Active Learning of GAV Schema Mappings

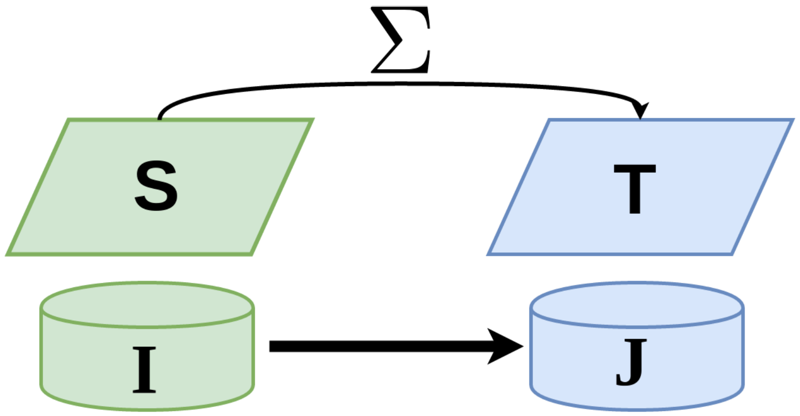
PODS 2018

Balder ten Cate[1], Phokion G. Kolaitis[2,3], **Kun Qian**[2], and Wang-Chiew Tan[4]

[1]Google Inc, [2]IBM Research – Almaden, [3]UC Santa Cruz, [4]Megagon Labs

Schema mappings and data exchange

- **Schema Mapping –** a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ describes the relationship between a source schema **S** and a target schema **T**



- **Data Exchange Problem** for a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$:
  - Given an **S**-instance $\mathbf{I}$, find a solution for $\mathbf{I}$,i.e., a **T**-instance $\mathbf{J}$ so that $(I, J) \models \Sigma$.
  - Universal solutions are the preferred solutions.

Schema-mapping languages

- The language of GLAV (Global-and-Local-As-View) – A FO logic formula of the form:

$$\forall \mathbf{x} \left(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y}) \right),$$

where $\varphi(\mathbf{x})$ is a conjunction of atoms over $\mathbf{S}$ and $\psi(\mathbf{x})$ is a conjunction of atoms over $\mathbf{T}$.

Example: $\forall s \forall c \left(\mathsf{Student}(s) \wedge \mathsf{Enroll}(s, c) \rightarrow \exists t \exists g \mathsf{Teacher}(t, c) \wedge \mathsf{Grade}(s, c, g) \right)$

- Two important sublanguages
  - **GAV (Global-As-View):** $\forall \mathbf{x}\ (\varphi(\mathbf{x}) \rightarrow T(\mathbf{x}))$:

    Example: $\forall v \forall u\ (\ \mathsf{Node}(v) \wedge \mathsf{Node}(u) \rightarrow \mathsf{Edge}(v, u)\ )$

  - **LAV (Local-As-View):** $\forall \mathbf{x}\ (S(\mathbf{x}) \rightarrow \exists y\ \psi(\mathbf{x}, \mathbf{y}))$:

    Example: $\forall v \forall u\ (\ \mathsf{Edge}(v, u) \rightarrow \exists m\ \mathsf{Edge}(v, m) \wedge \mathsf{Edge}(m, u)\ )$

Deriving schema mappings from examples

- **Problem**: derive a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ from a given set of data examples, where a data example is a pair (I,J) that satisfies Σ .

- **Learning framework** [ten Cate, Dalmau, Kolaitis 2013]
  - Exact learning algorithm for GAV with equivalence queries and labeling queries

Cast the problem as a computational learning problem.

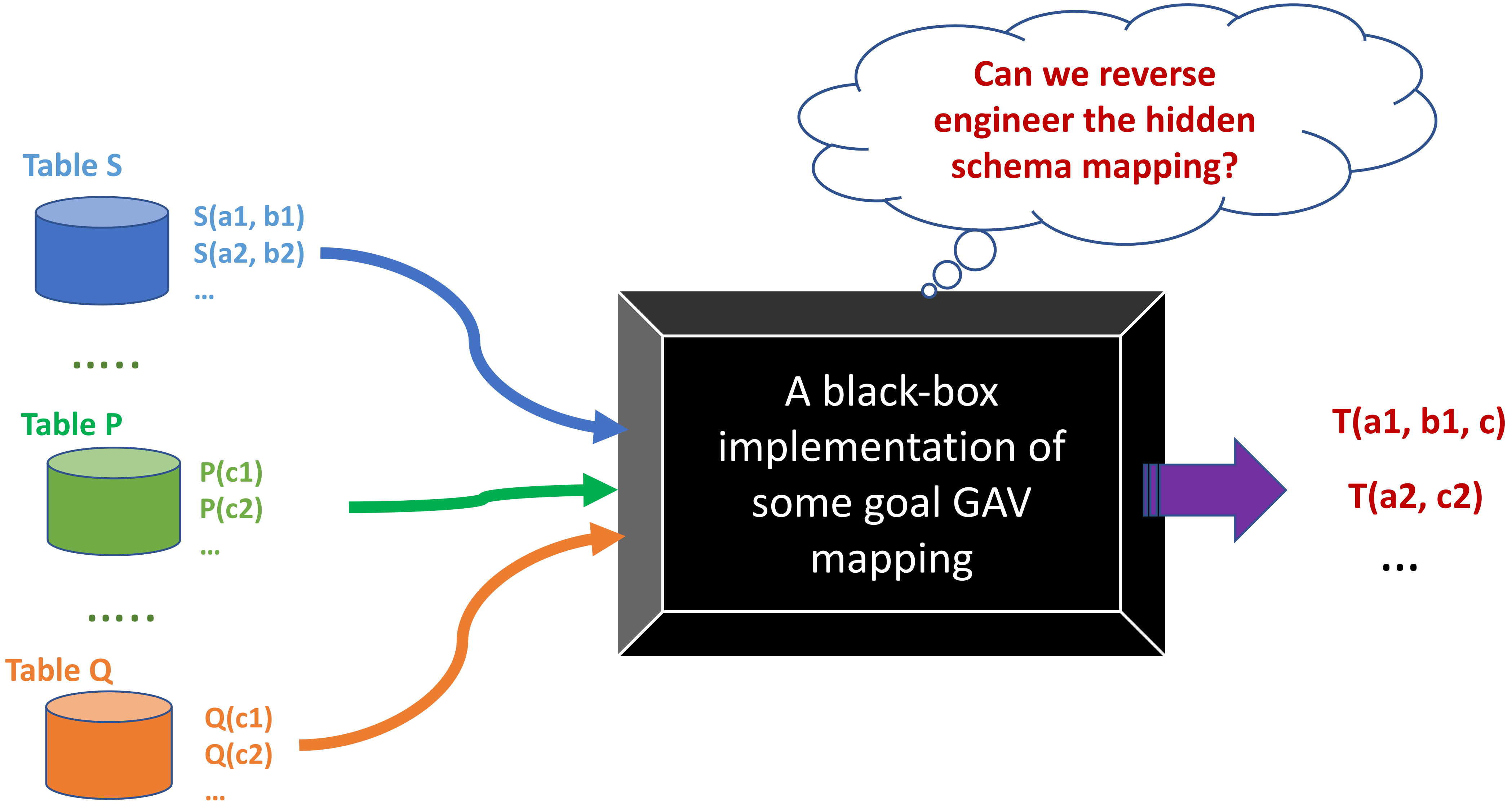
- **Other frameworks**
  - Fitting framework [Alexe, ten Cate, Kolaitis, Tan 2011]
  - Repair framework [Gottlob and Senellart, 2010]
  - Interactive Mapping Specification (IMS) [Bonifati et al., 2017]

Focused on the fitting decision problem for GLAV and GAV. Practical algorithms exist. One of our baselines

Proposed a cost model to measure the "quality" of schema Mappings. No practical algorithm for GAV

Human-in-the-loop approach. No quality guarantees for derived schema mappings

A motivating scenario: GAV reverse engineering



Learning Framework [ten Cate, Dalmau, Kolaitis 2013]

- Cast schema-mapping discovery as a computational learning problem.
  - There is a goal concept (i.e., goal mapping), whose specification needs to be discovered.

- **Theorem ([**ten Cate, Dalmau, Kolaitis 2013**]).**

GAV mappings are efficiently learnable with equivalence and labeling queries.

  - ***Equivalence oracle:*** given two GAV mappings M and M*, check if M and M* are logically equivalent. If not, a counter-example is returned.
  - ***Labeling oracle***: given a source instance I, return the canonical universal solution J for I.

ExactGAV learning algorithm

- The ExactGAV algorithm is hard to implement
  - A black-box implementation can serve as the labeling oracle.
  - However, an equivalence oracle may not be available in practice.

Contributions – the GAVLearn algorithm

- We design an active learning algorithm, called GAVLearn

- Main characteristics:
  - Adapted from the ExactGAV algorithm
  - Assumes a labeling oracle is given by a black-box implementation
  - Replaces the equivalence oracle with **conformance testing**
    - Approximate the equivalence oracle with a set of data examples
  - GAVLearn is an **active learning** algorithm
    - the learning is done by actively doing experiments with intermediate GAV mappings
  - Usefulness of GAVLearn verified through extensive experimentation.

The key ingredients of GAVLearn algorithm

Input: $\mathcal{G}$ - goal mapping (as a labeling oracle);
E - a set of universal examples for $\mathcal{G}$
Output: a mapping that fits E.

$\mathcal{H} \leftarrow \emptyset$
while true **do**
  if each $(I, J) \in E$ is canonical universal for $\mathcal{H}$ **then**
    return $\mathcal{H}$
  end if
  choose an $(I, J) \in E$ such that $J \neq \text{can-sol}_{\mathcal{H}}(I)$
  // In the proof of Thm 13, we show $\text{can-sol}_{\mathcal{H}}(I) \subsetneq J$
  $f \leftarrow$ choose a fact $f \in J \setminus \text{can-sol}_{\mathcal{H}}(I)$
  if $\mathcal{G}$ logically implies $(I, \{f\}) \times C$ for some $C \in \mathcal{H}$ **then**
    Choose $C \in \mathcal{H}$ such that $\mathcal{G}$ logically implies $(I, \{f\}) \times C$
    $\mathcal{H} \leftarrow (\mathcal{H} \setminus \{C\}) \cup \{Crit_{\mathcal{G}}((I, \{f\}) \times C)\}$
  else
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{Crit_{\mathcal{G}}((I, \{f\}))\}$
  end if
end while
return $\mathcal{H}$

Find more details of the algorithm in the paper

Input: **G** – a black-box implementation of the goal GAV mapping
E – a set of universal examples for **G**

randomly created or obtained from domain experts

Output: H – a GAV mapping that fits **E** whose size is at most the size of **G.**

- **Conformance testing**: use **E** to check if **H**'s data exchange behavior conforms to the specification of **G.**
  - **H** semantically conforms to **G** if **H** and **G** agree on **E.**
  - Otherwise, there is one $(I, J) \in$ **E**, on which **H** and **G** disagree.
- Use **active learning** to extract a **critically sound constraint** from a counterexample
  - Actively do experiments with the **counter-example** and the black-box implementation of **G.**

Theoretical Guarantees

Given a set E of universal examples for some hidden goal GAV mapping G, GAVLearn is guaranteed to produce a GAV mapping H that is **consistent with E such that size(H) <= size(G)**

Theorem 1 *GAVLearn is an optimal Occam algorithm for GAV mappings.*

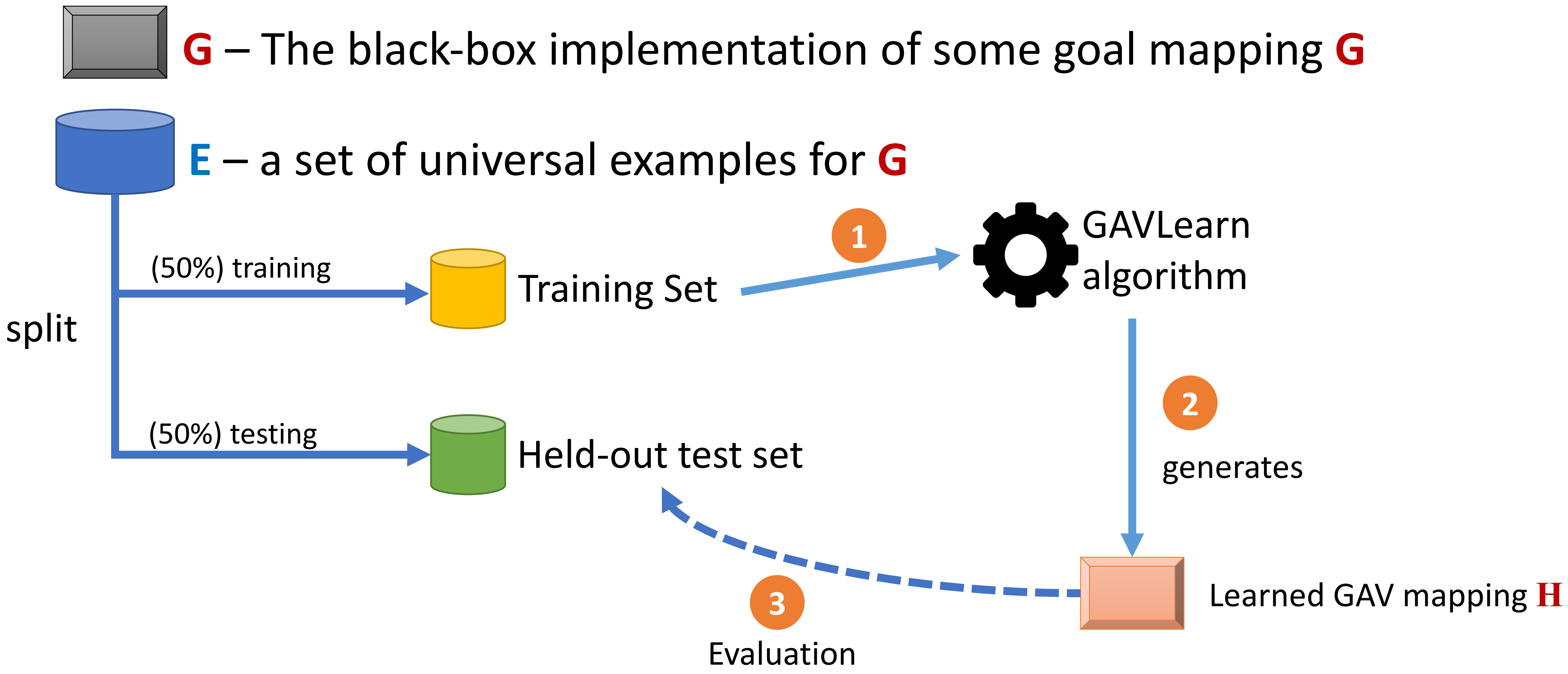
Occam learnability implies PAC learnability (Blumer et al., 1987)

Corollary 1 *GAVLearn is a PAC learning algorithm for GAV mappings*

with **sufficiently large number of random universal examples**, GAVLearn will produce a GAV mapping that is **approximately correct with high probability**

More theoretical results can be found in the paper

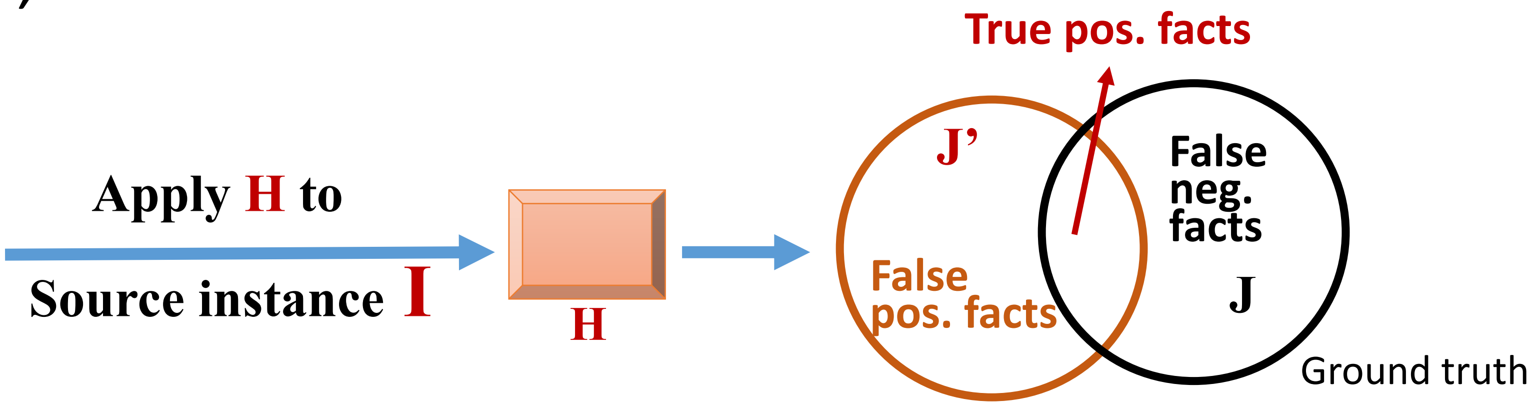
Experimental Evaluation - Methodology



Experimental Evaluation - Metrics

Precision, recall, and F-score

Given a held-out test example **(I,J)**



$$\text{Precision} = \frac{\text{\# True pos. facts}}{\text{\# True pos. facts + \# false pos. facts}}$$

We **DO NOT** report precision because the mappings returned by GAVLearn is always sound w.r.t. the goal mapping

$$\text{Recall} = \frac{\text{True pos. facts}}{\text{\# True pos. facts + \# false neg. facts}}$$

$$\text{F-score} = \frac{\text{2 x Precision x Recall}}{\text{precision + recall}}$$

Evaluation – Schema Mapping and Data Examples

- Schema mappings and data examples were generated by **iBench**
  - A metadata generator ([Arocena, Glavic, Ciucanu, Miller 2015])

- Schema mapping generation: created three types of mapping scenarios:
  - **SIMPLE, MODERATE, COMPLEX**
    using copy, merge, and projection constraints

  - Characteristics

	Copy	Projection	Merge	Join Size
SIMPLE	3	3	4	3
MODERATE	6	6	8	6
COMPLEX	9	9	12	9

- Data example generation: for each schema mapping,
  - generated {10, 30, 50, 70, 90} universal examples
  - used a 50-50 training/testing split

Evaluation – Standalone Evaluation of GAVLearn

Table. Results of GAVLearn on COMPLEX scenarios

α	n	$\|E\|$	$\overline{Comp}$	$\overline{Rep}$	$\overline{Recall}$	F-score	$\overline{\text{Time}}$
0.1	10	5	0.53±5%	0.87±10%	0.882±10%	0.937	17.8s
	30	15	0.60±0%	1	1	1	20.7s
	50	25					17.6s
	70	35					22.4s
	90	45					19.7s
0.3	10	5	0.60±0%	1	1	1	1m26s
	30	15	0.60±1%				1m35s
	50	25	0.60±0%				1m34s
	70	35	0.60±1%				1m33s
	90	45	0.60±0%	0.98±2%	0.999	0.999	1m45s
0.5	10	5	0.63±3%	0.98±2%	0.998±4‰	0.999	4m15s
	30	15	0.64±4%	0.92±5%	0.997±2‰	0.998	4m43s
	50	25	0.69±3%	0.92±4%	0.998±1%	0.999	6m55s
	70	35	0.73±4%	0.87±9%	0.998±1%	0.999	5m44s
	90	45	0.74±2%	0.88±8%	0.998±1%	0.999	10m9s

Results of SIMPLE and MODERATE schema mapping scenarios can be found in the paper

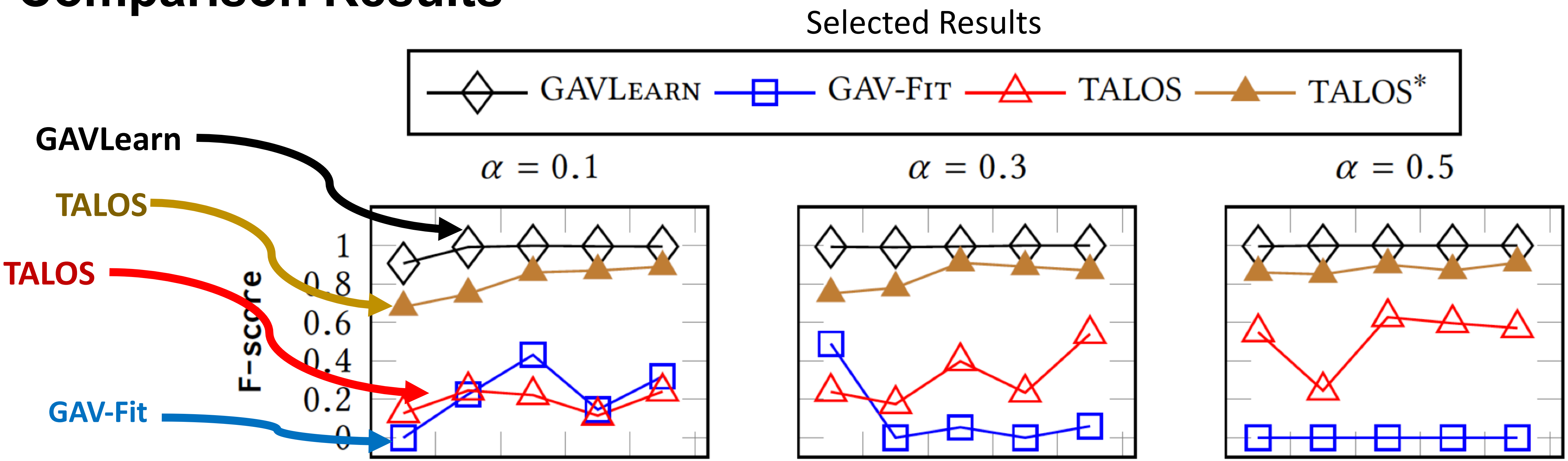
- **Highlights**
  - In all cases, F-scores are above 90% (100% in many cases)
  - Strong correspondence between number of training examples and Runtime
    - Size of training examples determines number of oracle calls

Evaluation – comparing to other algorithms

- Baselines
  - GAV-Fit: Fitting algorithm for GAV mappings [Alexe, ten Cate, Kolaitis, Tan 2011]
  - TALOS: Reverse-engineering algorithm for union of conjunctive queries [Tran et al., 2014]
    - Union of CQs can be viewed as a GAV mapping

Both algorithms have only one input:
a pre-existing set of data examples

Comparison Results



- GAVLearn outperforms both the GAV-Fit and TALOS
  - TALOS outperforms GAV-Fit, but suffers from overffiting
- Explanation: GAVLearn, being an active learning algorithm, can
  - generate "informative" examples that guide the algorithm towards better mappings
  - Both GAV-Fit and TALOS can perform significantly better if we give them the GAVLearn-generated examples.

Concluding Remarks

- Contributions:
  - Designed GAVLearn, an active learning algorithm adapted from an exact learning algorithm.
  - GAVLearn provides theoretical guarantees.
  - Experimental evaluation shows the efficacy of GAVLearn.

- Message
  - Conformance testing can be a good subsitute for an equivalence oracle.

- Open Problems
  - Exact learnability of LAV or GLAV schema mappings.
  - Active learning algorithms for LAV and GLAV.

References

- Angela Bonifati, Ugo Comignani, Emmanuel Coquery, and Romuald Thion. Interactive Mapping Specification with Exemplar Tuples. In Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017,
- A. Bonifati, E. Q. Chang, T. Ho, V. S. Lakshmanan, and R. Pottinger. HePToX: Marrying XML and Heterogeneity in Your P2P Databases. In VLDB, pages 1267-1270, 2005.
- L. M. Haas, M. A. Hernandez, H. Ho, L. Popa, and M. Roth. Clio Grows Up: From Research Prototype to Industrial Tool. In ACM SIGMOD, pages 805{810, 2005.
- R. J. Miller, L. M. Haas, and M. A. Hernandez. Schema Mapping as Query Discovery. In International Conference on Very Large Data Bases (VLDB), pages 77-88, 2000
- Balder ten Cate, Phokion Kolaitis, Kun Qian, and Wang-Chiew tan. Approximation Algorithms for Schema-Mapping Discovery from Data Examples. ACM TODS 2017.
- QT Tran et al. Query Reverse Engineering. VLDBJ 2014
- Patricia C. Arocena, Boris Glavic, Radu Ciucanu, and Renée J. Miller. The iBench Integartion Metadata Generation. PVLDB 2015
- Alselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1987. Occam's razor. Inf. Process. Lett. 24, 6 (April 1987), 377-380. DOI=http://dx.doi.org/10.1016/0020-0190(87)90114-1

Thank you

Backup

Example

$$\mathrm{M}(x,y)\wedge\mathrm{N}(y,z)\rightarrow\mathrm{Q}(x,y,z)$$
$$\mathrm{S}(x,y)\wedge\mathrm{R}(y,z)\rightarrow\mathrm{T}(x,z)$$

Goal Mapping - G

Reveal the specification for presentation purpose

(I,J)

Examples - E

$$\mathrm{I}=\{\underbrace{S(a,b),R(b,c)},\underbrace{M(a,b),N(b,c)}\},$$
$$\mathrm{J}=\{T(a,c),Q(a,b,c)\}$$

Many mappings can perfectly describe the semantics of (I,J)

$$\{\mathrm{M}(x,y)\wedge\mathrm{N}(y,z)\rightarrow\mathrm{T}(x,z),\mathrm{S}(x,y)\wedge\mathrm{R}(y,z)\rightarrow\mathrm{Q}(x,y,z)\}$$
$$\{\mathrm{M}(x,y)\wedge\mathrm{R}(y,z)\rightarrow\mathrm{T}(x,z),\mathrm{S}(x,y)\wedge\mathrm{N}(y,z)\rightarrow\mathrm{Q}(x,y,z)\}$$

…

1st iteration

Run of GAV-Learn algorithm

H = {}, and H and G disagree on (I,J)
Choose F = T(a,c) from J, and form a Counter-example (I,{F})

Active learning

I → {~~S(a,b),~~ R(b,c),M(a,b), N(b,c)} $\xrightarrow{\text{Apply G}}$ {Q(a,b,c)} **Keep S(a,b)**

I → {S(a,b), ~~R(b,c),~~ M(a,b), N(b,c)} $\xrightarrow{\text{Apply G}}$ {Q(a,b,c)} **Keep R(b,c)**

I → {S(a,b), R(b,c), ~~M(a,b)~~, N(b,c)} $\xrightarrow{\text{Apply G}}$ {T(a,c)} **Remove M(a,b)**

I → {S(a,b), R(b,c), ~~M(a,b), N(b,c)~~} $\xrightarrow{\text{Apply G}}$ {T(a,c)} **Remove N(b,c)**

S(a,b), R(b,c) $\xrightarrow{\text{T(a,c)}}$ $\mathrm{S}(x,y)\wedge\mathrm{R}(y,z)\rightarrow\mathrm{T}(x,z)$

H = { $\mathrm{S}(x,y)\wedge\mathrm{R}(y,z)\rightarrow\mathrm{T}(x,z)$ **}**

2nd iteration

H and G still disagree on (I,J). This time we choose F' = Q(a,b,c) from J, and form a Counter-example (I,{F'})

With similar computation in 1st iteration

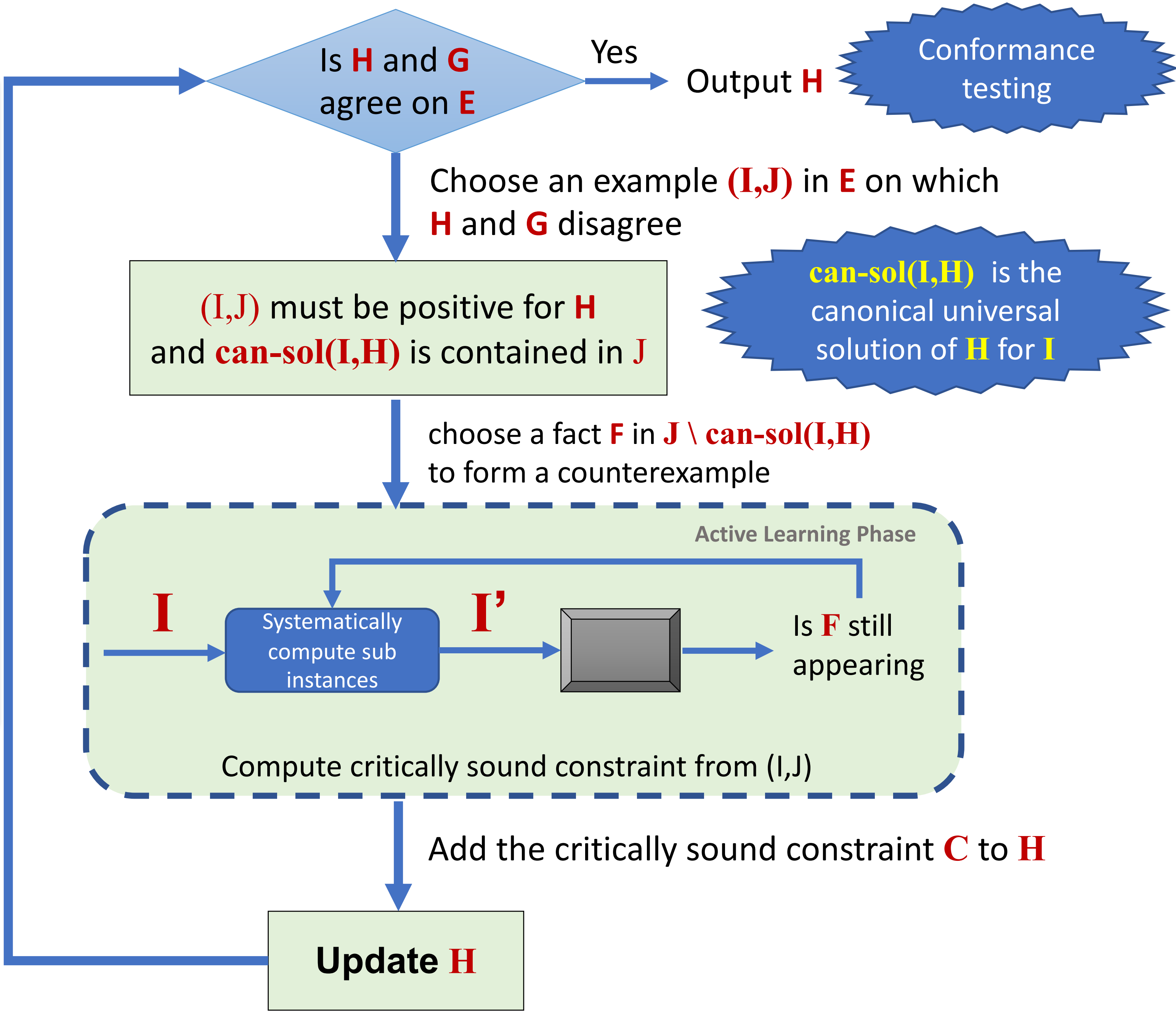
M(a,b), N(b,c) $\xrightarrow{\text{Q(a,b,c)}}$ $\mathrm{M}(x,y)\wedge\mathrm{N}(y,z)\rightarrow\mathrm{Q}(x,y,z)$

H = G

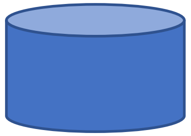
The GAV-Learn Algorithm

Input: $\mathcal{G}$ - goal mapping (as a labeling oracle);
E - a set of universal examples for $\mathcal{G}$
Output: a mapping that fits E.

$\mathcal{H} \leftarrow \emptyset$
while true **do**
  if each $(I, J) \in E$ is canonical universal for $\mathcal{H}$ **then**
    return $\mathcal{H}$
  end if
  choose an $(I, J) \in E$ such that $J \neq \text{can-sol}_{\mathcal{H}}(I)$
  `// In the proof of Thm 13, we show` $\text{can-sol}_{\mathcal{H}}(I) \subsetneq J$
  $f \leftarrow$ choose a fact $f \in J \setminus \text{can-sol}_{\mathcal{H}}(I)$
  if $\mathcal{G}$ logically implies $(I, \{f\}) \times C$ for some $C \in \mathcal{H}$ **then**
    Choose $C \in \mathcal{H}$ such that $\mathcal{G}$ logically implies $(I, \{f\}) \times C$
    $\mathcal{H} \leftarrow (\mathcal{H} \setminus \{C\}) \cup \{Crit_{\mathcal{G}}((I, \{f\}) \times C)\}$
  else
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{Crit_{\mathcal{G}}((I, \{f\}))\}$
  end if
end while
return $\mathcal{H}$



G – The black-box implementation of some goal mapping G



E – a set of universal example for G

H - the intermediate GAV mapping that tries to capture G; initially, it is empty

Occam Learnability

Definition of Occam algorithm and optimal Occam algorithm

Definition 1 *Let $\mathcal{C}$ be a concept class. An Occam algorithm for $\mathcal{C}$ with parameters $0 \leq \alpha \leq 1$ and $k \geq 1$ is an algorithm A that takes as input a collection*

$$(x_1, c(x_1)), \ldots, (x_m, c(x_m))$$

of examples labeled according to some unknown concept $c \in \mathcal{C}$ and produces a hypothesis h consistent with the input of size at most $m^\alpha n^k$, where $n = |c|$ is the size of c. If $\alpha = 0$ and $k = 1$, then A is an optimal Occam algorithm for $\mathcal{C}$

Main Theoretical Result

Theorem 1 *GAVLearn is an optimal Occam algorithm for GAV mappings.*

Given a goal GAV mapping G and a set E of (universal) examples for G, the GAVLearn is guaranteed to returns a GAV mapping H such that, H perfectly describe the semantics of E and the size of H is at most the size of G

Evaluation – Standalone evaluation of GAVLearn

Table 2: Results of GAVLEARN on simple type

α	n	$\|E\|$	$\overline{Comp}$	$\overline{Rep}$	$\overline{Recall}$	F_S	$\overline{Time}$
0.1	10	5	0.52±8%	0.83±20%	0.832±18%	0.907	0.3s
	30	15	0.6±0%	0.91±7%	0.984±1%	0.992	0.7s
	50	25	0.58±16%	0.97±3%	0.998±13%	0.999	1.1s
	70	35	0.66±5%	0.92±5%	0.992±1%	0.996	0.7s
	90	45	0.7±7%	0.89±15%	0.992±1%	0.995	0.7s
0.3	10	5	0.74±31%	0.95±6%	0.988±9%	0.993	2.2s
	30	15	0.88±8%	0.89±7%	0.982±2%	0.991	2.4s
	50	25	0.96±8%	0.96±8%	0.992±1%	0.995	2.3s
	70	35	1	1	1	1	4.2s
	90	45					2.2s
0.5	10	5	0.98±4%	0.97±4%	0.992±1%	0.996	3.4s
	30	35					3.8s
	50	45	1	1	1	1	3.6s
	70	45					4.1s
	90	45					3.8s

Table 3: Results of GAVLEARN on moderate type

α	n	$\|E\|$	$\overline{Comp}$	$\overline{Rep}$	$\overline{Recall}$	F_S	$\overline{Time}$
0.1	10	5	0.59±22%	0.98±4%	0.99±1%	0.996	4.4s
	30	15					4.9s
	50	25	0.60	1	1	1	4.2s
	70	35					4.9s
	90	45					5.4s
0.3	10	5	0.61±2%	0.98±3%	0.998±4‰	0.998	15.2s
	30	15	0.61±2%	1	1	1	16.2s
	50	25	0.65±3%	0.95±6%	0.998±1‰	0.998	25.6s
	70	35	0.63±2%	0.92±5%	0.997±2‰	0.997	13.6s
	90	45	0.65±3%	0.85±4%	0.997±1‰	0.998	18.2s
0.5	10	5	0.72±2%	0.88±8%	0.985±1‰	0.992	28.4s
	30	15	0.89±4%	0.90±3%	0.992±4‰	0.996	38.3s
	50	25	0.92±5%	0.92±5%	0.995±3‰	0.997	39s
	70	35	0.95±5%	0.95±4%	0.997±3‰	0.998	42s
	90	45	1	1	1	1	50s

Table 4: Results of GAVLEARN on complex type

α	n	$\|E\|$	$\overline{Comp}$	$\overline{Rep}$	$\overline{Recall}$	F_S	$\overline{Time}$
0.1	10	5	0.53±5%	0.87±10%	0.882±10%	0.937	17.8s
	30	15					20.7s
	50	25	0.60±0%	1	1	1	17.6s
	70	35					22.4s
	90	45					19.7s
0.3	10	5	0.60±0%				1m26s
	30	15	0.60±1%				1m35s
	50	25	0.60±0%	1	1	1	1m34s
	70	35	0.60±1%				1m33s
	90	45	0.60±0%	0.98±2%	0.999	0.999	1m45s
0.5	10	5	0.63±3%	0.98±2%	0.998±4‰	0.999	4m15s
	30	15	0.64±4%	0.92±5%	0.997±2‰	0.998	4m43s
	50	25	0.69±3%	0.92±4%	0.998±1%	0.999	6m55s
	70	35	0.73±4%	0.87±9%	0.998±1%	0.999	5m44s
	90	45	0.74±2%	0.88±8%	0.998±1%	0.999	10m9s

- **Highlights**
  - In all cases, F-scores are above 90%
    - Achieved 100% in many cases
  - More training examples lead to higher F-scores
  - Strong correspondence between number of training examples and Runtime
    - Size of training examples -> number of oracle calls