Materializing Knowledge Bases via Trigger Graphs

Efi Tsamoura

Samsung AI, Cambridge, UK

About this talk

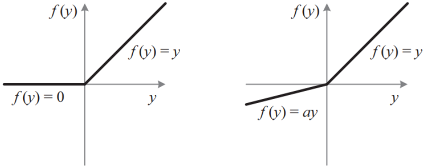
The aim is to:

- **Part I.** Motivate the use of logic and symbolic knowledge representation and reasoning techniques in developing AI applications.
- **Part II.** Present techniques to improve the efficiency of logical inference.

Part I. Motivation

Deep learning
-the successes

PReLU-nets surpass humans on classification in 2015



– PReLU networks achieve **4.94%** top-5 test error on ImageNet 2012 classification.

– Human-Level top-5 test error is **5.1%**.

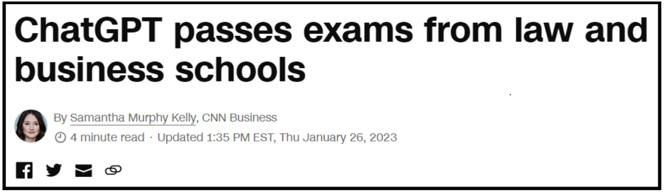
Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun. **Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification**. In ICCV, pages 1026–1034, 2015.

AlphaGo seals 4-1 victory over Go grandmaster in 2016

DeepMind's artificial intelligence astonishes fans to defeat human opponent and offers evidence computer software has mastered a major challenge

The world's top Go player, Lee Sedol, lost the final game of the Google DeepMind challenge match. Photograph: Yonhap/Reuters

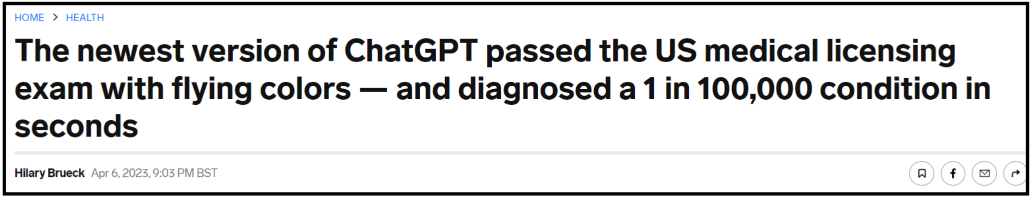
ChatGPT passes medical and law exams in 2023



ChatGPT passes exams from law and business schools

By Samantha Murphy Kelly, CNN Business

4 minute read · Updated 1:35 PM EST, Thu January 26, 2023



HOME > HEALTH

The newest version of ChatGPT passed the US medical licensing exam with flying colors — and diagnosed a 1 in 100,000 condition in seconds

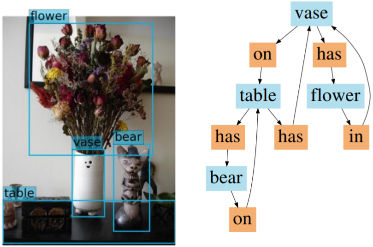
Hilary Brueck Apr 6, 2023, 9:03 PM BST

Deep learning strengths

- Pattern classification (in large).
- Learning via example.
- Tolerance to noise.

Deep learning
-the failures

Deep (vision) models are prone to biases



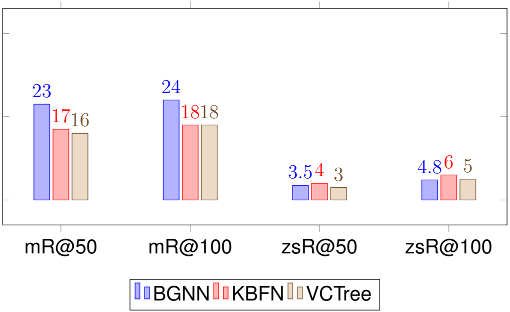


Figure: Accuracy of BGNN [11], KBFN [6] and VCTree [13] on Visual Genome [9]. Task: scene graph generation.

Large language models fail on abstract reasoning

Large Language Models Are Not Strong Abstract Reasoners

Gaël Gendron
University of Auckland
ggen187@aucklanduni.ac.nz

Qiming Bao
University of Auckland
qbao775@aucklanduni.ac.nz

Michael Witbrock
University of Auckland
m.witbrock@auckland.ac.nz

Gillian Dobbie
University of Auckland
g.dobbie@auckland.ac.nz

Abstract

Large Language Models have shown tremendous performance on a large variety of natural language processing tasks, ranging from text comprehension to common sense reasoning. However, the mechanisms responsible for this success remain opaque, and it is unclear whether LLMs can achieve human-like cognitive capabilities or whether these models are still fundamentally circumscribed. Abstract reasoning is a fundamental task for cognition, consisting of finding and applying a general pattern from few data. Evaluating deep neural architectures on this task could give insight into their potential limitations regarding reasoning and their broad generalisation abilities, yet this is currently an under-explored area. In this paper, we introduce a new benchmark for evaluating language models beyond memorization on abstract reasoning tasks. We perform extensive evaluations of state-of-the-art LLMs, showing that they currently achieve very limited performance in contrast with other natural language tasks, and we examine the reasons for this difference. We apply techniques that have been shown to improve performance on other NLP tasks and show that their impact on abstract reasoning is limited.

1 Introduction

Large Language Models (LLMs) have recently achieved impressive performance on a large variety of Natural Language Processing (NLP) tasks, including text comprehension [15, 31], commonsense reasoning [39], translation [32], and code generation [10, 8], and have shown promising results for out-of-distribution generalisation [7, 8]. The most recent and larger language models also perform well on mathematical problems, which had been out of reach for transformers for a long time [11, 38]. While empirical testing of LLMs trained on large corpora of data yields signs of high comprehension of presented problems, there is little theoretical evidence regarding why and how this performance has been achieved and whether these models are simply memorising the training data, extrapolating it, or some combination [41, 19]. A notable limitation of these models is a lack of control mechanisms, or possible misalignment [30], for which the absence of a world model or causal representation have been advanced as explanations [4, 47]. More recently, early experiments on GPT-4 showed signs of limitations on reasoning tasks requiring planning and backtracking [8]. Despite these early limitations, the question of whether or not LLMs can perform human-like reasoning remains open, as measuring the intelligence, or more broadly, the competence, of a system is a challenging task [12].

Preprint. Under review.

– LLMs have very limited performance in abstract reasoning.

– Techniques that can improve performance on other NLP tasks cannot improve the abstract reasoning capabilities of large language models.

Gaël Gendron, Qiming Bao, Michael Witbrock, Gillian Dobbie. **Large Language Models Are Not Strong Abstract Reasoners**. In Arxiv, 2023.

Deep learning weaknesses

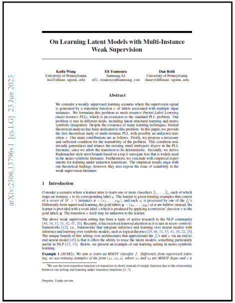
- Focus on single cognitive abilities.
- Requires large amounts of training data.
- Lacks transparency/interpretability.
- Its answers cannot be fully trusted.
- Prone to data biases.
- Difficult to incorporate background knowledge.

Gary Marcus. **Deep Learning: a Critical Appraisal**. In Arxiv, 2018.

Logic to the rescue

- Focus on ~~single~~ complex cognitive abilities.
- Requires ~~large~~ small amounts of training data.
- ~~Lacks~~ transparency/interpretability.
- Its answers can ~~not~~ be fully trusted.
- Not prone to data biases.
- ~~Difficult~~ Straighforward to incorporate background knowledge.

Via logic, we can indeed learn (deep) classifiers!



arXiv:2306.13796v1 [cs.LG] 23 Jun 2023

On Learning Latent Models with Multi-Instance Weak Supervision

Kaifu Wang
University of Pennsylvania
kaifu@sas.upenn.edu

Efi Tsamoura
Samsung AI
efi.tsamoura@samsung.com

Dan Roth
University of Pennsylvania
danroth@seas.upenn.edu

Abstract

We consider a weakly supervised learning scenario where the supervision signal is generated by a transition function σ of labels associated with multiple input instances. We formulate this problem as *multi-instance Partial Label Learning (multi-instance PLL)*, which is an extension to the standard PLL problem. Our problem is met in different fields, including latent structural learning and neuro-symbolic integration. Despite the existence of many learning techniques, limited theoretical analysis has been dedicated to this problem. In this paper, we provide the first theoretical study of multi-instance PLL with possibly an unknown transition σ. Our main contributions are as follows. Firstly, we propose a necessary and sufficient condition for the learnability of the problem. This condition non-trivially generalizes and relaxes the existing *small ambiguity degree* in the PLL literature, since we allow the transition to be deterministic. Secondly, we derive Rademacher-style error bounds based on a top-k surrogate loss that is widely used in the neuro-symbolic literature. Furthermore, we conclude with empirical experiments for learning under unknown transitions. The empirical results align with our theoretical findings; however, they also expose the issue of scalability in the weak supervision literature.

1 Introduction

Consider a scenario where a learner aims to learn one or more classifiers $f_1, \ldots, f_n$, each of which maps an instance x to its corresponding label y. The learner is given training examples that consist of a *vector* of $M > 1$ instances $\mathbf{x} = (x_1, \ldots, x_M)$, and each x_i is processed by one of the f_j's. Differently from supervised learning, the gold labels $\mathbf{y} = (y_1, \ldots, y_M)$ of $\mathbf{x}$ are *hidden*; instead, the learner is provided with a *weak* label s which is produced by applying a *transition*[1] *function* σ to the gold labels $\mathbf{y}$. The transition σ itself may be unknown to the learner.

The above weak supervision setting has been a topic of active research in the NLP community [40, 34, 33, 31, 42, 47, 20]. Recently, it has received renewed attention as it is met in *neuro-symbolic* frameworks [12], i.e., frameworks that integrate inference and learning over neural models with inference and learning over symbolic models, such as logical theories [29, 48, 11, 53, 41, 30, 22, 25]. The unique benefit of this setting over architectures that approximate the f_i's and σ via an end-to-end neural model [45] is that it offers the ability to reuse the latent models, something particularly useful in NLP [33, 31]. Below, we present an example of our learning setting in neuro-symbolic learning.

Example 1 (SUM2). *We aim to learn an MNIST classifier f. Differently from supervised learning, we use training examples of the form (x_1, x_2, s), where x_1 and x_2 are MNIST digits and s is*

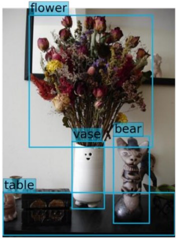
[1] We use the term transition function (or transition in short) instead of simply function due to the relationship between our setting and learning under transition functions [6, 7].

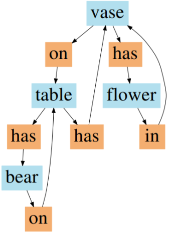
Preprint. Under review.

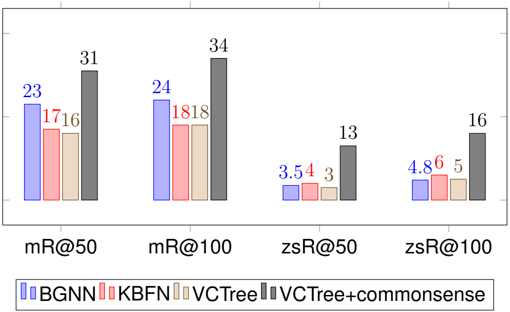
- (Deep) classifier learnability under **unknown** logical theories.
- (Deep) classifier learnability under **probabilistic** logical theories.

Kaifu Wang, and **Efthymia Tsamoura**, Dan Roth. **On learning latent models with multi-instance weak supervision**. In NeurIPS, 2023.

Via logic, we can overcome data biases!







Davide Buffelli, and **Efthymia Tsamoura**. **Scalable Theory-Driven Regularization of Scene Graph Generation Models**. In AAAI, 2023.

Scene Graph Generation (AAAI 2023)

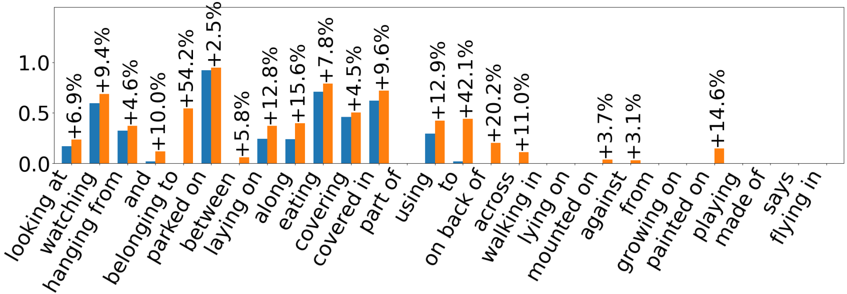
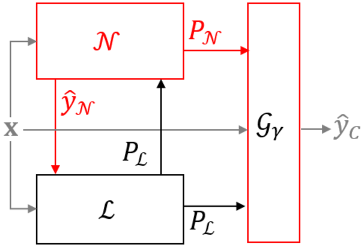


Figure: Recall of VCTree [13] on the 28 least frequent predicates: without NGP; with NGP. Benchmark: Visual Genome [9].

Knowledge Distillation into Deep Networks (ICML 2023)

Concordia

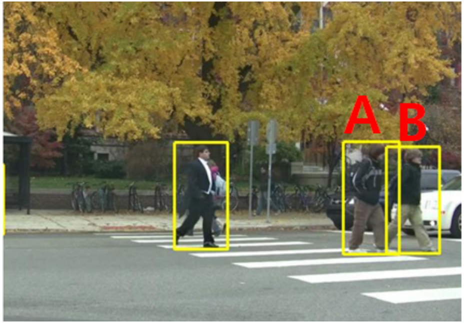


- First to support general first-order theories.
- Supports semi-/un-/supervised learning.

Operation	Equation
Inference	$\widehat{\mathbf{y}} = \arg\max_{\mathbf{y}} P_{\mathcal{N}}(\mathbf{Y} = \mathbf{y} \vert \mathbf{X} = \mathbf{x}, \boldsymbol{\theta})$
Training	$\widehat{\boldsymbol{\theta}}_{t+1} = \arg\min_{\boldsymbol{\theta}} (\ell(\widehat{\mathbf{y}}_{\mathcal{N}}, \mathbf{y}) + KL(P_{\mathcal{N}}, P_{\mathcal{L}}))$
	$\widehat{\boldsymbol{\lambda}}_{t+1} = \arg\max_{\boldsymbol{\lambda}} \prod_{(\mathbf{x}) \in \mathcal{D}} P_{\mathcal{L}}(\mathbf{X} = \mathbf{x}, \boldsymbol{\lambda}_t)$

Leon Jonathan Feldstein, Jurčius Modestas and **Efthymia Tsamoura**. **Parallel neurosymbolic integration with Concordia**. In ICML (*to appear*), 2023.

Video Activity Detection (ICML 2023)



$$\mathsf{SEQ}(B_1, B_2) \wedge \mathsf{CLOSE}(B_1, B_2) \rightarrow \mathsf{SAME}(B_1, B_2)$$

$$\mathsf{DOING}(B_1, A) \wedge \mathsf{SAME}(B_1, B_2) \rightarrow \mathsf{DOING}(B_2, A)$$

Accuracy over 5 runs

Model	Avg (%)	Max (%)	Min (%)
ACD+$\mathcal{L}$ [12]	86.00	-	-
MobileNet	90.07	91.36	89.61
IARG(MobileNet) [10]	90.18	92.39	87.55
Concordia(MobileNet, $\mathcal{L}$)	**90.73**	**93.19**	**89.54**
Inception	89.72	91.83	86.84
IARG(Inception) [10]	88.88	91.67	85.33
Concordia(Inception, $\mathcal{L}$)	**92.75**	**93.34**	**92.31**

Leon Jonathan Feldstein, Jurčius Modestas and **Efthymia Tsamoura**. **Parallel neurosymbolic integration with Concordia**. In ICML (*to appear*), 2023.

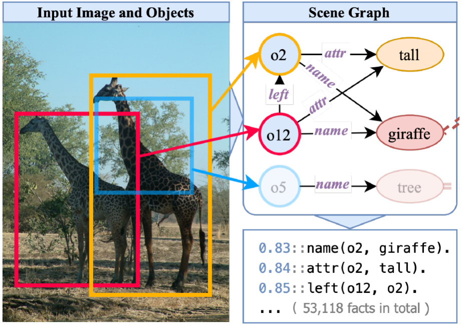
Entity Linking (ICML 2023)

Table: Results on entity linking.

Model	F_1	Acc (%)
BERT (sp)	0.88	**88.5**
Concordia(BERT) (sm)	**0.91**	**91.4**

Leon Jonathan Feldstein, Jurčius Modestas and **Efthymia Tsamoura**. **Parallel neurosymbolic integration with Concordia**. In ICML, 2023 (*to appear*).

Visual QA (SIGMOD 2023)



```
0.83::name(o2, giraffe).
0.84::attr(o2, tall).
0.85::left(o12, o2).
... ( 53,118 facts in total )
```

$$\mathsf{Q}(O) \leftarrow \mathsf{NAME}(herbivore, O)$$

$$\mathsf{NAME}(N, O) \wedge \mathsf{NAME}(N', O) \rightarrow \mathsf{ISA}(N', N)$$

$$\rightarrow \mathsf{ISA}(giraffe, herbivore)$$

$$\rightarrow \mathsf{ISA}(dear, herbivore)$$

Table: Recall@5 on VQAR [7].

Testset	LXMERT [14]	RVC [4]	TG-Guided VQA
C5	64.05%	74.62%	**87.01%**
C6	56.51%	72.04%	**85.45%**

Efthymia Tsamoura, Jaehun Lee, and Jacopo Urbani. **Probabilistic Reasoning as Scale: Trigger Graphs to the Rescue**. In SIGMOD, 2023 (*to appear*).

Part II: Reasoning at Scale via Trigger Graphs

Efthymia Tsamoura, David Carral, Enrico Malizia, and Jacopo Urbani. **Materializing Knowledge Bases via Trigger Graphs**. In VLDB, pages 943-951, 2021.

Trigger Graphs: Why

- **Key to support goal-driven QA over transitive rules**.
- **Standard bottom-up evaluation**:
  - may derive logically redundant facts;
  - may try to execute rules that derive no facts.
- **The above negatively impact the runtime and the memory.**

How: Trigger Graphs

Rules

$$r(X,Y) \rightarrow R(X,Y) \qquad (r_1)$$

$$R(X,Y) \rightarrow T(Y,X,Y) \qquad (r_2)$$

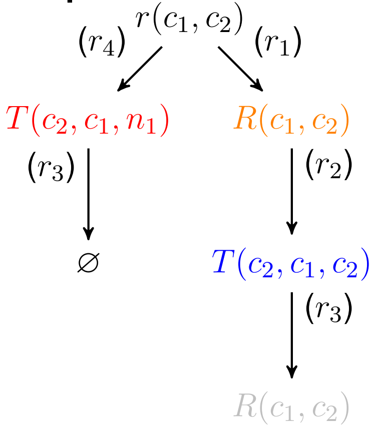
$$T(Y,X,Y) \rightarrow R(X,Y) \qquad (r_3)$$

$$r(X,Y) \rightarrow \exists Z.T(Y,X,Z) \qquad (r_4)$$

Facts

$$\rightarrow r(c_1,c_2)$$

Bottom-Up evaluation



How: Trigger Graphs

Rules

$$r(X,Y) \to R(X,Y) \qquad (r_1)$$

$$R(X,Y) \to T(Y,X,Y) \qquad (r_2)$$

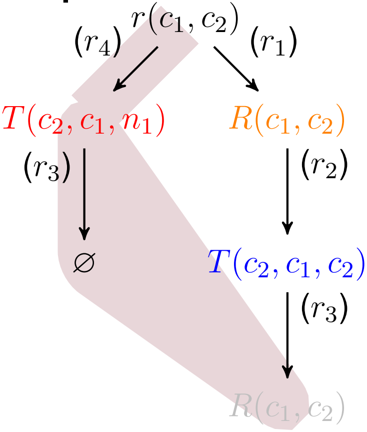
$$T(Y,X,Y) \to R(X,Y) \qquad (r_3)$$

$$r(X,Y) \to \exists Z.T(Y,X,Z) \qquad (r_4)$$

Facts

$$\to r(c_1,c_2)$$

Bottom-Up evaluation



$r(c_1,c_2)$

(r_4)

(r_1)

$T(c_2,c_1,n_1)$

$R(c_1,c_2)$

(r_3)

(r_2)

$\varnothing$

$T(c_2,c_1,c_2)$

(r_3)

$R(c_1,c_2)$

How: Trigger Graphs

Rules

$$r(X,Y) \rightarrow R(X,Y) \qquad (r_1)$$

$$R(X,Y) \rightarrow T(Y,X,Y) \qquad (r_2)$$

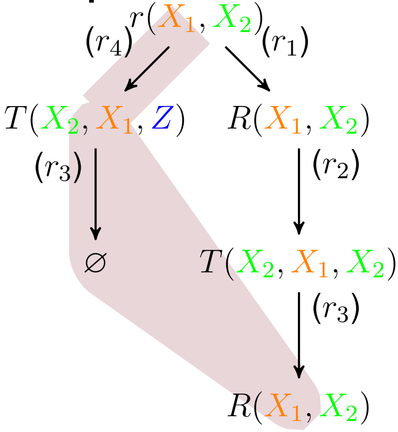
$$T(Y,X,Y) \rightarrow R(X,Y) \qquad (r_3)$$

$$r(X,Y) \rightarrow \exists Z.T(Y,X,Z) \qquad (r_4)$$

Facts

$$\rightarrow r(c_1,c_2)$$

Bottom-Up evaluation



How: Trigger Graphs

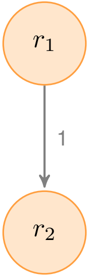
Rules

$$
\begin{aligned}
r(X,Y) &\rightarrow R(X,Y) && (r_1)\\
R(X,Y) &\rightarrow T(Y,X,Y) && (r_2)\\
T(Y,X,Y) &\rightarrow R(X,Y) && (r_3)\\
r(X,Y) &\rightarrow \exists Z.T(Y,X,Z) && (r_4)
\end{aligned}
$$

Facts

$$\rightarrow r(c_1,c_2)$$

Trigger graph



Trigger graph-based reasoning

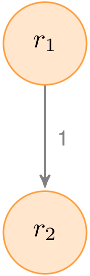
TGs delineate the rule executions

- Execute r_1 over the input instance.
- Execute r_2 over the derivations of r_1.
- **No other operation is taking place.**

Important to node

- Facts are stored inside the nodes, i.e., not stored in a single set like in all bottom-up engines.
- This data separation makes joins run faster.

Trigger graph



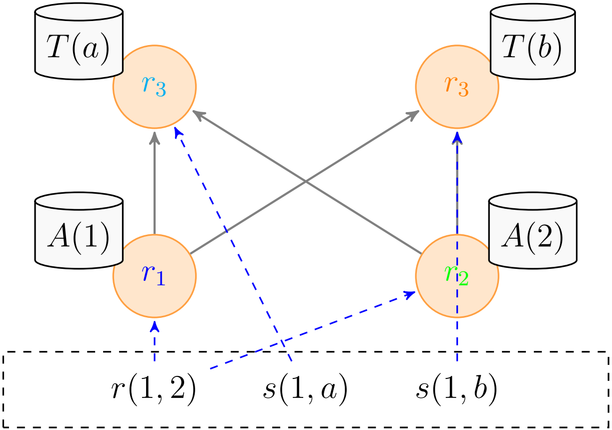
Trigger graph-based reasoning

Rules

$$r(X,Y) \to A(X) \quad (r_1)$$

$$r(X,Y) \to A(Y) \quad (r_2)$$

$$A(X) \wedge s(X,Z) \to T(Z) \quad (r_3)$$



Trigger Graphs for Linear Rules

- **Phase I: Static TG Computation**.
  - Compute a *representative* instance B^*, i.e., one that captures *all* possible rule execution paths.
  - Compute a *plan* G that mimics the rule execution when reasoning over B^*.
- **Phase II: Redundancy Elimination**.
  - Eliminate nodes that lead to redundanct facts (via detecting preserving homomorphisms).
- **Phase III: Reasoning**.
  - The computed TG can be used to reason over *all* input instances.

Trigger Graphs for Linear Rules: Complexity

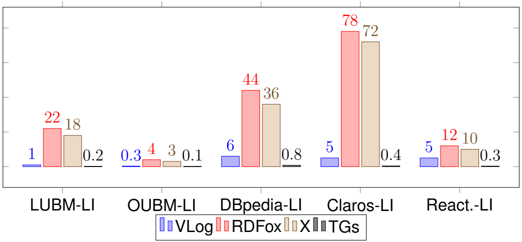
Let P be a linear program that admits a finite universal model.

Theorem (Complexity)

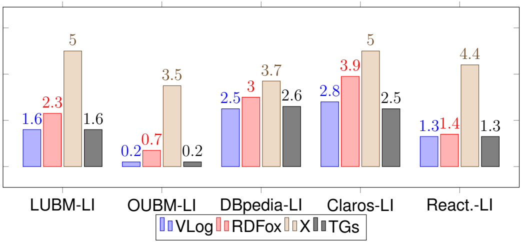
Computing a TG for P is double exponential in P. If the arity of the predicates in P is bounded, the computation time is (single) exponential.

Reasoning over Linear Rules

Total materialization times in s



Pick memory in GB



Trigger Graphs for Datalog Rules

TGs for Linear Rules

- Static TG computation.
- Use the pre-computed TG to reason over *all* instances.
- Redundancy elimination via detecting preserving homomorphisms.

TGs for Datalog Rules

- Interleave TG creation with reasoning.
- The computed TG can be used to reason over the given instance only.
- Redundancy elimination via query containment [2].

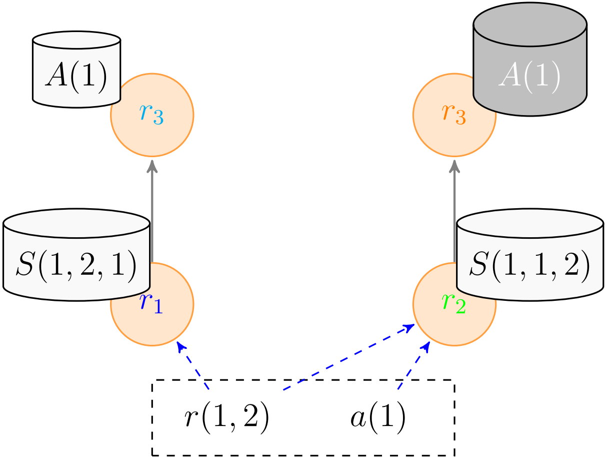
Trigger Graphs for Datalog Rules: Example

Rules

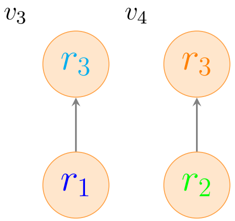
$$r(X,Y) \rightarrow S(X,Y,X) \quad (1)$$

$$a(X) \wedge r(X,Y) \rightarrow S(X,X,Y) \quad (2)$$

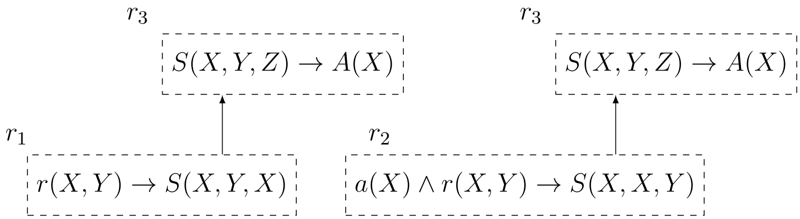
$$S(X,Y,Z) \rightarrow A(X) \quad (3)$$



Trigger Graphs for Datalog Rules: Example



Trigger Graph



r_3

$S(X, Y, Z) \to A(X)$

r_1

$r(X, Y) \to S(X, Y, X)$

r_3

$S(X, Y, Z) \to A(X)$

r_2

$a(X) \land r(X, Y) \to S(X, X, Y)$

$\boxed{Q(X) = \exists Y. r(X, Y)}$

Query for v_3

$\boxed{Q'(X) = \exists Y. a(X) \land r(X, Y)}$

Query for v_4

Trigger Graphs for Datalog Rules: Results

Let P be a Datalog program.

Theorem (Soundness)

For a TG G for P, $\mathsf{minDatalog}(G)$ is a TG for P.

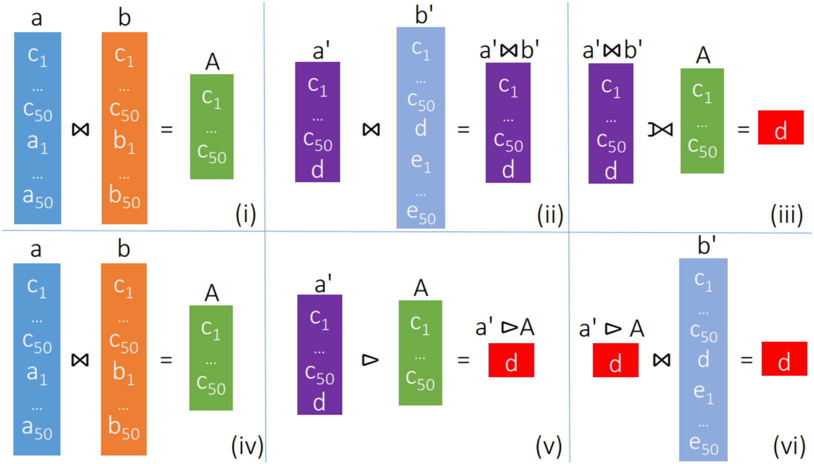
Theorem (Minimality)

Any TG for P has at least as many nodes as $\mathsf{minDatalog}(G)$.

Theorem (Complexity)

Deciding whether G is a TG of minimum size for P is co-NP-complete.

More: TG-Aware Rule Execution Strategy



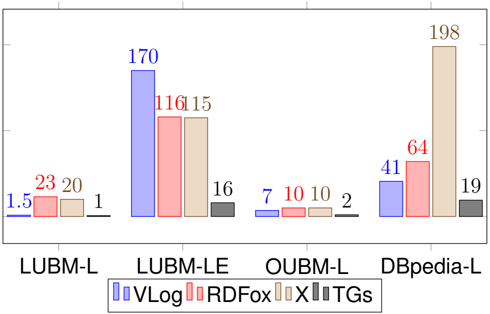
Datalog Reasoning with Trigger Graphs

	1B	2B	4B	8B	17B
Runtime (s)	203	226	520	993	2272
Memory (GB)	23	34	49	98	174
#IDPs	1B	2B	5B	10B	20B

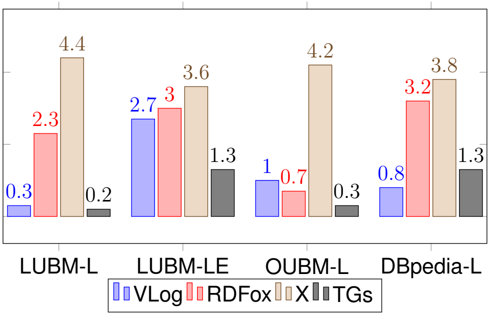
Table: Reasoning over LUBM for 1B–17B of database triples.

Datalog Reasoning with Trigger Graphs

Materialization times in s

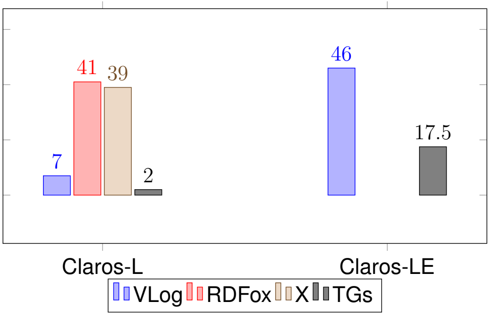


Pick memory in GB

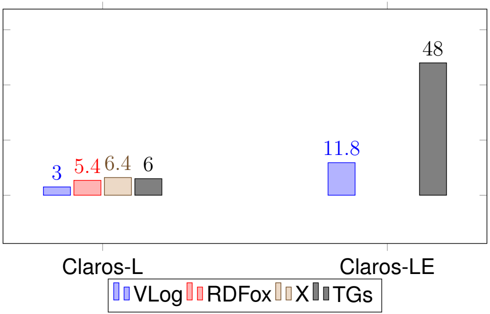


Datalog Reasoning with Trigger Graphs

Materialization times in minutes



Pick memory in GB



Reasoning at Scale: How -Lineage Trigger Graphs

Efthymia Tsamoura, Jaehun Lee, and Jacopo Urbani. **Probabilistic Reasoning as Scale: Trigger Graphs to the Rescue**. In SIGMOD, 2023 (*to appear*).

Aim

- Develop highly-scalable reasoning techniques that support uncertainty.
- Adopt well-established semantics.

Key Challenge: Complexity

Rules

$$e(X,Y) \rightarrow p(X,Y)$$

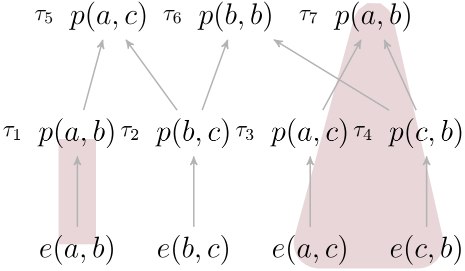
$$p(X,Z) \wedge p(Z,Y) \rightarrow p(X,Y)$$

Facts

$$\rightarrow e(a,b) \qquad \rightarrow e(a,c)$$

$$\rightarrow e(b,c) \qquad \rightarrow e(c,b)$$

Derivations



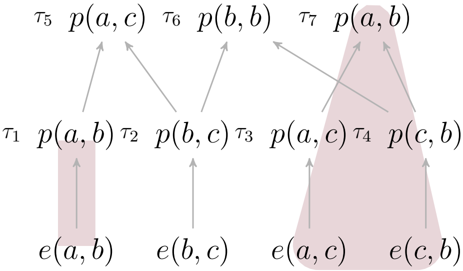
Prior Art: Key Limitations

- Relies on provenance semirings [5], i.e., associates a Boolean formula to each derivation.
  - Super-polynomial size blowup in data complexity: *any monotone formula to test connectivity in a graph with n nodes has size $n^{\Omega(\log n)}$ (lower bound holds even for undirected graphs)* [8].
- Requires Boolean checks at each reasoning step for termination.
  - Runtime bottleneck.

Efthymia Tsamoura, Victor Gutierrez-Basulto, and Angelika Kimmig. **Beyond the Grounding Bottleneck: Datalog Techniques for Inference in Probabilistic Logic Programs**. In AAAI, pages 10284-10291, 2020.

Probabilistic Reasoning via Provenance Semirings

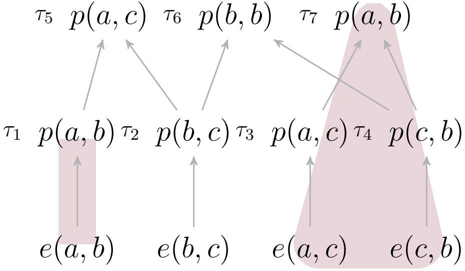
R	Derivation@R	Comparison	Formula@R
1	$e(a,b)$	$\varnothing$	$e(a,b)$
2	$e(a,c) \wedge e(c,b)$	$e(a,c) \wedge e(c,b) \overset{?}{\equiv} e(a,b)$	$e(a,c) \wedge e(c,b) \vee e(a,b)$



Lineage Trigger Graphs

- Efficient maintenance of derivation history.
  - Natural for TGs.
  - Storing pointer offsets.
- Reduces termination checks for detecting cyclic derivations!
  - No Boolean checks are required!

Derivations



Lineage Trigger Graphs: (Adaptive) Provenance Circuits

- Extended the notion of provenance circuits [3] to allow a more space-efficient reasoning:
  - Polynomial size representation.

Probabilistic Datalog Reasoning with Trigger Graphs

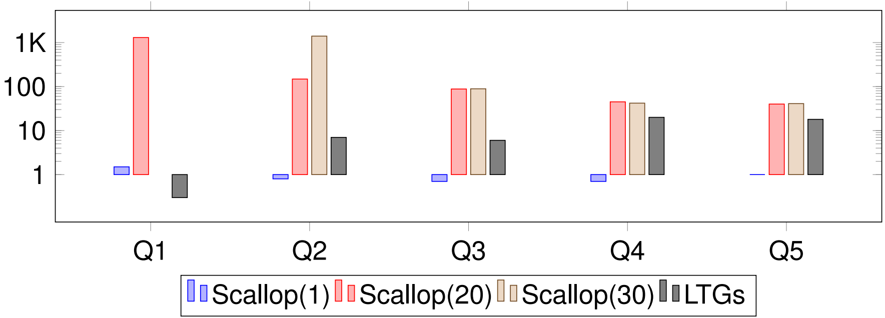


Figure: Time in seconds for goal-driven QA over sample queries from VQAR [7].

Cool Research not Covered

Goal-driven QA over existential rules with equality (AAAI 2018)

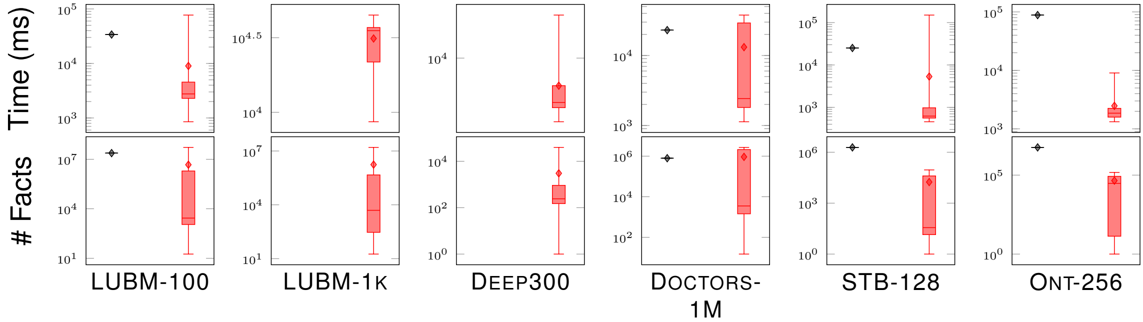


Figure: Time in msec to answer the ChaseBench queries [1].

Michael Benedikt, Boris Motik, and **Efthymia Tsamoura**. **Goal-Driven Query Answering over Existential Rules with Equality**. In AAAI, pages 1761–1770, 2018.

Thanks!

Contact info: `efi.tsamoura@samsung.com`.

References I

Michael Benedikt, George Konstantinidis, Giansalvatore Mecca, Boris Motik, Paolo Papotti, Donatello Santoro, and Efthymia Tsamoura.
Benchmarking the Chase.
In *PODS*, pages 37–52, 2017.

Ashok K. Chandra and Philip M. Merlin.
Optimal implementation of conjunctive queries in relational data bases.
In *STOC*, pages 77–90, 1977.

Daniel Deutch, Tova Milo, Sudeepa Roy, and Val Tannen.
Circuits for datalog provenance.
In *ICDT*, pages 201–212, 2014.

References II

Difei Gao, Ruiping Wang, Shiguang Shan, and Xilin Chen.
From two graphs to N questions: A VQA dataset for compositional reasoning on vision and commonsense.
CoRR, abs/1908.02962, 2019.

Todd J. Green, Grigoris Karvounarakis, and Val Tannen.
Provenance semirings.
In *PODS*, page 31–40, 2007.

Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling.
Scene graph generation with external knowledge and image reconstruction.
In *CVPR*, pages 1969–1978, 2019.

References III

- Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, and Xujie Si.
  Scallop: From probabilistic deductive databases to scalable differentiable reasoning.
  In *NeurIPS*, pages 25134–25145, 2021.

- Mauricio Karchmer and Avi Wigderson.
  Monotone circuits for connectivity require super-logarithmic depth.
  In *STOC*, page 539–550, 1988.

References IV

- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei.
  Visual genome: Connecting language and vision using crowdsourced dense image annotations.
  Int. J. Comput. Vis., 123(1):32–73, 2017.

- Zijian Kuang and Xinran Tie.
  Video understanding based on human action and group activity recognition.
  CoRR, abs/2010.12968, 2020.

References V

- Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He.
  Bipartite graph network with adaptive message passing for unbiased scene graph generation.
  In *CVPR*, pages 11109–11119, 2021.

- B. London, S. Khamis, S. H. Bach, B. Huang, L. Getoor, and L. Davis.
  Collective activity detection using hinge-loss markov random fields.
  In *CVPR Workshops*, pages 566–571, 2013.

- Knot Pipatsrisawat and Adnan Darwiche.
  New compilation languages based on structured decomposability.
  In *AAAI*, page 517–522, 2008.

References VI

Hao Tan and Mohit Bansal.
LXMERT: Learning cross-modality encoder representations from transformers.
In *EMNLP*, pages 5100–5111, 2019.