Verifying Graph Neural Networks

François Schwarzentruber

ÉCOLE **NORMALE**
SUPÉRIEURE
DE **LYON**

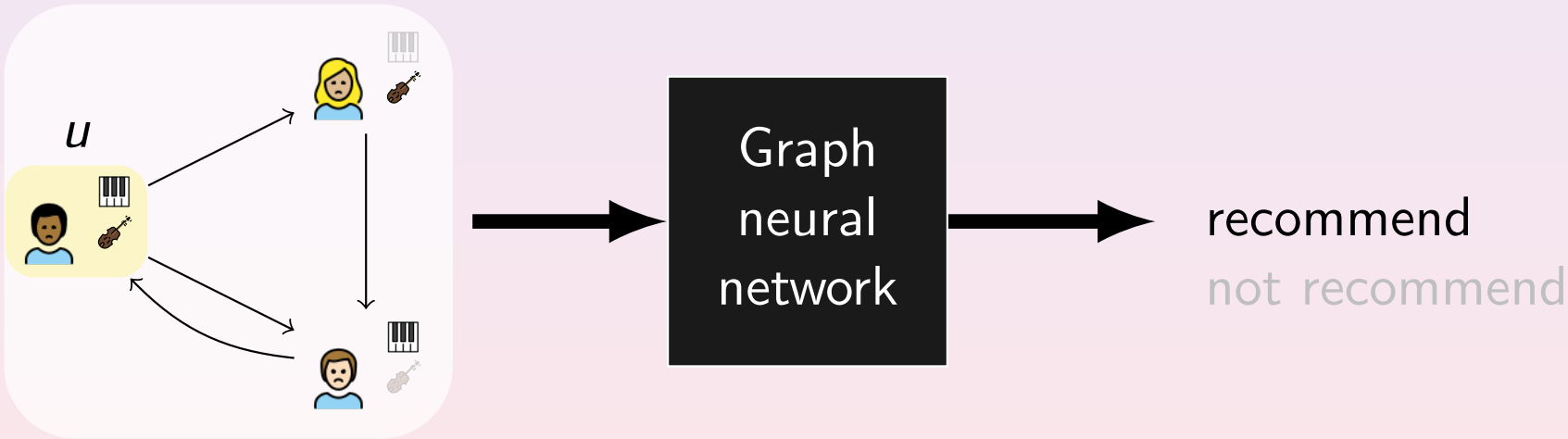
LORI 2025 - Xi'an

Lecture notes

https://arxiv.org/abs/2510.11617

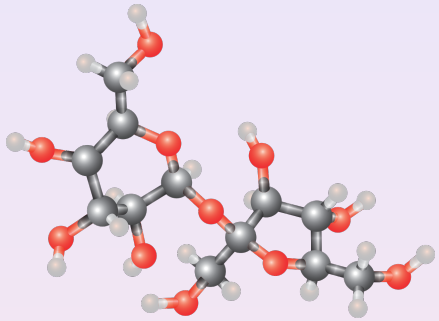
Motivation

A GNN = a machine learning model for classifying graphs/vertices



labeled pointed graph

Applications



- Toxicity of a molecule [Reiser et al. 2022]
- Drug discovery [Xiong et al., 2021]
- Recommendation in social network [Salamat et al., 2021]
- Voice segmentation in music scores
  [Karystinaios et al., IJCAI 2023]
- Link prediction etc. in knowledge graphs [Ye et al. 2022]
- Heuristics in epistemic planning [Briglia et al. 2025]

GNNs = Blackboxes

We need:

- 💬 Explanations on decisions made
- 💡 Make GNNs more interpretable
- ✔ Guarantees

$\rightsquigarrow$ research on the interaction logic$\leftrightarrow$GNNs

Some references

- GNNs and Weisfeiler-Leman tests: [Grohe LICS 2021]
- GNNs and graded modal logic [Barceló et al. ICLR 2020]
- Verification [Nunn et al. 2023] [Nunn et al. IJCAI 2024] [Benedikt et al. ICALP 2024] [Sälzer et al. IJCAI 2025]
- Explosion of connections logic↔GNNs:
  - mu-calculus and recurrent GNNs [Ahvonen et al. NeurIPS 2024]
  - standard modal logic and mean-GNNs [Schönherr et al. 2025]
  - datalog and max-GNNs [Cucala et al. KR 2024]

etc.

Outline

Outline

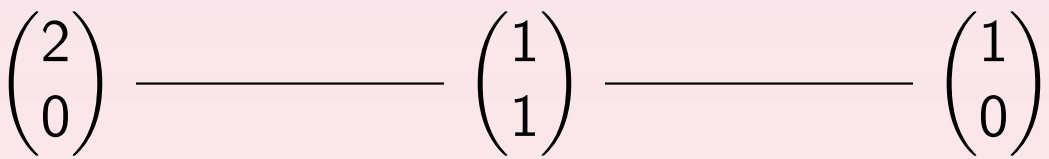
Labelled graphs

Definition

A *labelled graph* is (V, E, ℓ) where:

- V is a set of vertices;
- E is a set of edges;
- $\ell : V \to \mathbb{Q}^d$ is a labelling function.

$\mathbb{Q}$ = set of rational numbers
d = dimension of vectors



Definition of a graph neural network

A GNN is an algorithm $\mathcal{A}$ of the form:

input: a labelled pointed graph (G, u, ℓ_0)
output: yes/no
function $\mathcal{A}(G, u, \ell_0)$
$\quad \ell_1 := \mathit{layer}_1(G, \ell_0)$
$\quad \vdots$
$\quad \ell_L := \mathit{layer}_L(G, \ell_{L-1})$
$\quad$ **return** yes **if** $w^t \ell_L(u) + b \geq 0$ **else** no

Example of a GNN layer

Previous labelling: $\begin{pmatrix}2\\0\end{pmatrix}$ —— $\begin{pmatrix}1\\1\end{pmatrix}$ —— $\begin{pmatrix}1\\0\end{pmatrix}$

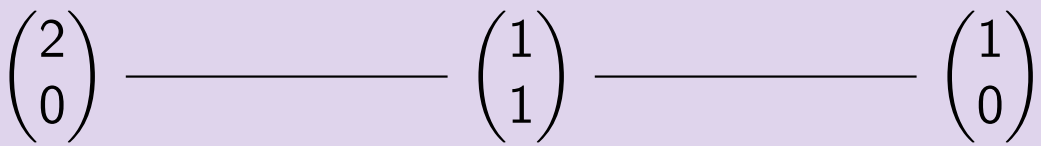
Example of a GNN layer

Previous labelling: $\binom{2}{0}$ —— $\binom{1}{1}$ —— $\binom{1}{0}$

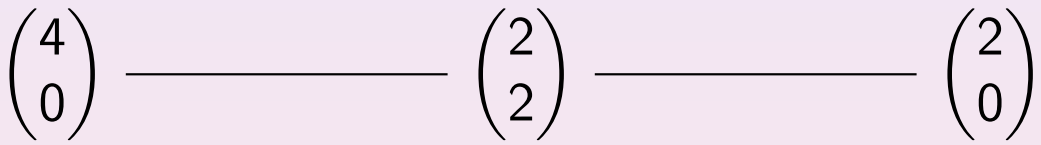
Multiply the current vector by 2

Example of a GNN layer

Previous labelling:



Multiply the current vector by 2



Example of a GNN layer

Previous labelling: $\binom{2}{0}$ —— $\binom{1}{1}$ —— $\binom{1}{0}$

↓ Multiply the current vector by 2

$\binom{4}{0}$ —— $\binom{2}{2}$ —— $\binom{2}{0}$

↓ Add $-1\times$ the sum of its neighbor vectors

Example of a GNN layer

Previous labelling: $\begin{pmatrix}2\\0\end{pmatrix}$ —— $\begin{pmatrix}1\\1\end{pmatrix}$ —— $\begin{pmatrix}1\\0\end{pmatrix}$

Multiply the current vector by 2

$\begin{pmatrix}4\\0\end{pmatrix}$ —— $\begin{pmatrix}2\\2\end{pmatrix}$ —— $\begin{pmatrix}2\\0\end{pmatrix}$

Add $-1\times$ the sum of its neighbor vectors

$\begin{pmatrix}3\\-1\end{pmatrix}$ —— $\begin{pmatrix}-1\\2\end{pmatrix}$ —— $\begin{pmatrix}1\\-1\end{pmatrix}$

Example of a GNN layer

Previous labelling: $\begin{pmatrix}2\\0\end{pmatrix}$ —— $\begin{pmatrix}1\\1\end{pmatrix}$ —— $\begin{pmatrix}1\\0\end{pmatrix}$

↓ Multiply the current vector by 2

$\begin{pmatrix}4\\0\end{pmatrix}$ —— $\begin{pmatrix}2\\2\end{pmatrix}$ —— $\begin{pmatrix}2\\0\end{pmatrix}$

↓ Add $-1\times$ the sum of its neighbor vectors

$\begin{pmatrix}3\\-1\end{pmatrix}$ —— $\begin{pmatrix}-1\\2\end{pmatrix}$ —— $\begin{pmatrix}1\\-1\end{pmatrix}$

↓ Replace negative values by 0

Example of a GNN layer

Previous labelling: $\begin{pmatrix}2\\0\end{pmatrix}$ —— $\begin{pmatrix}1\\1\end{pmatrix}$ —— $\begin{pmatrix}1\\0\end{pmatrix}$

↓ Multiply the current vector by 2

$\begin{pmatrix}4\\0\end{pmatrix}$ —— $\begin{pmatrix}2\\2\end{pmatrix}$ —— $\begin{pmatrix}2\\0\end{pmatrix}$

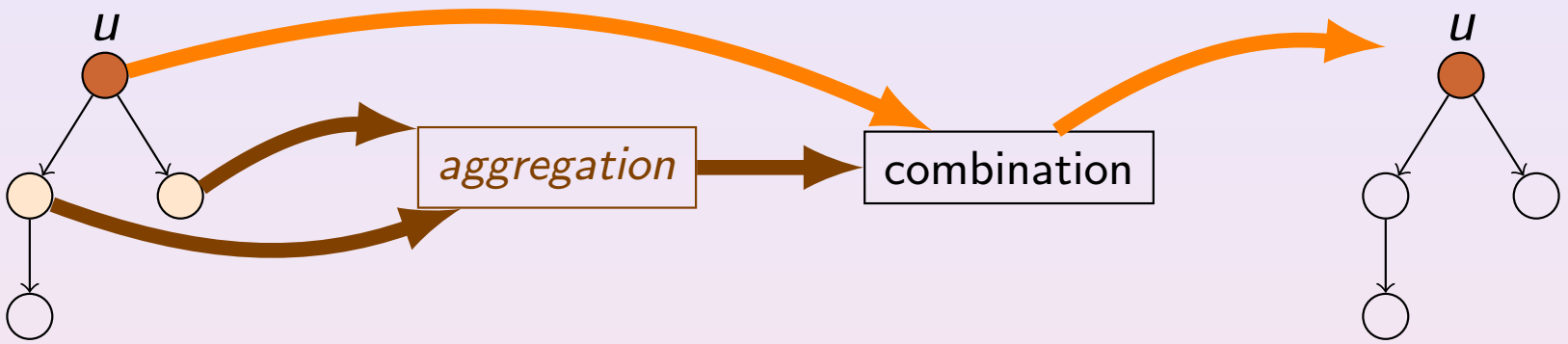
↓ Add $-1\times$ the sum of its neighbor vectors

$\begin{pmatrix}3\\-1\end{pmatrix}$ —— $\begin{pmatrix}-1\\2\end{pmatrix}$ —— $\begin{pmatrix}1\\-1\end{pmatrix}$

↓ Replace negative values by 0

New labelling: $\begin{pmatrix}3\\0\end{pmatrix}$ —— $\begin{pmatrix}0\\2\end{pmatrix}$ —— $\begin{pmatrix}1\\0\end{pmatrix}$

Each layer

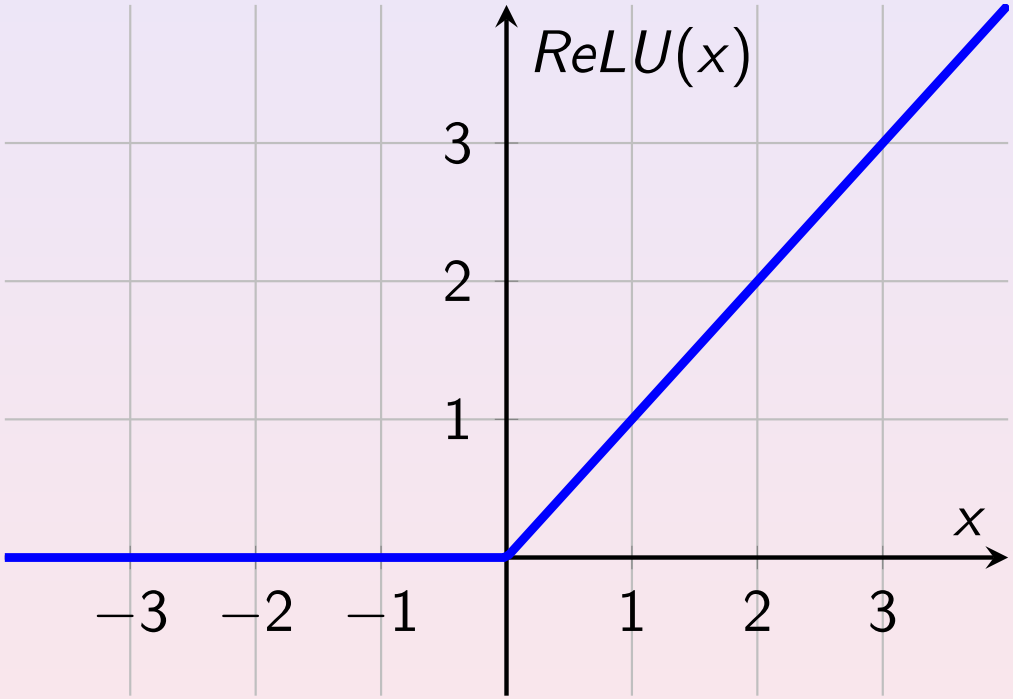


function $layer_i(V, E, \ell)$
  $\ell' :=$ new labelling $V \to \mathbb{Q}^d$
  for vertices $u \in V$ **do**
    $aggregation := \sum\{\!\{\ell[v] \mid v \in E(u)\}\!\}$
    $\ell'[u] := \vec{\sigma}(A_i \times \ell[u] + B_i \times aggregation + b_i)$
  return ℓ'

where $A_i, B_i \in \mathbb{Q}^{d \times d}$ and $b_i \in \mathbb{Q}^d$.

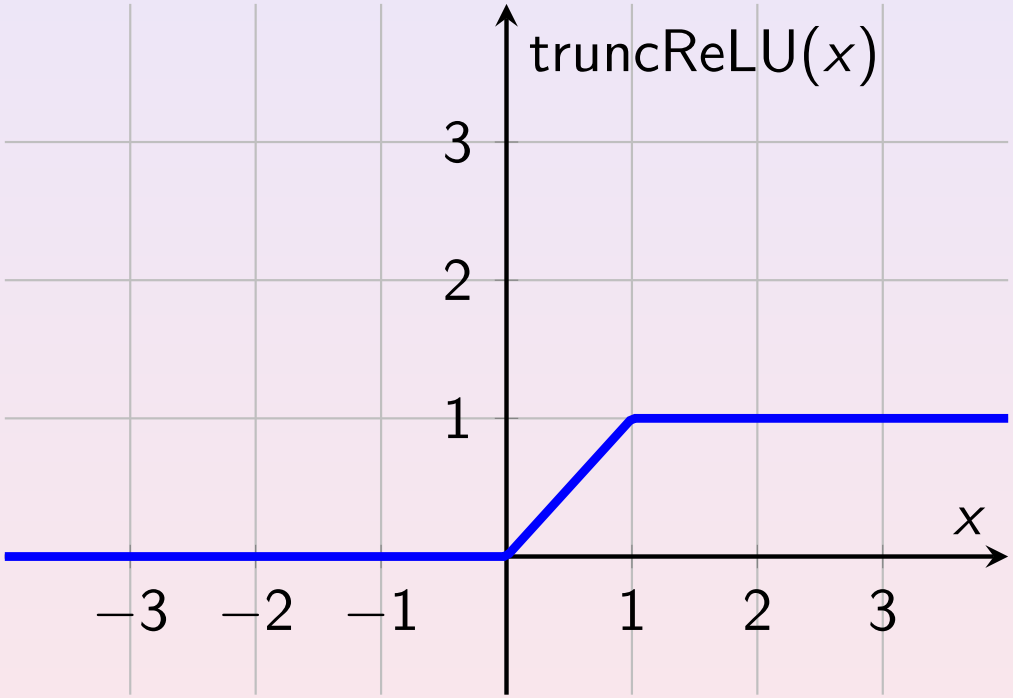
Outline

ReLU activation function



$$ReLU(x) = \max(0, x)$$

TruncReLU activation function



$$\text{truncReLU}(x) = \max(0, \min(1, x))$$

Outline

Syntax and semantics

Syntax of GNN expressions [Sälzer et al., IJCAI 2025]

$$\vartheta ::= c \mid x_i \mid \sigma(\vartheta) \mid agg(\vartheta) \mid \vartheta + \vartheta \mid c \times \vartheta$$

Semantics

$$\begin{aligned}
[\![c]\!]_{G,u} &= c, \\
[\![x_i]\!]_{G,u} &= \ell(u)_i, \\
[\![\vartheta + \vartheta']\!]_{G,u} &= [\![\vartheta]\!]_{G,u} + [\![\vartheta']\!]_{G,u}, \\
[\![c \times \vartheta]\!]_{G,u} &= c \times [\![\vartheta]\!]_{G,u}, \\
[\![\sigma(\vartheta)]\!]_{G,u} &= [\![\sigma]\!]([\![\vartheta]\!]_{G,u}), \\
[\![agg(\vartheta)]\!]_{G,u} &= \Sigma_{v|uEv} [\![\vartheta]\!]_{G,v},
\end{aligned}$$

GNN expressions capture GNNs!

Proposition

Given a GNN $\mathcal{A}$, there exists an expression ϑ such that

$$(G, u) \in [\![\mathcal{A}]\!] \quad \textit{iff} \quad [\![\vartheta \geq 1]\!]_{G,u} = \textit{true}.$$

Example

- $\mathcal{A}^{(0)}(G, u) = \ell_0(G, u)$;
- $\mathcal{A}^{(1)}(G, u) = \vec{\sigma}\left(\begin{pmatrix} 2 & 1 \\ -1 & 4 \end{pmatrix} \times \mathcal{A}^{(0)}(G, u) + \begin{pmatrix} 5 & 3 \\ 2 & 6 \end{pmatrix} \times \sum\{\!\{\mathcal{A}^{(0)}(G, v) \mid v \in E(u)\}\!\} + \begin{pmatrix} 1 \\ -2 \end{pmatrix}\right)$
- $\mathcal{A}^{(2)}(G, u) = \vec{\sigma}\left(\begin{pmatrix} 3 & 0 \\ -2 & 0 \end{pmatrix} \times \mathcal{A}^{(1)}(G, u) + \begin{pmatrix} -1 & 0 \\ 0 & 5 \end{pmatrix} \times \sum\{\!\{\mathcal{A}^{(1)}(G, v) \mid v \in E(u)\}\!\} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)$

$$\begin{aligned}
\psi_1 &= \sigma(2x_1 + x_2 + 5agg(x_1) - 3agg(x_2) + 1),\\
\psi_2 &:= \sigma(-x_1 + 4x_2 + 2agg(x_1) + 6agg(x_2) - 2),\\
\chi_1 &:= \sigma(3\psi_1 - agg(\psi_1),\\
\chi_2 &:= \sigma(-2\psi_1 + 5(agg(\psi_2)),\\
\varphi_A &:= 2(\chi_1) - \chi_2 \geq 1.
\end{aligned}$$

Outline

Outline

Modal logic and Graded modal logic

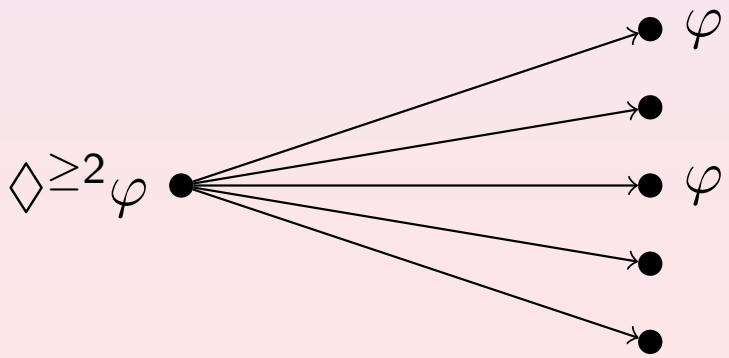
Modal logic

$\Diamond\varphi$: φ holds in at least one successor

Graded Modal logic

$\Diamond^{\geq k}\varphi$: φ holds in at least k successors

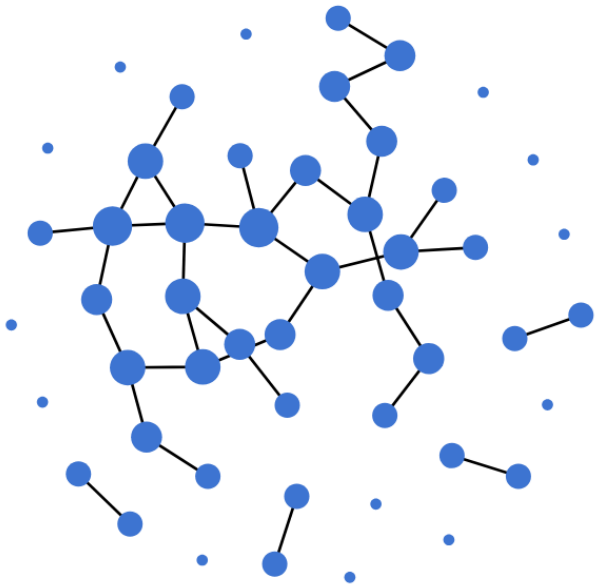
$G, u \models \Diamond^{\geq k}\varphi$ if there are $v_1, \ldots, v_k \in E(u)$ all distinct such that $G, v_i \models \varphi$ for all $i = 1..k$.



Outline

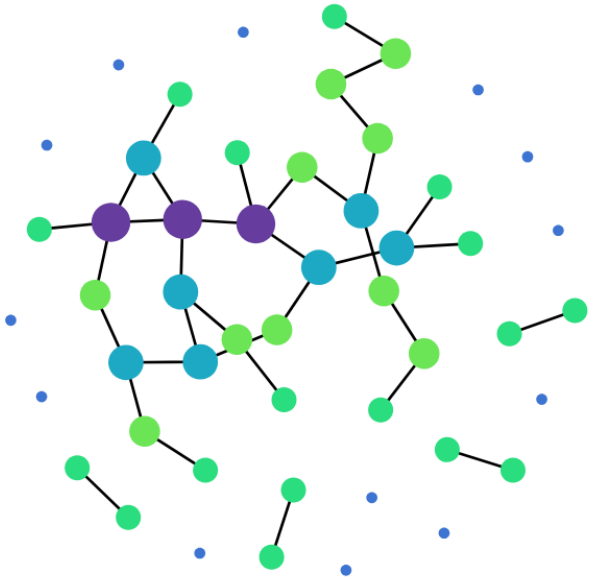
Color Refinement

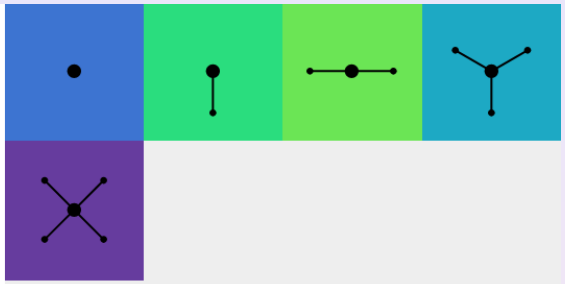
Round ‹ 0 › of 4



Color Refinement

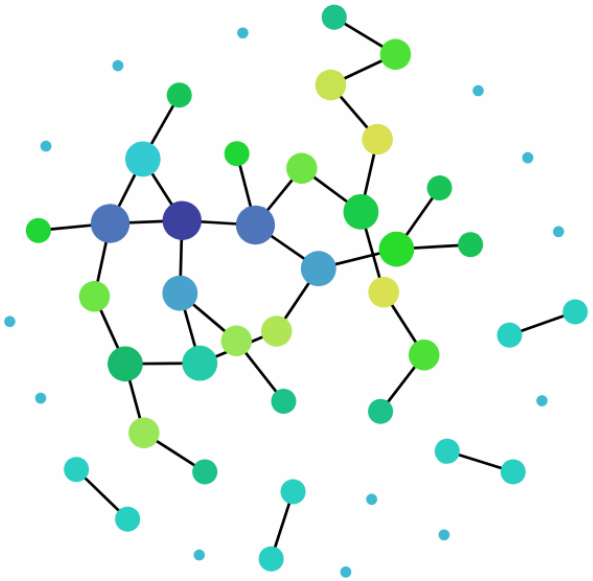
Round ‹ 1 › of 4

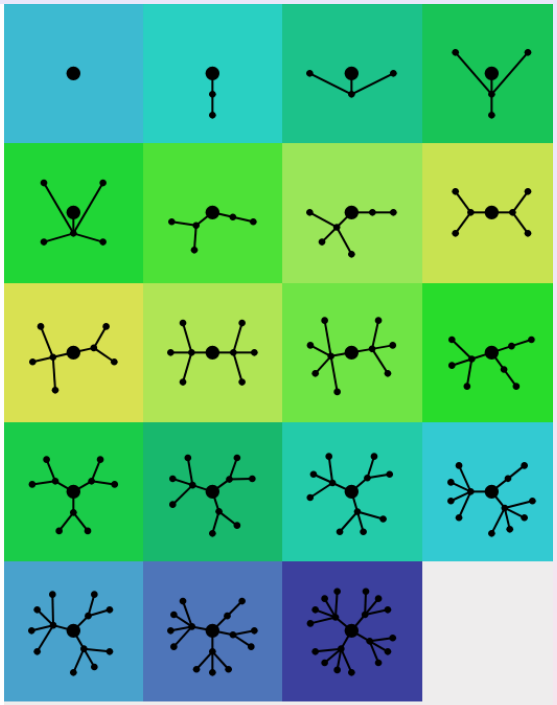




Color Refinement

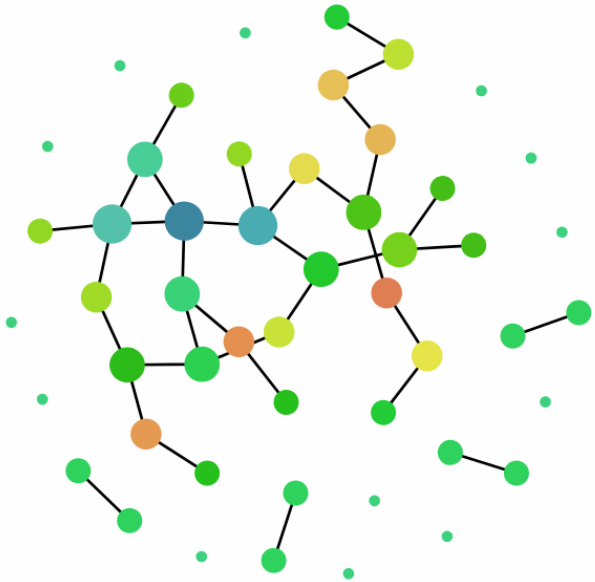
Round ‹ 2 › of 4

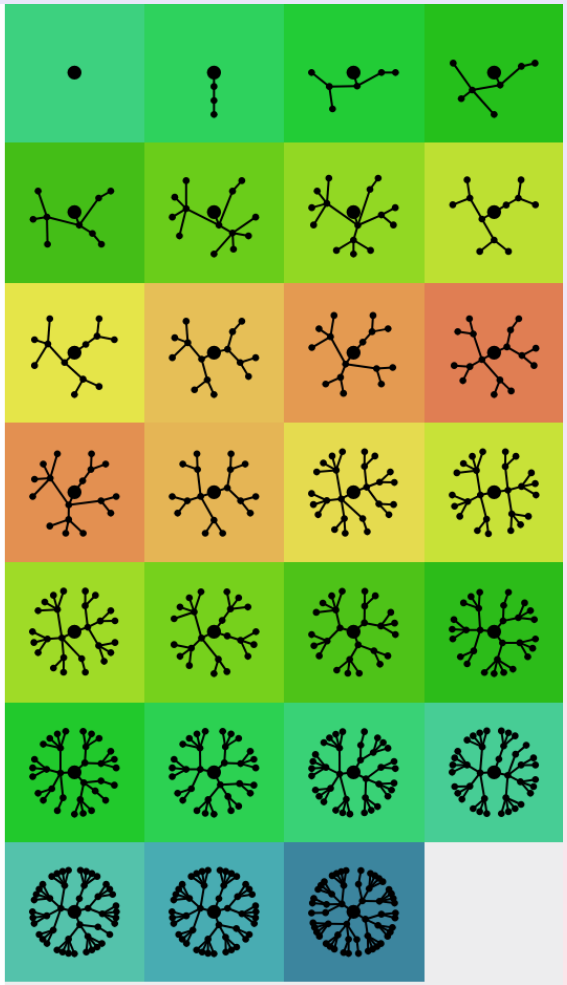




Color Refinement

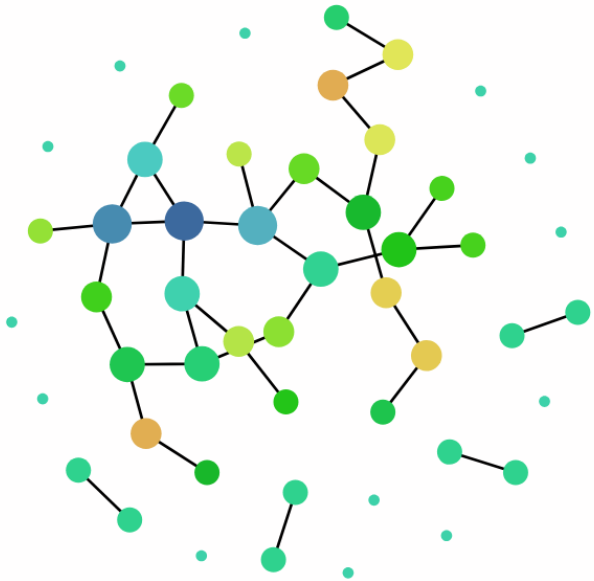
Round ‹ 3 › of 4



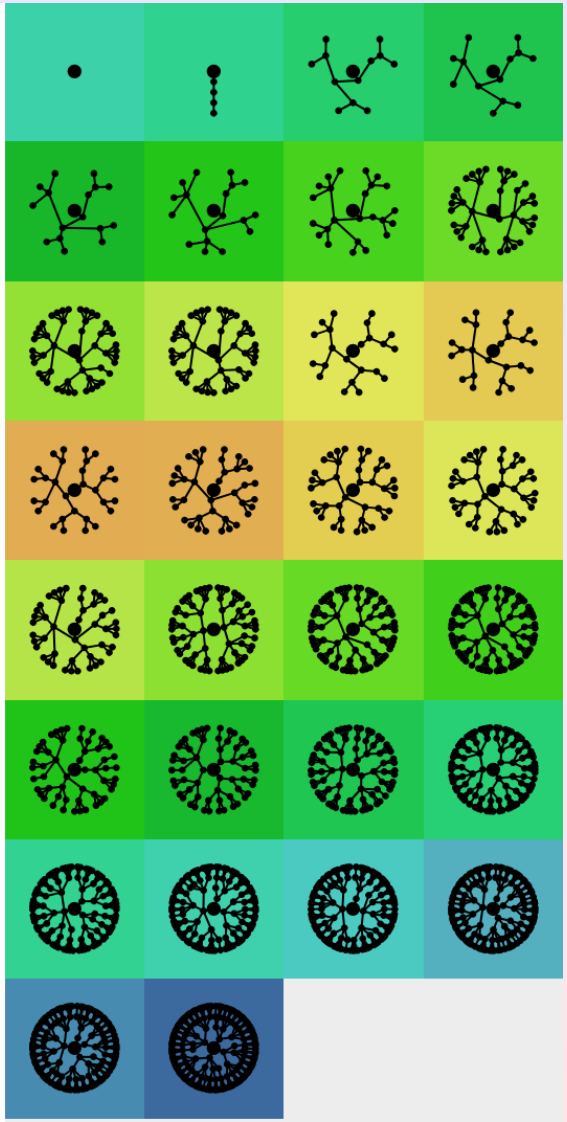


Color Refinement

Round ‹ 4 › of 4



by Holger Dell (2020). Source code on github.



Definition of Colour refinement

Tool: `https://holgerdell.github.io/color-refinement/`

Definition (Morgan, 1965)

- Initially, $\mathrm{color}^{(0)}(G, u) =$ label at u in G;
- $\mathrm{color}^{(t+1)}(G, u) = \Big(\mathrm{color}^{(t)}(G, u), \{\!\{\mathrm{color}^{(t)}(G, v) \mid uEv\}\!\}\Big)$.

When it stabilizes, we get $\mathrm{color}(G, u)$ for all $u \in V$.

$\mathrm{color}(G, u) =$ the information used in the computation of a GNN.

Link between GML, colour refinement and GNNs

Theorem

Let $\mathcal{A}$ be a GNN.
$color(G, u) = color(G, v)$ implies $\mathcal{A}(G, u) = \mathcal{A}(G, v)$.

Theorem (folklore, see Grohe LICS 2021)

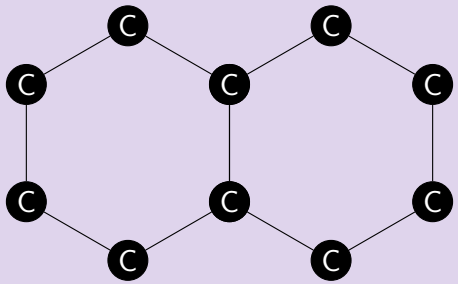
$color(G, u)$=$color(G, v)$ iff both u and v satisfy the same GML-formulas.

Colour refinement: a partial test for isomorphism

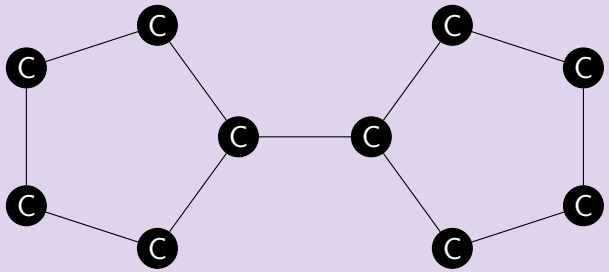
Proposition

$$G \equiv_{iso} G' \implies \{\!\{color(G,u) \mid u \in V\}\!\} = \{\!\{color(G',u') \mid u' \in V'\}\!\}.$$

⚠ The converse $\Longleftarrow$ fails e.g. on regular graphs:



Decalin



Bicyclopentyl

$\rightsquigarrow$ GNNs have been generalized to 2-WL, 3-WL, etc.
[Morris et al., AAAI 2019]

Outline

Expressivity of GNNs

Theorem (Barcélo et al. 2020)

For all GML-formulas φ, there is a GNN $\mathcal{A}$ such that $[\![\mathcal{A}]\!] = [\![\varphi]\!]$.

Theorem (Barcélo et al. 2020)

FO-expressible

For all GNNs $\mathcal{A}$, there is a GML-formula φ such that $[\![\mathcal{A}]\!] = [\![\varphi]\!]$.

In the theorems, the activation function is TruncReLU.

Outline

Outline

Verification problems

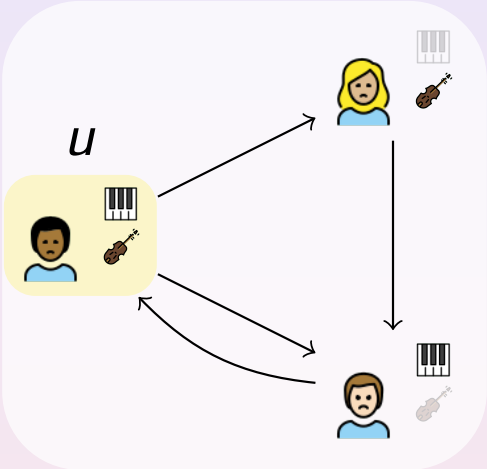
$[[\mathcal{A}]] :=$ set of *pointed graphs* recommended by the GNN $\mathcal{A}$

$[[\varphi]] =$ set of *pointed graphs* satisfying the property φ
e.g. having a violinist friend

$[[\mathcal{A}]] \subseteq [[\varphi]]?$	$[[\varphi]] \subseteq [[\mathcal{A}]]?$	$[[\mathcal{A}]] \cap [[\varphi]] \neq \emptyset?$
Do all recommended persons have a violinist friend?	Are all persons having a violinist friend recommended?	Is it possible to recommend a person having a violinist friend

Outline

Example of a formula of $K^{\#}$



$$pianist \quad \wedge \quad (\#violinist + 2 \times \#pianist \leq 3)$$

 is pianist and $\Big($ the number of  's friends that are violonist $+2\times$ the number of  's friends that are pianist $\leq 3\Big)$

$K^{\#}$ syntax

- $K^{\#}$-formulas φ are Boolean combinations of atomic propositions and inequalities.
- Expressions ξ in inequalities are linear over:
  - 1_φ which 1 if φ holds, 0 otherwise;
  - $\#\varphi$, equals to the number of φ-successors.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \xi \geq 0$$
$$\xi ::= c \mid 1_\varphi \mid \#\varphi \mid \xi + \xi \mid c \times \xi$$

$K^{\#}$ semantics

$$
\begin{array}{lll}
(G,u) \models p & \text{if} & \ell(u)(p) = 1, \\
(G,u) \models \neg\varphi & \text{if} & \text{it is not the case that } (G,u) \models \varphi, \\
(G,u) \models \varphi \wedge \psi & \text{if} & (G,u) \models \varphi \text{ and } (G,u) \models \psi, \\
(G,u) \models \xi \geq 0 & \text{if} & [[\xi]]_{G,u} \geq 0,
\end{array}
$$

$$
\begin{aligned}
[[c]]_{G,u} &= c, \\
[[\xi_1 + \xi_2]]_{G,u} &= [[\xi_1]]_{G,u} + [[\xi_2]]_{G,u}, \\
[[c \times \xi]]_{G,u} &= c \times [[\xi]]_{G,u}, \\
[[1_\varphi]]_{G,u} &= \begin{cases} 1 & \text{if } (G,u) \models \varphi \\ 0 & \text{else,} \end{cases} \\
[[\#\varphi]]_{G,u} &= |\{v \in V \mid (u,v) \in E \text{ and } (G,v) \models \varphi\}|.
\end{aligned}
$$

Outline

From $K^{\#}$ to GNNs

Inspired from [Barcélo et al. 2020], [Nunn et al., 2023, IJCAI 2024], [Benedikt et al. ICALP 2024]

Theorem

For all $K^{\#}$-formulas φ, there is a GNN $tr(\varphi)$ sth. $[\![tr(\varphi)]\!] = [\![\varphi]\!]$.

$$
\begin{aligned}
tr(x_i = 1) &= x_i \text{ provided } x_i \text{ takes its value in } \{0,1\} \\
tr(\neg\varphi) &= 1 - \mathsf{truncReLU}(tr(\varphi)) \\
tr(\varphi \wedge \psi) &= \mathsf{truncReLU}(tr(\varphi) + tr(\psi) - 1) \\
tr(\vartheta \geq 1) &= \mathsf{truncReLU}(\tau(\vartheta)) \\
\tau(\#\varphi) &= agg(tr(\varphi)) \\
\tau(\vartheta + \vartheta') &= \tau(\vartheta) + \tau(\vartheta') \\
\tau(1_\varphi) &= tr(\varphi) \\
\tau(c) &= c
\end{aligned}
$$

From GNNs to $K^{\#}$

Theorem

For all GNNs $\mathcal{A}$, there is a $K^{\#}$-formula $tr'(\mathcal{A})$ sth. $[\![tr'(\mathcal{A})]\!] = [\![\varphi]\!]$.

Proof idea for integer weights:

$$
\begin{aligned}
tr'(x_i) &= x_i \\
tr'(c) &= c \\
tr'(c\vartheta) &= c \times tr'(\vartheta) \\
tr'(\vartheta + \vartheta') &= tr'(\vartheta) + tr'(\vartheta') \\
tr'(\mathsf{truncReLU}(\vartheta)) &= 1_{tr'(\vartheta)\geq 1} \\
tr'(agg(\vartheta)) &= \#(tr'(\vartheta) \geq 1) \text{ provided } \vartheta \text{ is of the form truncReLU(.)}
\end{aligned}
$$

Outline

Satisfiability of $K^{\#}$

Theorem

Satisfiability of $K^{\#}$ is decidable and is in PSPACE.

we have an algorithm using a polynomial amount of space.
$\sim$ "implementable"

Alternative proofs:

- [Nunn et al. 2024]
  By poly-time reduction to a logic in [Demri and Lugiez 2010]
- [Lecture notes]
  Tableau method + oracle to QFBAPA-sat
  (close to [Baader, 2017])

Quantifier-free Boolean algebra and Presburger arithmetics

[Kuncak et al. 2007]

Example (of a QFBAPA formula)

$$|\mathit{pianist} \cap \mathit{student}| + x \geq 5 \quad \wedge \quad (|\mathit{pianist}| \leq 10 \vee |\mathit{student}| \leq 10)$$

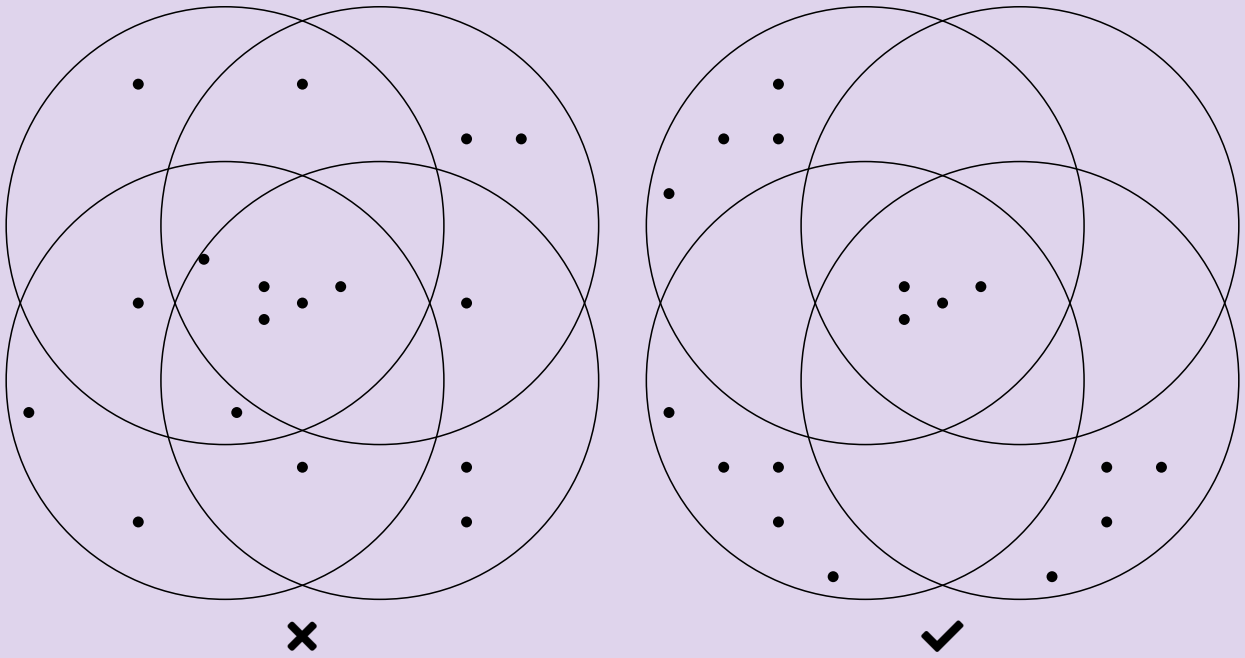
- Integer variables: x, etc.
- Set variables: *pianist*, *student*, etc.
- Cardinality of some sets expressed with Boolean algebras
- Linear inequalities involving integer variables and cardinality

Quantifier-free Boolean algebra and Presburger arithmetics

Theorem (Kuncak et al. 2007)

QFBAPA-satisfiability is in NP.

Proof.



□

Outline

Outline

Global readout

function $layer_i(V, E, \ell)$
  $\ell' :=$ new labelling $V \to \mathbb{Q}^d$
  for vertices $u \in V$ **do**
    $aggregation := \sum \{\!\{ \ell[v] \mid v \in E(u) \}\!\}$
    $globalreadoutaggregation := \sum \{\!\{ \ell[v] \mid v \in V \}\!\}$

$$\ell'[u] := \vec{\sigma} \left(\begin{array}{l} A_i \times \ell[u] \\ +B_i \times aggregation \\ +C_i \times globalreadoutaggregation \\ +b_i \end{array} \right)$$

  return ℓ'

Global readout

$K^{\#,\#^g}$ extends $K^{\#}$ with global counting modality $\#^g\varphi$:

$$[\![\#^g\varphi]\!]_{G,u} = \text{number of } \varphi\text{-vertices in } G.$$

Proposition

GlobalreadoutTruncReLUGNNs and $K^{\#,\#^g}$ are equivalent.

Theorem

$K^{\#,\#^g}$-satisfiability is:

- *NEXPTIME-complete for directed graphs; [Chernobrovkin et al. 2025]*
- *undecidable if undirected graphs. [Benedikt et al. 2024]*

Outline

ReLU GNNs

Theorem (Benedikt et al. 2024)

The satisfiability of ReLU-GNNs is:

- *NEXPTIME-complete for directed graphs;*
- *undecidable if undirected graphs or global readout.*

Outline

Idealistic GNN vs Quantized GNN

Idealistic GNN

- Arbitrary big integers
  35467487612987698761230
- Arbitrary large rationals
  4238761289293/123876298375
- Real numbers

$$\sqrt{2} \qquad \pi \qquad e$$

Quantized GNN

0	0	1	0	1	1	0	1

- 32-bit floating-point arithmetics
- 16-bit fixed-point arithmetics
- 8-bit signed integers

[Gholami et al. 2021] [Zhu et al. ICLR 2023]

Verification of quantized GNNs

Theorem (Sälzer et al. IJCAI 2025)

The verification problems are in PSPACE.

- ☺ in PSPACE for *many* activation and aggregation functions
- ☺ We can check GML-formulas...
- ☹ ...for all modalities $\Diamond^{\geq k}\varphi$, number k should representable with n bits

Tableau method

$$(\vee)\ \frac{(w\ \varphi \vee \psi)}{(w\ \varphi) \mid (w\ \psi)} \qquad (\neg\wedge)\ \frac{(w\ \neg(\varphi \wedge \psi))}{(w\ \neg\varphi) \mid (w\ \neg\psi)}$$

$$(\neg\vee)\ \frac{(w\ \neg(\varphi \vee \psi))}{(w\ \neg\varphi), (w\ \neg\psi)} \qquad (\neg\neg)\ \frac{(w\ \neg\neg\varphi)}{(w\ \varphi)}$$

$$(\wedge)\ \frac{(w\ \varphi \wedge \psi)}{(w\ \varphi), (w\ \psi)}$$

$$(\neg\geq)\ \frac{(w\ \neg(\vartheta \geq k))}{(w\ \vartheta = k') \text{ for some } k' \in \mathbb{K}_n \text{ with } k' <_{\mathbb{K}_n} k}$$

$$(+)\ \frac{(w\ \vartheta_1 + \vartheta_2 = k)}{(w\ \vartheta_1 = k_1), (w\ \vartheta_2 = k_2) \text{ for some } k_1, k_2 \in \mathbb{K}_n, \text{ with } k_1 +_{\mathbb{K}_n} k_2 = k}$$

$$(degree)\ \frac{(w\ \varphi) \text{ and no term } (w\ degree = ...)}{(w\ degree = \delta') \text{ for some } \delta' \leq \delta}$$

$$(clash_c)\ \frac{(w\ c = k) \text{ if } c \neq k}{\text{fail}}$$

$$(clash_=)\ \frac{(w\ x_i = k), (w\ x_i = k') \text{ if } k \neq k'}{\text{fail}}$$

$$(\sigma)\ \frac{(w\ \sigma(\vartheta) = k)}{(w\ \vartheta = k') \text{ for some } k' \in \mathbb{K}_n \text{ with } [\![\sigma]\!](k') = k}$$

$$(\geq)\ \frac{(w\ \vartheta \geq k)}{(w\ \vartheta = k') \text{ for some } k' \in \mathbb{K}_n \text{ with } k' \geq_{\mathbb{K}_n} k}$$

$$(\times)\ \frac{(w\ c\vartheta = k)}{(w\ \vartheta = k') \text{ for some } k' \in \mathbb{K}_n \text{ with } c \times_{\mathbb{K}_n} k' = k}$$

$$(agg)\ \frac{(w\ agg(\vartheta) = k) \qquad (w\ degree = \delta')}{(w1\ \vartheta = k_1), \ldots, (w\delta'\ \vartheta = k_{\delta'}) \text{ for some } (k_u)_{u=1..\delta'}, \text{ with } k_1 +_{\mathbb{K}_n} \cdots +_{\mathbb{K}_n} k_{\delta'} = k}$$

where $\mathbb{K}_n$ is the set of quantized numbers over n bits

Implementations for quantized GNNs

- Tableau method:

  `https://github.com/francoisschwarzentruber/ijcai2025-verifquantgnn`

- Bounded verification, also with global readout:

  `https://github.com/francoisschwarzentruber/gnn_verification`

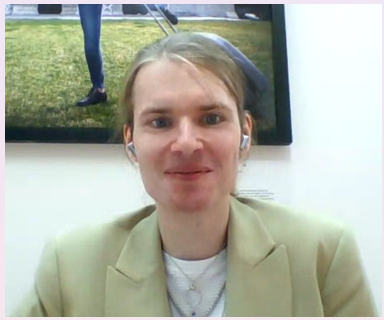
Outline

Perspectives

- Other ML models: GNN with attention, etc.
- Efficient implementation
- Applications
- Define new verification tasks
- Design interpretable GNNs

Thanks to:

- the existence of M1 research project at ENS Rennes
- Stéphane Demri
- Artem Chernobrovkin, Pierre Nunn, Marco Sälzer, Nicolas Troquard




- master (M2) students at ENS de Lyon

and

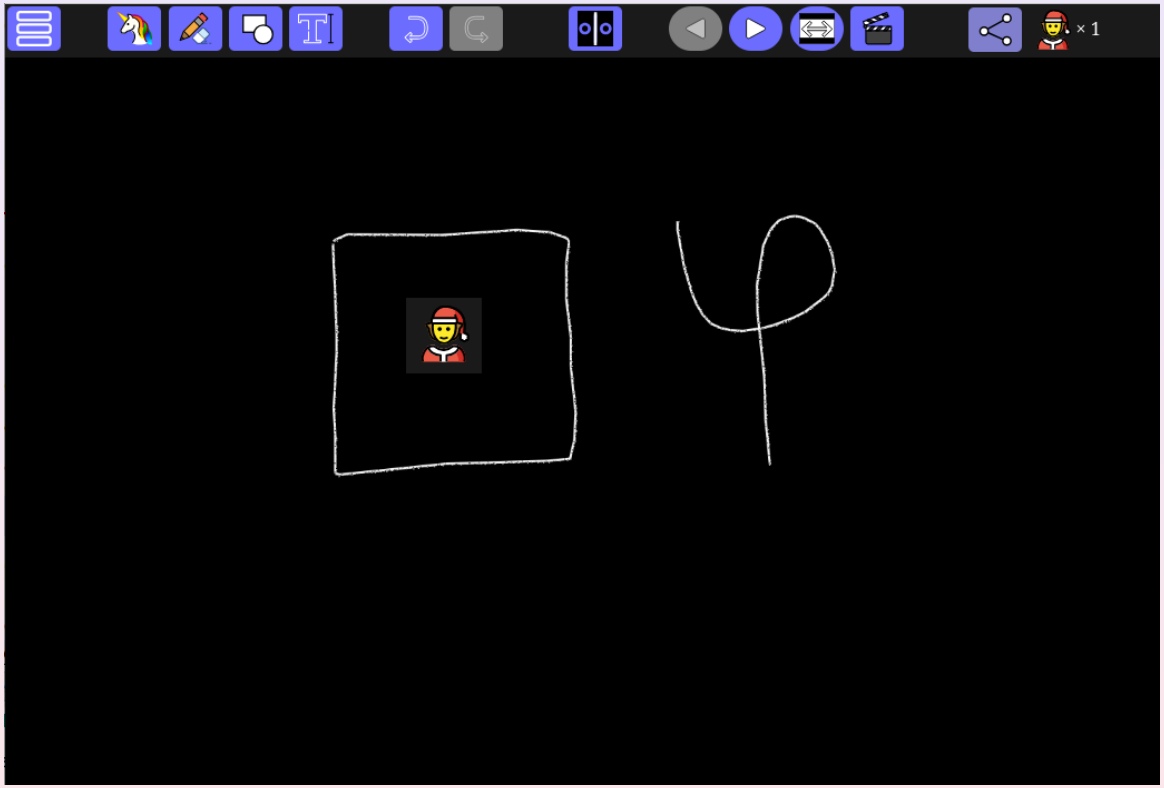
Thank you! 谢谢大家

LORI-II 2009 - Chongqing - October 8-11 2009

Lecture notes

https://arxiv.org/abs/2510.11617

Advertisement for Tableaunoir (open-source online blackboard solution)



https://tableaunoir.github.io/