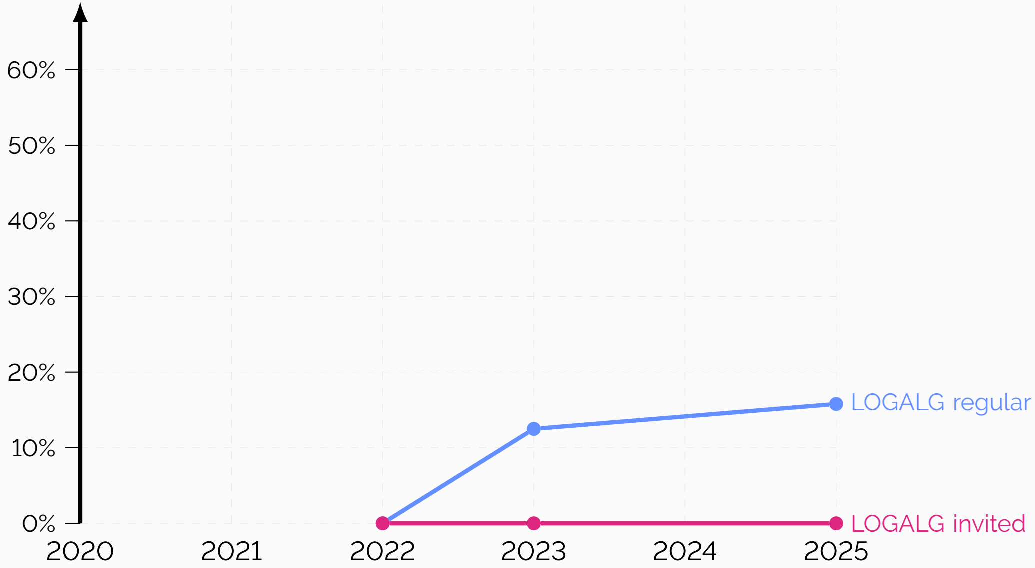
The Parameterized Complexity of Learning Monadic Second-Order Logic

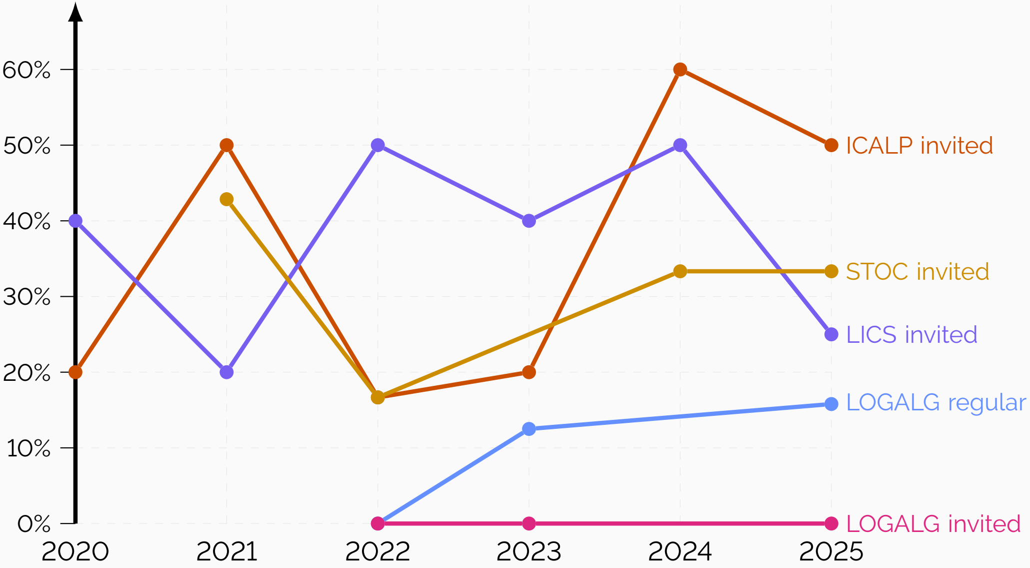
Steffen van Bergerem, Martin Grohe, and Nina Runde

Workshop on Logic, Graphs, and Algorithms 2025

Women in Logic, Graphs, and Algorithms



Women in Logic, Graphs, and Algorithms



Women in Logic, Graphs, and Algorithms

Sandra Kiefer

descriptive complexity theory

Weisfeiler–Leman algorithm

Nicole Schweikardt

counting logics

FO enumeration for nowhere dense classes

Isolde Adler

nowhere dense = stable = dependent

graph decompositions

Nina Runde

MSO learning

homomorphism reconstructibility

The Parameterized Complexity of Learning Monadic Second-Order Logic

Steffen van Bergerem, Martin Grohe, and Nina Runde

Workshop on Logic, Graphs, and Algorithms 2025

Algorithmic Problems

Model Checking

Given a graph G and a sentence φ

Decide whether $G \models \varphi$

Algorithmic Problems

Model Checking

Given a graph G and a sentence φ

Decide whether $G \models \varphi$

Counting

Given a graph G and a formula $\varphi(\bar{x})$

Output number of tuples $\bar{v}$ in $V(G)$ with $G \models \varphi(\bar{v})$

Algorithmic Problems

Model Checking

Given a graph G and a sentence φ

Decide whether $G \models \varphi$

Counting

Given a graph G and a formula $\varphi(\bar{x})$

Output number of tuples $\bar{v}$ in $V(G)$ with $G \models \varphi(\bar{v})$

Enumeration

Given a graph G and a formula $\varphi(\bar{x})$

Enumerate all tuples $\bar{v}$ with $G \models \varphi(\bar{v})$

Learning

Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$

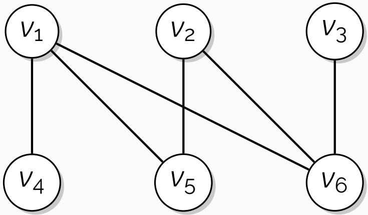
Output a formula $\varphi(x)$
such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Learning

Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$

Output a formula $\varphi(x)$
such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$



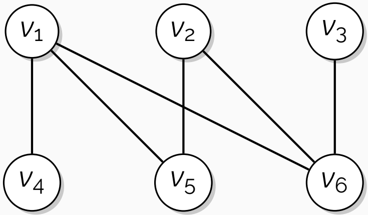
$$S_+ = \{v_1, v_3\}$$
$$S_- = \{v_4, v_5\}$$

Learning

Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$

Output a formula $\varphi(x)$
such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$



$$S_+ = \{v_1, v_3\}$$
$$S_- = \{v_4, v_5\}$$

Output $\varphi(x) = x{=}v_1 \lor x{=}v_3$

Learning

MSO Consistent Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Reject if there is no such formula

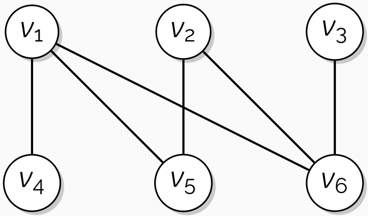
Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Reject if there is no such formula



$S_+ = \{v_1, v_3\}$

$S_- = \{v_4, v_5\}$

$\ell = 1$

$q = 3$

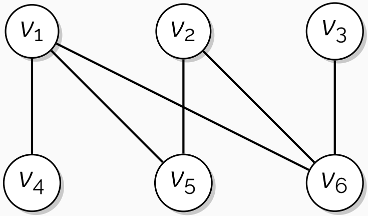
Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Reject if there is no such formula



$S_+ = \{v_1, v_3\}$

$S_- = \{v_4, v_5\}$

$\ell = 1$

$q = 3$

Output $\varphi(x) = \exists Y\ (\text{bipartite}(Y) \wedge Y(v_1) \wedge Y(x))$

Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Reject if there is no such formula

v. B., Grohe, and Runde, CSL 2025

In general, the MSO consistent-learning problem is **para-NP-hard**.

v. B., Grohe, and Runde, CSL 2025

The MSO consistent-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**.

Learning

MSO Consistent Learning

Given a graph G, sets of vertices $S_+, S_- \subseteq V(G)$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants such that $G \models \varphi(v)$ for all $v \in S_+$ and $G \not\models \varphi(v)$ for all $v \in S_-$

Reject if there is no such formula

v. B., Grohe, and Runde, CSL 2025

In general, the MSO consistent-learning problem is **para-NP-hard**.

v. B., Grohe, and Runde, CSL 2025

There is an algorithm that solves the MSO consistent-learning problem in time $f(\ell, q, c) \cdot |V(G)|$, where c is the clique-width of G and f is a computable function.

Proof idea

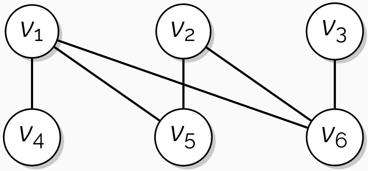
v. B., Grohe, and Runde, CSL 2025

The MSO consistent-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**.

Proof idea

v. B., Grohe, and Runde, CSL 2025

The MSO consistent-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**.



$S_+ = \{v_1, v_3\}$

$S_- = \{v_4, v_5\}$

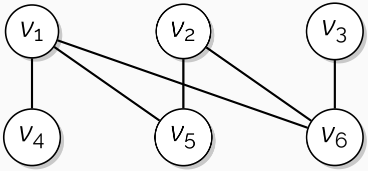
$\ell = 1$

$q = 3$

Proof idea

v. B., Grohe, and Runde, CSL 2025

The MSO consistent-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**.



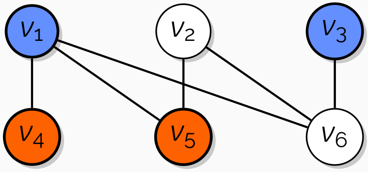
$$S_+ = \{v_1, v_3\}$$
$$S_- = \{v_4, v_5\}$$
$$\ell = 1$$
$$q = 3$$

- with placeholders instead of constants, the number of formulas only depends on ℓ, q $\longrightarrow$ try all of them

Proof idea

v. B., Grohe, and Runde, CSL 2025

The MSO consistent-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**.



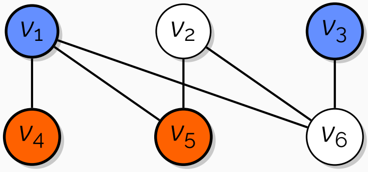
$$S_+ = \{v_1, v_3\}$$
$$S_- = \{v_4, v_5\}$$
$$\ell = 1$$
$$q = 3$$

- with placeholders instead of constants, the number of formulas only depends on $\ell, q \longrightarrow$ try all of them
- encode examples using two colours S_+, S_-

Proof idea

v. B., Grohe, and Runde, CSL 2025

The MSO consistent-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**.



$S_+ = \{v_1, v_3\}$
$S_- = \{v_4, v_5\}$
$\ell = 1$
$q = 3$

- with placeholders instead of constants, the number of formulas only depends on $\ell, q \longrightarrow$ try all of them
- encode examples using two colours S_+, S_-
- for formula $\varphi(x, y_1, \ldots, y_\ell)$, check
  $G \models \exists y_1 \ldots \exists y_\ell \forall x \Big((S_+(x) \to \varphi(x, \bar{y})) \land (S_-(x) \to \neg\varphi(x, \bar{y})) \Big)$
- model-checking is fixed-parameter linear for bounded clique-width

Higher-Dimensional Learning

k-Dimensional MSO Consistent Learning

Given a graph G, sets of tuples $S_+, S_- \subseteq (V(G))^k$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x_1, \ldots, x_k)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants
such that $G \models \varphi(\bar{v})$ for all $\bar{v} \in S_+$ and $G \not\models \varphi(\bar{v})$ for all $\bar{v} \in S_-$

Reject if there is no such formula

Higher-Dimensional Learning

k-Dimensional MSO Consistent Learning

Given a graph G, sets of tuples $S_+, S_- \subseteq (V(G))^k$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x_1, \ldots, x_k)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants
such that $G \models \varphi(\bar{v})$ for all $\bar{v} \in S_+$ and $G \not\models \varphi(\bar{v})$ for all $\bar{v} \in S_-$

Reject if there is no such formula

v. B., Grohe, and Runde, CSL 2025

There is an algorithm that solves the k-dimensional MSO consistent-learning problem in time $(m+1)^{f(k,\ell,q,c)} \cdot |V(G)|^2$, where c is the clique-width of G, $m = |S_+ \cup S_-|$, and f is a computable function.

Higher-Dimensional Learning

k-Dimensional MSO Consistent Learning

Given a graph G, sets of tuples $S_+, S_- \subseteq (V(G))^k$, and $\ell, q \in \mathbb{N}$

Output an MSO formula $\varphi(x_1, \ldots, x_k)$ with $qr(\varphi) \leqslant q$ using at most ℓ constants
such that $G \models \varphi(\bar{v})$ for all $\bar{v} \in S_+$ and $G \not\models \varphi(\bar{v})$ for all $\bar{v} \in S_-$

Reject if there is no such formula

v. B., Grohe, and Runde, CSL 2025

There is an algorithm that solves the k-dimensional MSO consistent-learning problem in time $(m+1)^{f(k,\ell,q,c)} \cdot |V(G)|^2$, where c is the clique-width of G, $m = |S_+ \cup S_-|$, and f is a computable function.

v. B., Grohe, and Runde, CSL 2025

This is optimal.

PAC Learning

- Probably Approximately Correct Learning
- assume probability distribution on $(V(G))^k \times \{+,-\}$

PAC Learning

- Probably Approximately Correct Learning
- assume probability distribution on $(V(G))^k \times \{+,-\}$
- algorithm draws examples from the distribution
- **Goal:** return formula with small expected error

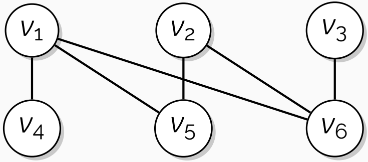
PAC Learning

- Probably Approximately Correct Learning
- assume probability distribution on $(V(G))^k \times \{+,-\}$
- algorithm draws examples from the distribution
- **Goal:** return formula with small expected error

v. B., Grohe, and Runde, CSL 2025

The MSO PAC-learning problem is **fixed-parameter linear** on classes of **bounded clique-width**, even in **higher dimensions** k.

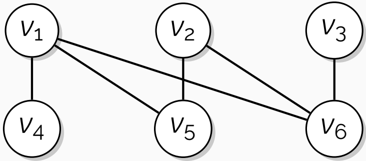
Main Results



$S_+ = \{v_1, v_3\}$
$S_- = \{v_4, v_5\}$
$\ell = 1$
$q = 3$

1. **In general**, the MSO consistent-learning problem is **para-NP-hard**.

Main Results



$S_+ = \{v_1, v_3\}$
$S_- = \{v_4, v_5\}$
$\ell = 1$
$q = 3$

1. **In general**, the MSO consistent-learning problem is **para-NP-hard**.

On classes of **bounded clique-width**:

2. The MSO **consistent-learning** problem is **fixed-parameter linear**.
3. The k-dim. consistent-learning problem can be solved in time quadratic in the size of the graph, but XP in the number of examples. This is optimal.
4. The MSO **PAC-learning** problem is **fixed-parameter linear**, even in **higher dimensions**.