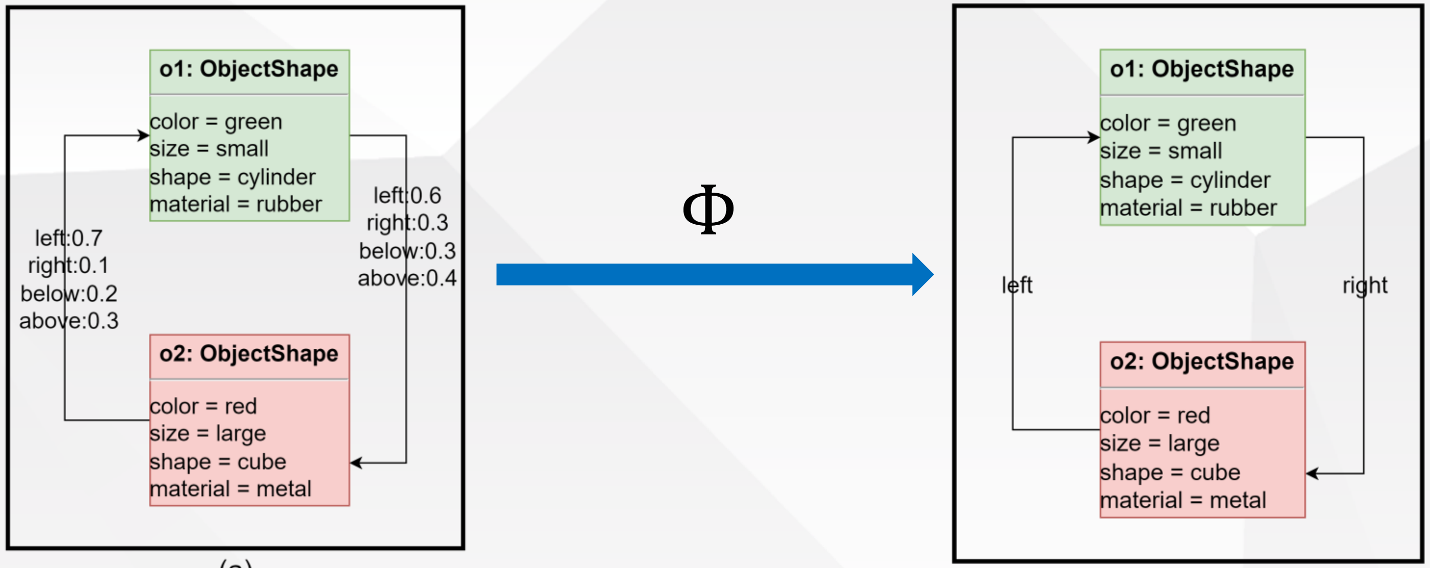
Φ:

1. $left \leftrightarrow right$
2. $\text{below} \leftrightarrow above$

No consideration of Φ!

Approach: Constraint Optimization

Instead, we propose to select the *most probable scene* subject to Φ



(a) Probabilistic graph

(c) Concrete graph

Φ:

1. $left \leftrightarrow right$
2. below $\leftrightarrow$ $above$

Constraint Optimization!

Approach: MAXSAT

MAXSAT is an optimization problem aiming to find the maximum subset of clauses with weights (in CNF)

Many existing solvers: MaxSatz, WBO, SAT4J, **Gurobi**

Given a set of hard constraints Φ, a set of clauses C with weights w

$$\neg\phi:\ -\infty \text{ for each } \phi \in \Phi$$

$$\mathrm{C_i}:\ \mathrm{w_i} \text{ for } \mathrm{i} \in [0, \mathrm{n}]$$

$$max_x \sum_i w_i \cdot 1(x \vDash C_i) \text{ subject to } x \vDash \phi \text{ for each } \phi\ in\ \Phi$$

Approach: MAXSAT

Given a probabilistic graph $\mathbb{G}$ with P_R and P_A, our approach transform it into a MAXSAT problem by:

1. The hard constraints are respected

$$\neg\phi\text{:}\ -\infty \text{ for each } \phi \in \Phi$$

2. Edges and attributes are clauses with weights being the log probabilities

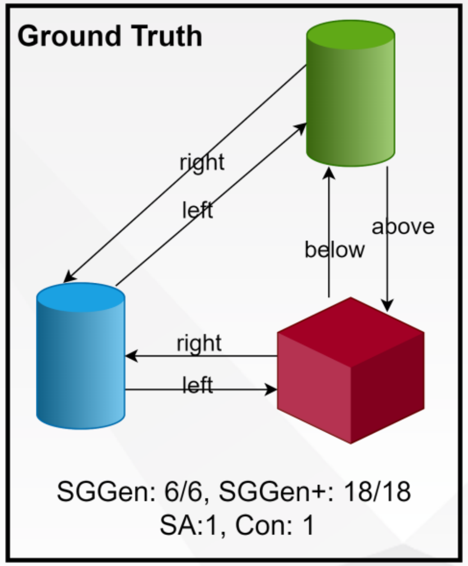
$$x_{\overrightarrow{n_1 n_2}^r}\text{:} \log P_R(\overrightarrow{n_1 n_2}^r) \qquad x_{\overrightarrow{n_1 v}^a}\text{:} \log P_A(\overrightarrow{n_1 v}^a)$$

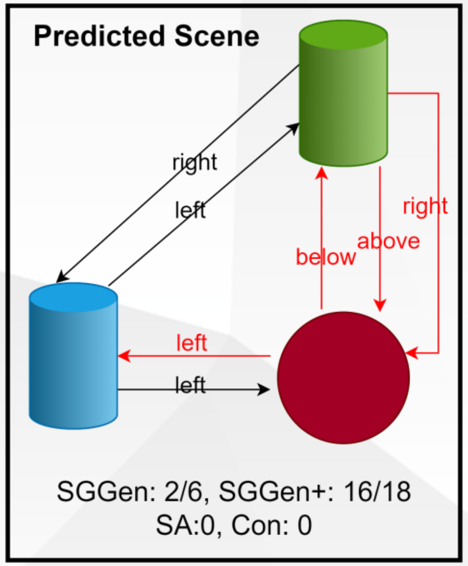
3. The optimization target is to maximize the sum of log probabilities

Evaluation: Metrics

- **SGGen**: measures recall of relations if all attribute of an object is identified correctly
- **SGGen+**: measures recall separately for relations and attributes

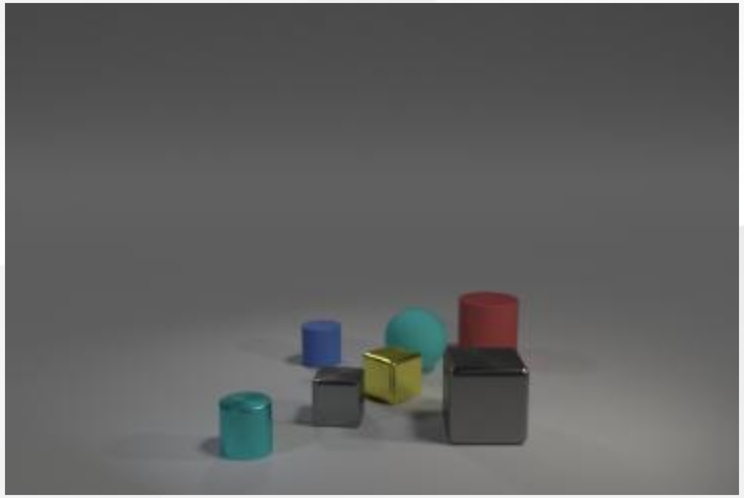
- **Con**: measures *consistency* of the scene
- **SA**: *scene accuracy* measures if the predicted scene is *isomorphic* to the ground truth scene



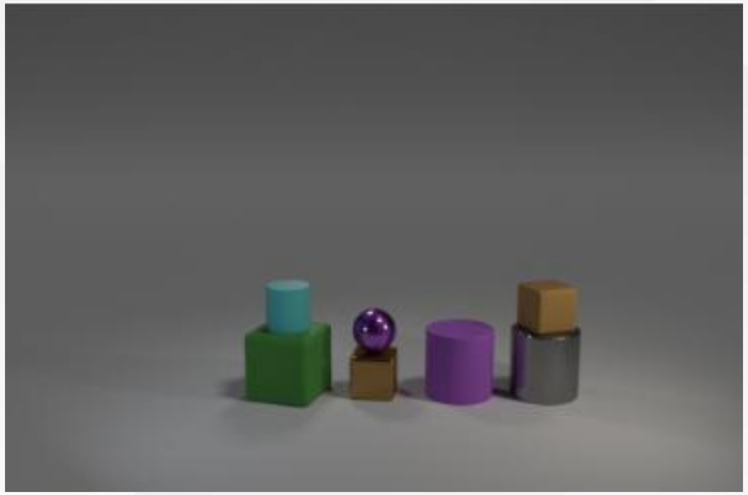


Evaluation: Synthetic Dataset

CLEVR:



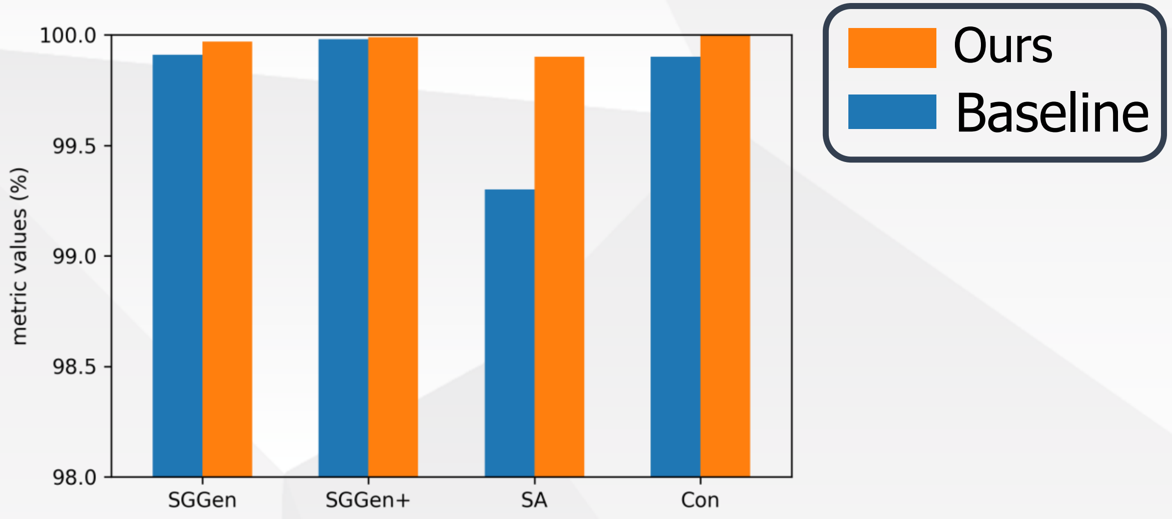
BLOCKWORLD:



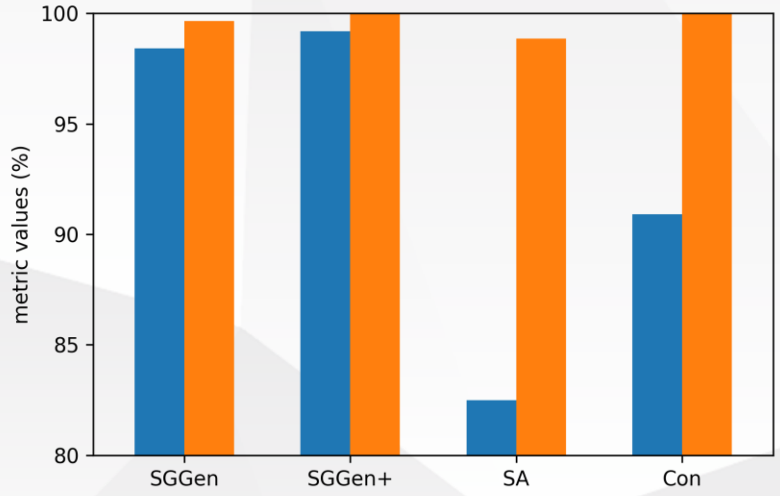
- 4 types of constraints with different complexity were created
- Scenes are generated to satisfy the constraints
- 4000 scenes for training and 2000 for testing

Evaluation: Synthetic Dataset

- Our approach is better than the baseline in all cases
- High values in relation recall (SGGen, SGGen+) does not mean high SA
- Our approach always improves SA by improving Con
  - In fact, we can prove SA is *always at least as good* as the baseline scenes



CLEVR



Block

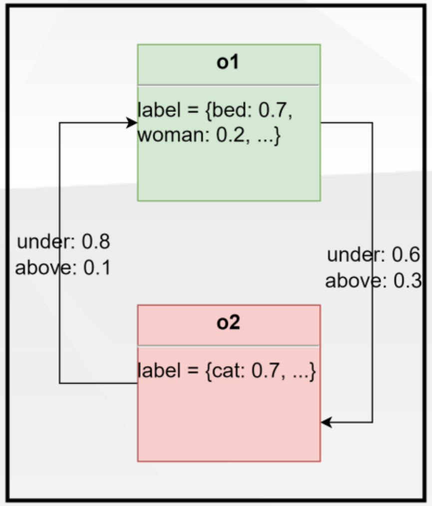
Evaluation: Real Dataset

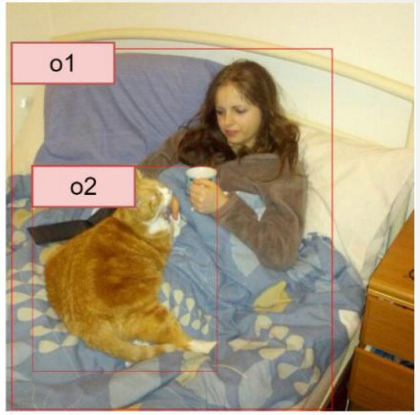
- What about the performance on real-world images?

- We applied our approach on a subset of the Visual Genome dataset with two types of constraints:
    - There must be at least one person in the scene
    - There is no cycle on relations such as 'Above', 'Under'
    - We filtered the datasets with the first constraint
    - 99.85% ground truth satisfies the second type of constraints

Evaluation: Real Dataset

- Probabilistic scenes are derived from a model pre-trained on *original VG dataset*





Bed(o1) → Woman(o1)
Under(o2, o1) → Above(o2, o1)

Metric	Improvement (%)
SGGen	33.04 → **33.15**
SGGen+	63.44 → **63.48**
Con	64.43 → **100**

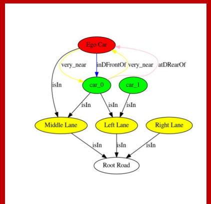
- SA is not measured because the labelled graph is not complete
- Our approach is still able to improve on all metrics while ensuring consistency

Image credit: https://www.flickr.com/photos/todoleo/8310164456/

Conclusion

Scene Graphs

Scene Graphs represents objects and their relations in an image.
Scene graphs can be used by safety critical systems in which consistency is a key.



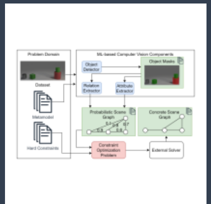
$$P(SG \cong SG_{gt}|\mathcal{M})$$
$$SG \vDash \Phi$$

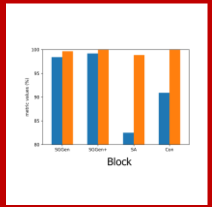
Consistent Scene Graph Generation

Generate scene graphs from images that comply to the set of constraints Φ.

Constraint optimization

use existing DL methods to optimize for $P(SG \cong SG_{gt}|\mathcal{M})$, and handle $SG \vDash \Phi$ later with a *constraint optimization.*





Neural Symbolic Reasoning

- Applications to autonomous vehicles
- Incorporate with neural MAXSAT solvers
- Certify consistency directly from deep learning component

Thank you

Artifacts available at: https://github.com/20001LastOrder/Clevr-Relational