The Role of Logic in Advancing Machine Learning: Three Case Studies

Pablo Barceló
Institute for Mathematical and Computational Engineering
Universidad Católica de Chile &
IMFD Chile & CENIA Chile

Machine learning requires
solid foundations to be fully deployable

Both the theory of computation and logic can be
instrumental in this endeavor

Some examples of the application of logical methods in machine learning (ML):

- Representing knowledge (Neuro-symbolic AI)
- Understanding the expressive power of ML models
- Verifying the correctness of such models
- Improving the interpretability of ML decisions
- Designing declarative, general purpose languages for ML tasks

Some examples of the application of logical methods in machine learning:

- Representing knowledge (Neuro-symbolic AI)
- Understanding the expressive power of ML models
- Verifying the correctness of such models
- Improving the interpretability of ML decisions
- Designing declarative, general purpose languages for ML tasks

THE EXPRESSIVE POWER OF GRAPH NEURAL NETWORKS

Motivation

Graph Neural Networks (GNNs) have become ubiquitous in applications where data is structured in the form of graphs:

- Chemical and biological networks
- Social networks
- Recommendation systems
- Semantic parsing and question answering

Question: Which graph properties can GNNs express?

Graphs in This Presentation

Node-labeled, undirected graphs

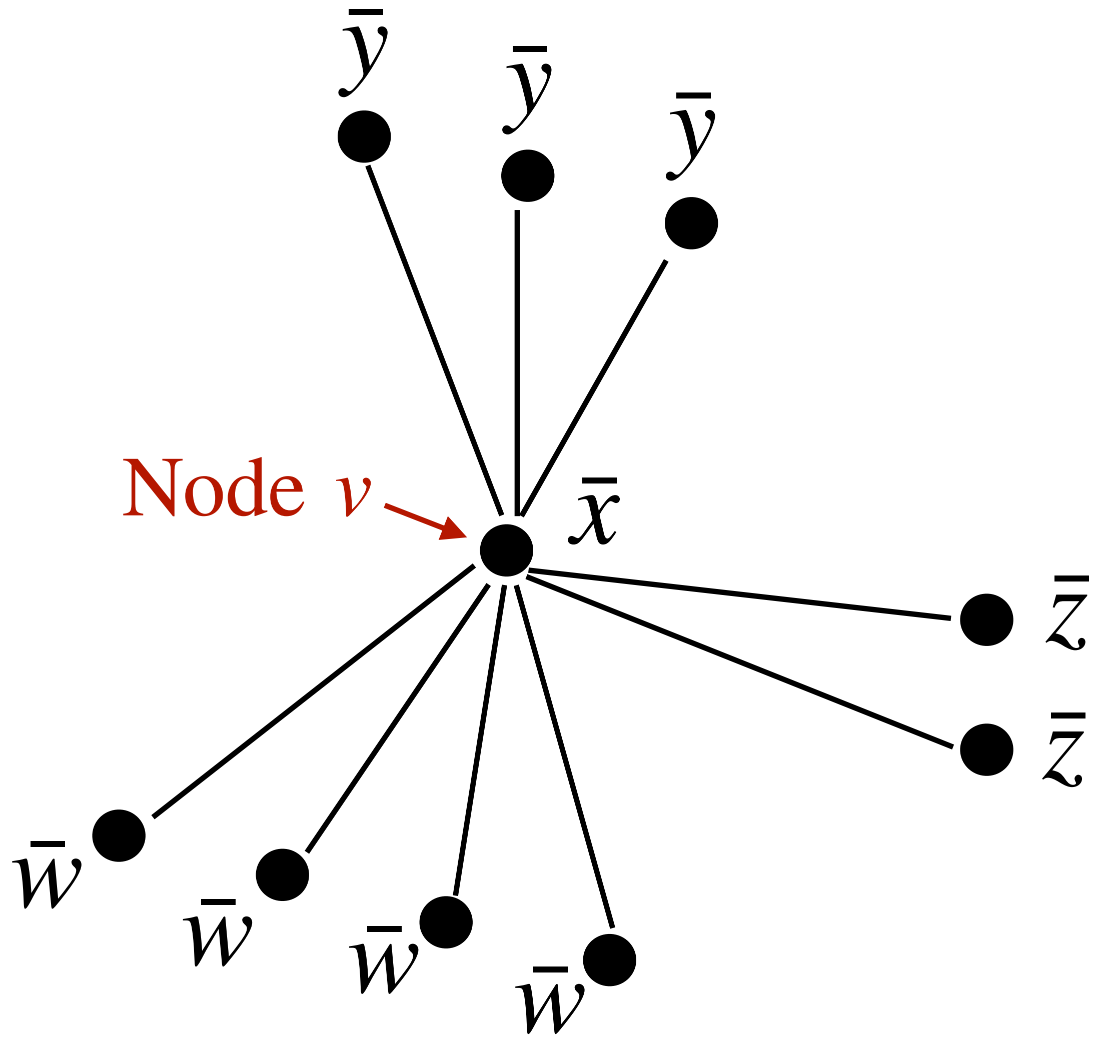
$$G = (V, E, \gamma)$$

$V =$ finite set of nodes

$E =$ set of undirected edges over V

$\gamma =$ function that maps each node v to some label $\gamma(v)$

Message-Passing GNNs (MPGNNs)



$\bar{x}, \bar{y}, \bar{z}, \bar{w} \in \mathbb{R}^d$

$$\text{New\_embedding}(v) = \text{Comb}(\bar{x}, \text{Agg}(\{\{\bar{y}, \bar{y}, \bar{y}, \bar{z}, \bar{z}, \bar{w}, \bar{w}, \bar{w}, \bar{w}\}\}))$$

(Embeddings are initialized as one-hot encodings of node labels)

A Particular MPGNN Architecture

$\text{MPGNN}^t(v)$ = embedding vector computed for node v after t steps

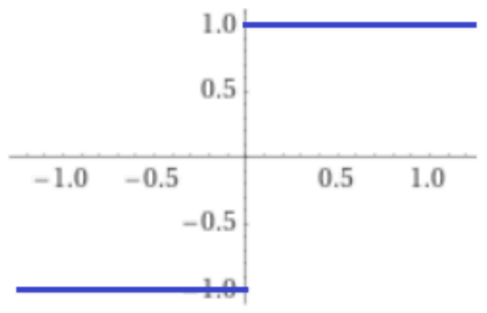
$\text{MPGNN}^0(v)$ = one-hot encoding of $\gamma(v)$

$$\text{MPGNN}^t(v) \;=\; \sigma\left(\text{MPGNN}^{t-1}(v)\cdot W_1^t + \sum_{(u,v)\in E} \text{MPGNN}^{t-1}(u)\cdot W_2^t + b^t\right)$$

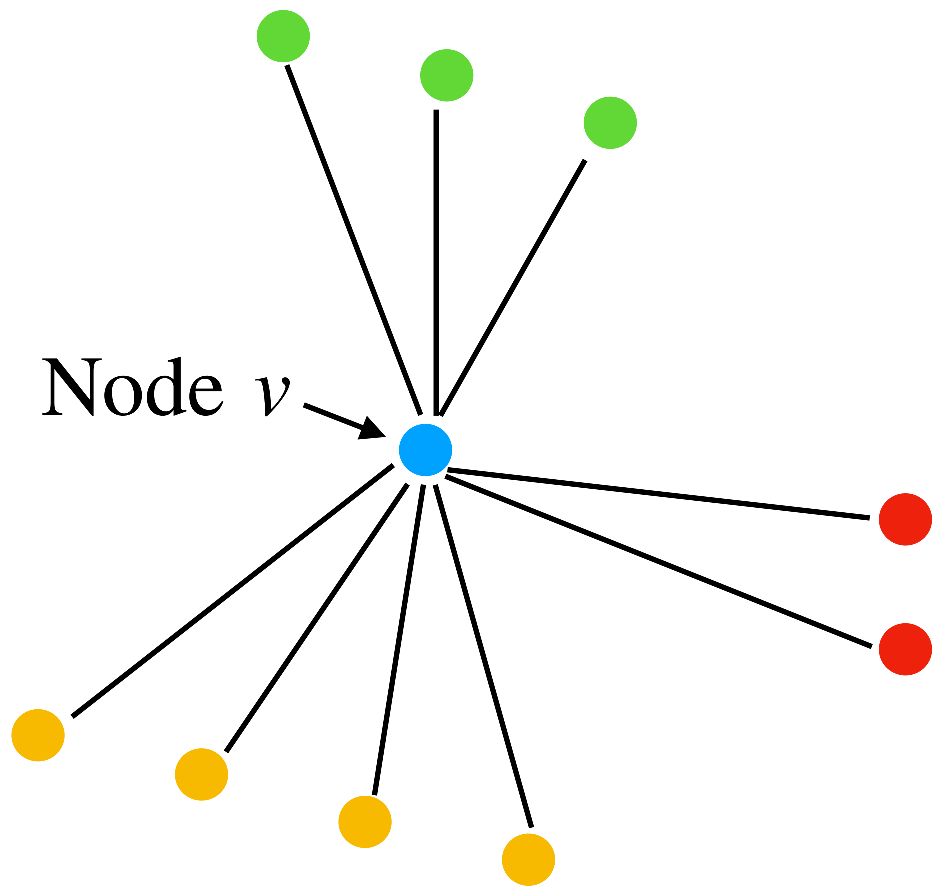
W_1^t, W_2^t = matrices of parameters, b^t = bias vector

σ is a non-linear function, e.g., *sign*, *ReLU*, *truncated ReLU*, *sigmoid*, *tanh*

Positive results based on *sign* function:



Weisfeiler-Leman Test (WL Test)



Newcolor(v) = Hash(🔵 , {{ 🟢 , 🟢 , 🟢 , 🔴 , 🔴 , 🟡 , 🟡 , 🟡 , 🟡 }})

(Initial colors are an injective representation of node labels)

Expressive Power of MPGNNs

$\text{WL}^t(v)$ = color assigned to node v by WL after t steps

Theorem (Morris et al. 2019 & Xu et al. 2019)

For each nodes u, v in a graph G,

$$\text{WL}^t(u) = \text{WL}^t(v) \implies \text{MPGNN}^t(u) = \text{MPGNN}^t(v)$$

Moreover, there exists a sequence $(W_1^1, W_2^1), \ldots, (W_1^t, W_2^t)$ of parameter matrices for which

$$\text{WL}^t(u) = \text{WL}^t(v) \iff \text{MPGNN}^t(u) = \text{MPGNN}^t(v)$$

Logic over Colored Graphs

FO over the vocabulary that contains:

unary symbols $\mathrm{Red}(x), \mathrm{Blue}(x), \ldots$
for the node colors

a binary symbol $E(x, y)$ that represents
the adjacency relationship

Example

$$\exists x \, \exists y \, \exists x \big(E(x, y) \wedge E(y, z) \wedge E(z, x) \wedge \mathrm{Red}(x) \big)$$

Two-variable Fragment

C_2 = Two-variable fragment of FO extended with *counting quantifiers*:

$\exists^{\geq k} x\, \phi(x)$ = number of a's that satisfy ϕ is at least k

The following is a C_2 formula:

$$\exists^{\geq 3} y \Big(E(x, y) \wedge \text{Red}(y) \wedge \neg \exists x (E(y, x) \wedge \text{Blue}(x)) \Big)$$

But the following is not:

$$\exists x\, \exists y\, \exists x \Big(E(x, y) \wedge E(y, z) \wedge E(z, x) \wedge \text{Red}(x) \Big)$$

Two-variable Fragment

C_2^t = Formulas from C_2 with quantifier depth at most t

For a graph G and nodes u, v:

$G, u \equiv_{C_2^t} G, v$ expresses that for every $\phi(x) \in C_2^t$: $G \vDash \phi(u) \iff G \vDash \phi(v)$

A Logical Characterization of WL

Theorem (Cai, Fürer, and Immerman 1992)

For each nodes u, v in a graph G:

$$\mathsf{WL}^t(u) = \mathsf{WL}^t(v) \iff G, u \equiv_{C_2^t} G, v$$

In Terms of Logic

Corollary

For nodes u, v in a graph G:

$$G, u \equiv_{C_2^t} G, v \implies \text{MPGNN}^t(u) = \text{MPGNN}^t(v)$$

Moreover, there exists a sequence $(W_1^1, W_2^1), \ldots, (W_1^t, W_2^t)$ of parameter matrices for which

$$G, u \equiv_{C_2^t} G, v \iff \text{MPGNN}^t(u) = \text{MPGNN}^t(v)$$

A Caveat

Previous characterizations are non-uniform:

The matrices $(W_1^1, W_2^1), \ldots, (W_1^t, W_2^t)$
depend on the size of the graph

The formula from C_2^t that distinguishes u and v
is constructed from the underlying graph

Question

How can we obtain <u>uniform</u> characterizations of the expressive power of GNNs?

Definition

Let M be an MPGNN with t layers

We write $M(u)$ for the embedding obtained by M on u after t iterations

M *expresses* an FO formula $\phi(x)$, if for every graph G and node u:

$$G \models \phi(u) \quad \Longleftrightarrow \quad M(u)_1 = 1$$

the first component of $M(u)$

A Lower Bound

Theorem (B., Kostylev, Monet, Pérez, Reutter, Silva 2019)

Let $\phi(x)$ be a *guarded* formula in C_2. There is an MPGNN that expresses $\phi(x)$

A formula is *guarded* if all its quantified sub-formulas
are of the form:

$$\exists y (E(z,y) \wedge \phi(y)) \qquad \forall y (E(z,y) \rightarrow \phi(y))$$

Guarded Formulas

The following C_2 formula is guarded:

$$\exists^{\geq 3} y \big(E(x,y) \wedge \mathrm{Red}(y) \wedge \neg \exists x (E(y,x) \wedge \mathrm{Blue}(x)) \big)$$

But the following is not:

$$\exists^{\geq 3} y \, \mathrm{Red}(y)$$

Proof Idea

Rewrite $\phi(x)$ as an equivalent formula in *graded modal logic*

This is the extension of basic modal logic with expressions:

$\Diamond^{\geq k}\phi$ = nodes with at least k neighbors satisfying ϕ

The embedding computed by the MPGNN on node u has one component for each sub-formula

At each layer t, we consider sub-formulas of $\Diamond$-depth bounded by t

This layer assigns a 1 to such component iff the formula is satisfied in node u

An Upper Bound

Proposition (B., Kostylev, Monet, Pérez, Reutter, Silva 2019)

Let $\phi(x)$ be an FO formula that can be expressed as an MPGNN. Then $\phi(x)$ is equivalent to a guarded C_2 formula

Formulas expressed by MPGNNs are closed under *counting bisimulations*

Otto (2019) has shown that such formulas are equivalent to formulas in graded modal logic, and thus in C_2

Capturing C_2

Only guarded C_2 formulas can be expressed by MPGNNs

Is there a meaningful extension of MPGNNs that
can express all C_2 formulas?

Global Readouts

We extend MPGNNs with *global readouts*:

$$\mathrm{MPGNN}^0(v) = \text{one-hot encoding of } \gamma(v)$$

$$\mathsf{MPGNN}^t(v) \;=\; \sigma\Big(\mathsf{MPGNN}^{t-1}(v)\cdot W_1^t + \sum_{(u,v)\in E} \mathsf{MPGNN}^{t-1}(u)\cdot W_2^t + \sum_{u\in V} \mathrm{MPGNN}^{t-1}(u)\cdot W_3^t + b^t\Big)$$

W_1^t, W_2^t, W_3^t = matrices of parameters, b^t = bias vector

A Lower Bound

Theorem (B., Kostylev, Monet, Pérez, Reutter, Silva 2019)

Let $\phi(x)$ be a formula in C_2. There is an MPGNN with global readouts that expresses $\phi(x)$

A single global readout in the last layer suffices

Proof Idea

Rewrite $\phi(x)$ as an equivalent formula in
graded modal logic with global modalities
(Lutz, Sattler, Wolter, 2001)

Use an inductive procedure on sub-formulas,
similar to the one applied for graded modal logic

THE EXPRESSIVE POWER OF TRANSFORMER ENCODERS

Transformers

Transformers are a deep learning architecture that acts as a sequence-to-sequence transducer

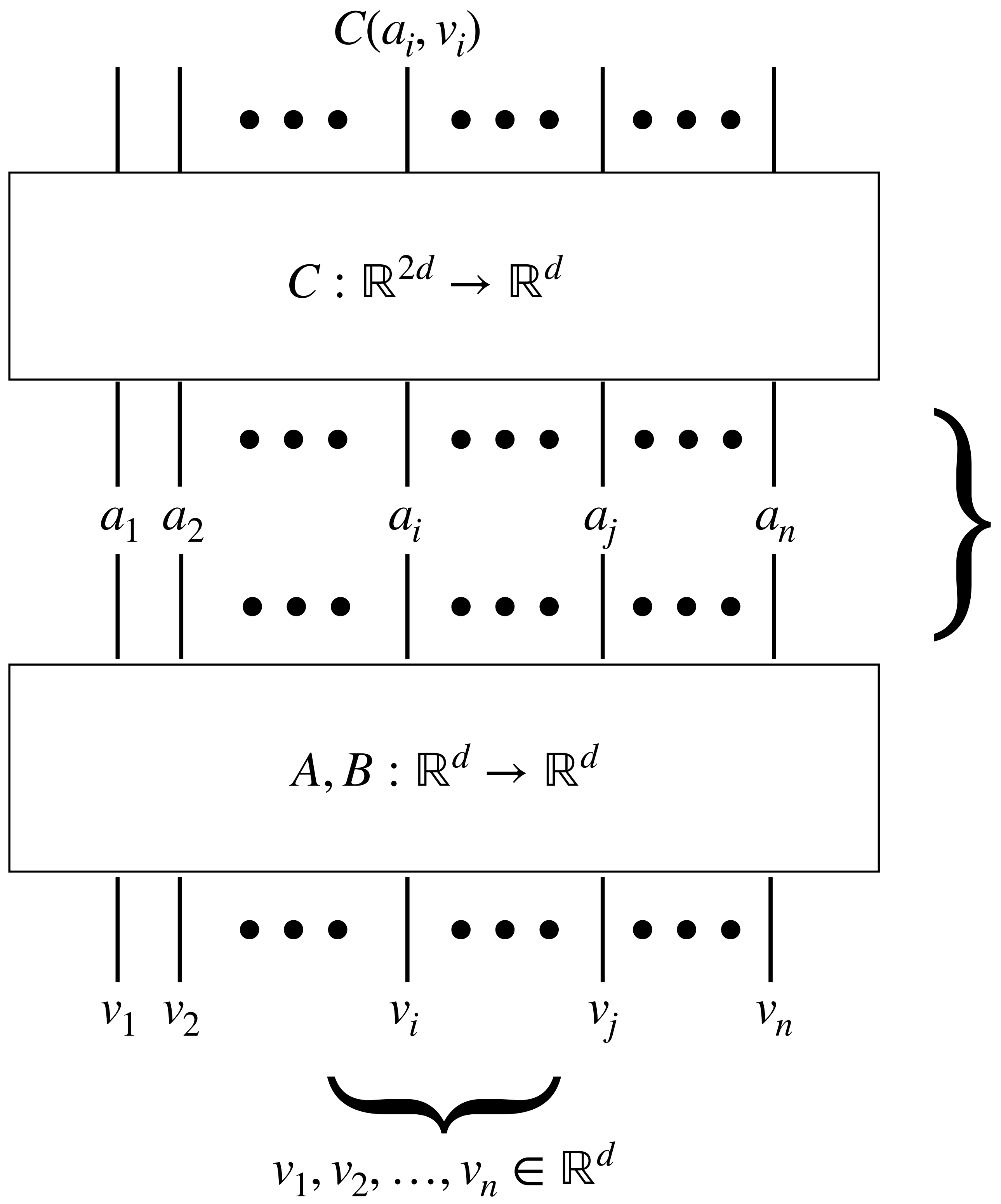
They lie at the core of many of the most popular LLMs:
GPT-2, GPT-3, GPT-4, AlbertAGPT, Claude, BERT, ChatGPT

Motivation

What are the computational limits of Transformers?

Which languages are accepted by Transformers?

Standard Encoder Layer


$C(a_i, v_i)$
$C : \mathbb{R}^{2d} \to \mathbb{R}^d$
a_1 a_2 a_i a_j a_n
$A, B : \mathbb{R}^d \to \mathbb{R}^d$
v_1 v_2 v_i v_j v_n
$v_1, v_2, \ldots, v_n \in \mathbb{R}^d$

$$a_i = v_k$$

for $k = \arg \max_{\ell \in \{1,\ldots,n\}} A\bar{v}_i \cdot B\bar{v}_\ell$

(breaking ties by choosing the leftmost element)

Attention Mechanism

$$a_i = v_k$$

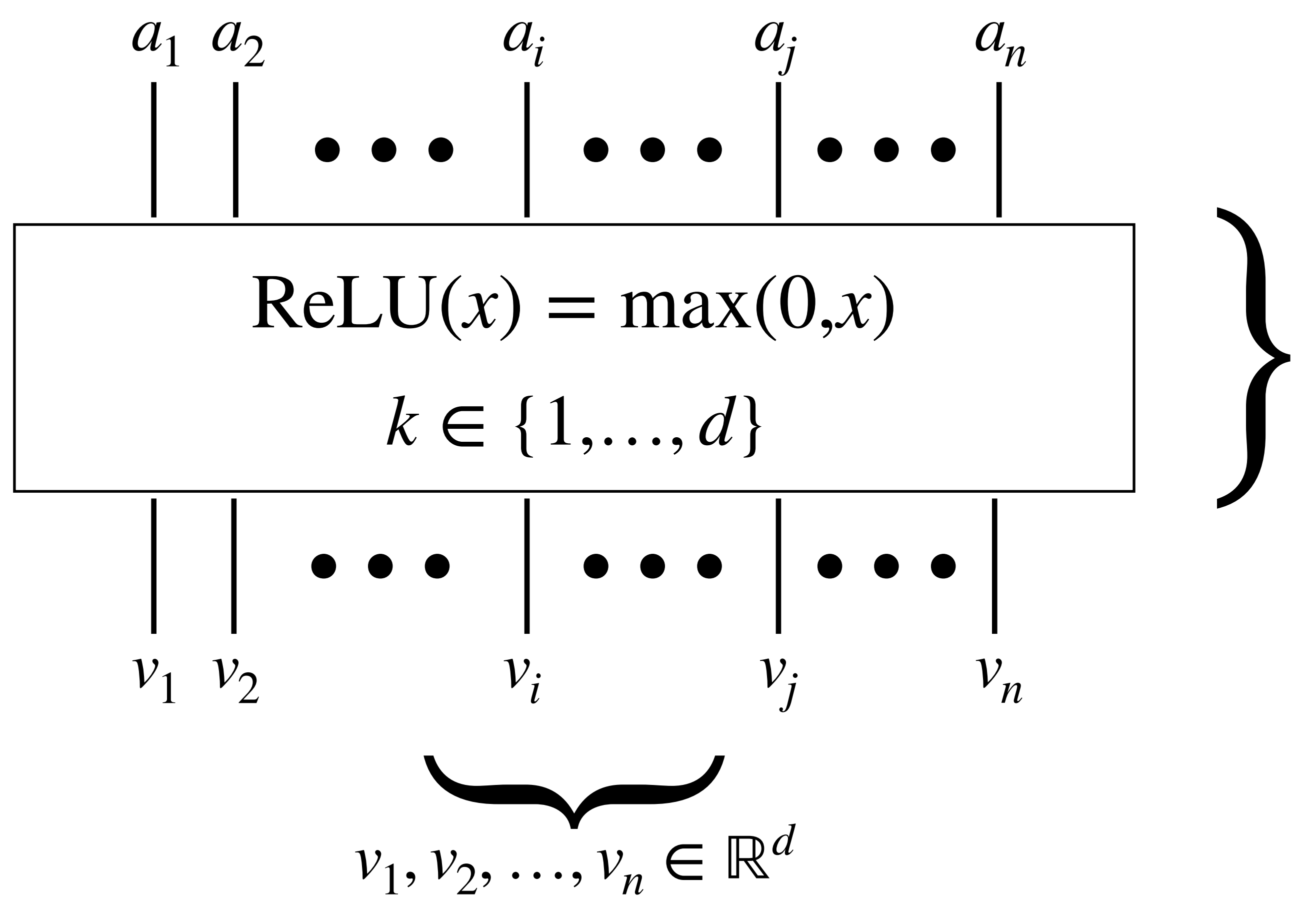
for $k = \arg\max_{\ell \in \{1,\ldots,n\}} A\bar{v}_i \cdot B\bar{v}_\ell$

(breaking ties by choosing the leftmost element)

This is known as *unique hard attention*

It is *hard*: only elements that maximize $Av_i \cdot Bv_\ell$ are considered
It is *unique*: ties are broken by choosing a single element

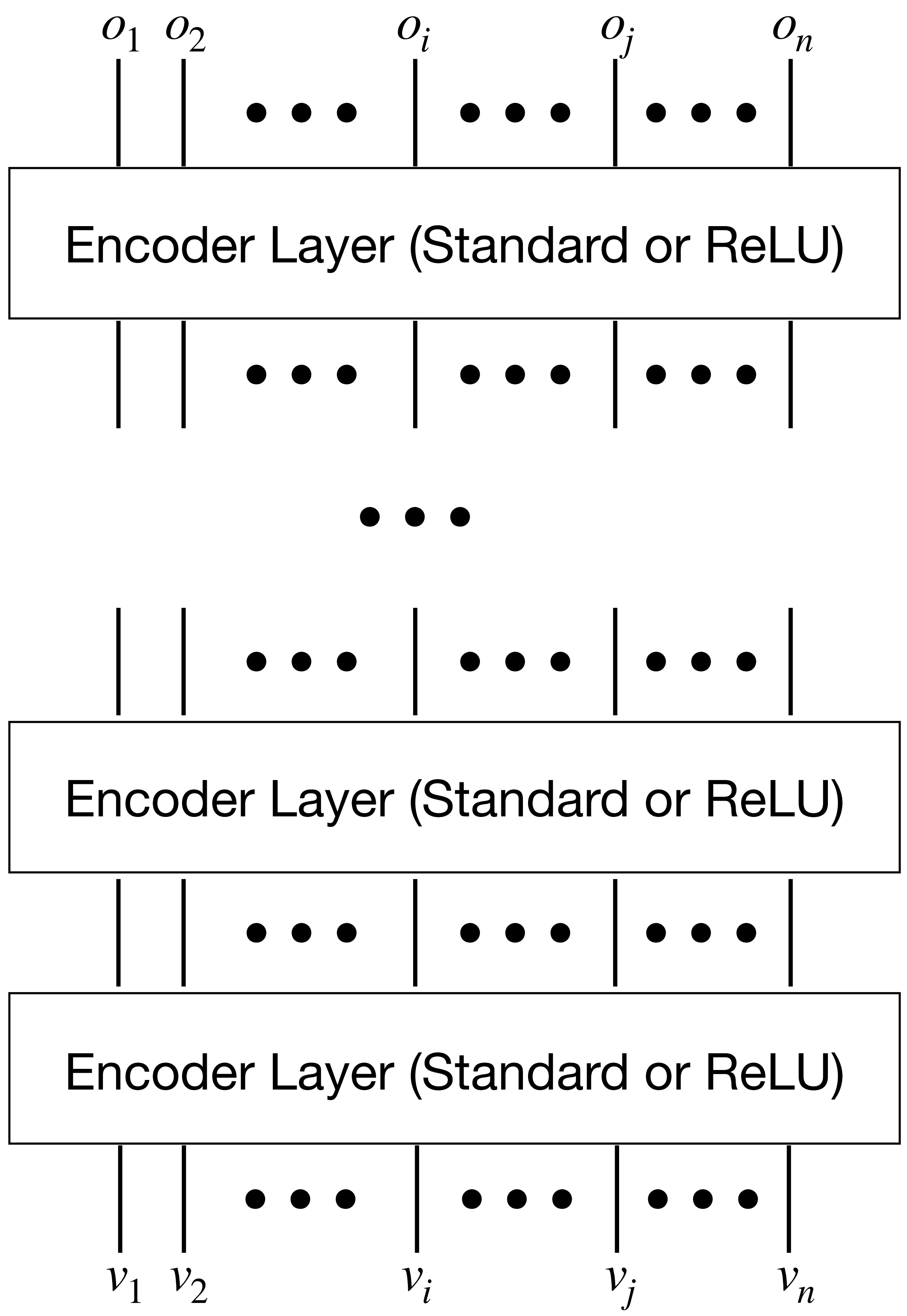
ReLU Encoder Layer



a_1 a_2 a_i a_j a_n
$\mathrm{ReLU}(x) = \max(0,x)$
$k \in \{1,\ldots,d\}$
v_1 v_2 v_i v_j v_n
$\underbrace{\qquad}_{v_1, v_2, \ldots, v_n \in \mathbb{R}^d}$

a_i is obtained from v_i by applying ReLU on its k-th component

Transformer Encoder



o_1 o_2 o_i o_j o_n
Encoder Layer (Standard or ReLU)
Encoder Layer (Standard or ReLU)
Encoder Layer (Standard or ReLU)
v_1 v_2 v_i v_j v_n

Encodings

Consider a finite alphabet Σ

An *encoding of* Σ is a function $f : \Sigma \to \mathbb{R}^d$

A *positional encoding* is a function $p : \mathbb{N} \to \mathbb{R}^e$

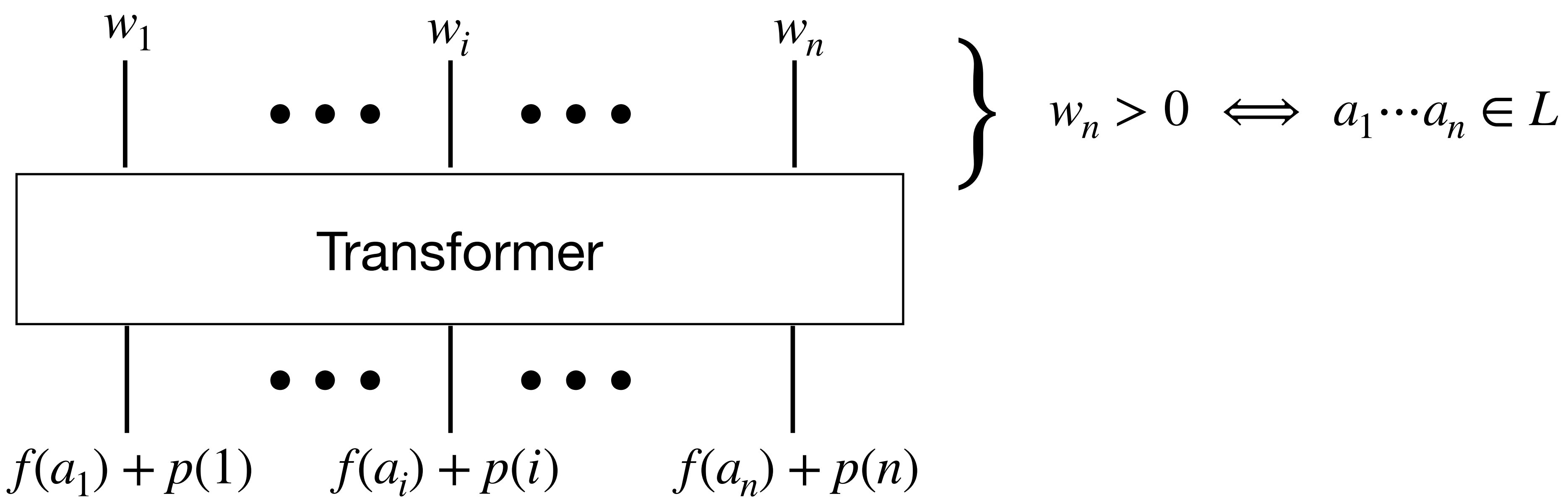
Language Recognizers

Consider a language $L \subseteq \Sigma^+$ and
T a Transformer encoder with unique hard attention

L is accepted by T if there exist:

(1) An alphabet encoding $f : \Sigma \to \mathbb{R}^d$
(2) A positional encoding $p : \mathbb{N} \to \mathbb{R}^e$

such that for every $a_1 \cdots a_n \in \Sigma^+$:



$w_n > 0 \iff a_1 \cdots a_n \in L$

An Upper Bound

Theorem (Hao, Angluin, Frank, 2022)

Every language that can be accepted by a Transformer encoder with unique hard attention is in the class AC^0

This is Not Optimal

Proposition (B., Kozachiskiy, Lin, Podolskii, 2024)

There is an AC^0 language that is not accepted by any Transformer encoder with unique hard attention

A Significant Lower Bound

Recall: $\mathrm{AC}_0 = \mathrm{FO}$ $+$ arbitrary numerical predicates

Theorem (B., Kozachiskiy, Lin, Podolskii, 2024)

Let L be an AC^0 language that is definable in FO $+$ arbitrary *monadic* numerical predicates.

Then L is accepted by a Transformer encoder with unique hard attention.

Proof Idea

Apply Kamp's Theorem and
rewrite the FO formula that defines L as a
formula in *linear temporal logic* (LTL)

Show that every language definable by an LTL formula is
accepted by a Transformer encoder with unique hard attention

An Application

Every regular language in AC^0 is definable in
FO with monadic numerical predicates
(Barrington, Compton, Straubing, Thérien, 1992)

Corollary (B., Kozachiskiy, Lin, Podolskii, 2024)

Let L be a regular language in AC^0. Then L is accepted by a Transformer encoder with unique hard attention.

A Different Attention Mechanism

$$\boxed{\begin{array}{c} S_i := \text{set of } \ell's \text{ that maximize } Av_i \cdot Bv_\ell \\ \\ a_i = \sum_{\ell \in S_i} v_\ell \,/\, |S_i| \end{array}}$$

This is known as *average hard attention*

A Lower Bound

Transformers with average hard attention can recognize languages in TC^0

Theorem (B., Kozachiskiy, Lin, Podolskii, 2024)

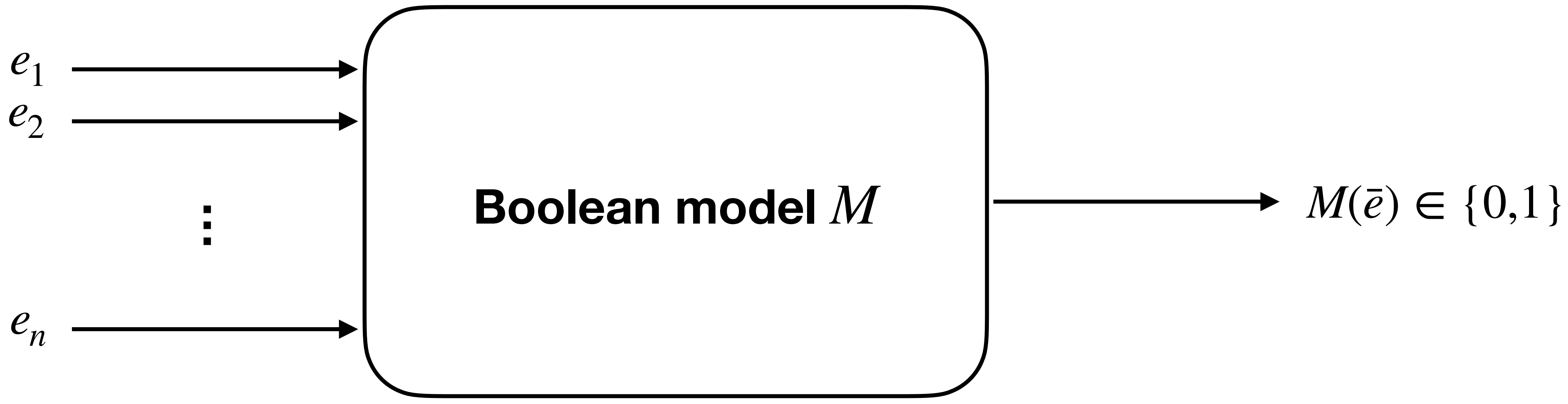
Every permutation-closed language over alphabet $\{0,1\}$ is accepted by a Transformer encoder with average hard attention

As a corollary, we obtain that both *majority* and *parity* are accepted by Transformer encoders with average hard attention

DECLARATIVE LANGUAGES FOR EXPLAINING DECISION TREES

Explainability Problem

Boolean input $\bar{e} = (e_1, e_2, \ldots, e_n) \in \{0,1\}^n$



Question: How can we explain the output of M on $\bar{e}$?

Notation

We assume the set of features of the Boolean model M to be:

$$X = \{x_1, \ldots, x_n\}$$

An *input* is a function $\bar{e} : X \to \{0,1\}$

We write $M(\bar{e})$ for the output of M on input $\bar{e}$

Counterfactual Explanations

Given Boolean model M over set X of features, and input $\bar{e} : X \to \{0,1\}$, we call $Y \subseteq X$ a *counterfactual explanation* for $(M, \bar{e})$ if:

$M(\bar{e}) \neq M(\bar{e}')$, where $\bar{e}'$ is obtained from $\bar{e}$ by flipping the values of features in Y

We look for counterfactual explanations that are:

- *Minimum*: There is no counterfactual explanation of smaller size

Sufficient Reasons

Given Boolean model M over set X of features, and input $\bar{e} : X \to \{0,1\}$, we call $Y \subseteq X$ a *sufficient reason* for $(M, \bar{e})$ if:

$M(\bar{e}) = M(\bar{e}')$, for every $\bar{e}' : X \to \{0,1\}$ with $\bar{e}'(y) = \bar{e}(y)$ for each $y \in Y$

The output of M on $\bar{e}$ is invariant to interpretation of features in $X \setminus Y$

We look for sufficient reasons that are:
- *Minimum*: There is no sufficient reason of smaller size
- *Minimal*: There is no sufficient reason Z with $Z \subsetneq Y$

Example

Consider the model $M = (x_1 \vee x_2)$ and input $\bar{e} = (1,1)$

The minimum counterfactual explanation for $(M, \bar{e})$ is $\{x_1, x_2\}$

There are two minimum sufficient reasons for $(M, \bar{e})$:
$\{x_1\}$ and $\{x_2\}$

What’s the Need?

These are just two examples of explainability queries, but …
many other explainability notions have been proposed in the literature

The logical community has handled this challenge by developing
declarative query languages

Declarative query languages for explainability tasks could help in:
- Allowing more flexibility in writing queries for different applications
- Providing a clear syntax and semantics for such queries
- Making explainability tools more accessible for practitioners
- Opening a path for optimization of explainability tasks

Models as Logical Structures

We represent a Boolean model M over set X of features as a logical structure

$\mathscr{A}_M = (\{\perp,0,1\}^n, \subseteq, \mathrm{Pos})$, where:

- $\perp$ represents an *undefined value* in an input
- The domain $\{\perp,0,1\}^n$ is the set of all *partial* inputs
- The binary predicate $\subseteq$ contains pairs $(\bar{e}, \bar{e}')$ of partial inputs, such that $\bar{e}'$ is *more complete* than $\bar{e}$;
  e.g., $(1,0,\perp)$ is more complete than $(1, \perp, \perp)$
- The unary predicate Pos is interpreted as the set of *positive* inputs: $\{\bar{e} : X \to \{0,1\} \mid M(\bar{e}) = 1\}$

The Logic FOIL

FOIL is FO over structures of the form $\mathscr{A}_M$,
where M is a Boolean model
(Arenas, Baez, B., Pérez, Subsercaseaux, 2021)

Expressing Properties in FOIL

The set of *full* instances is defined as

$$\mathrm{Full}(x) \;=\; \forall y\,(x \subseteq y \;\rightarrow\; x = y)$$

The binary predicate full *completion* corresponds to

$$\mathrm{FullComp}(x, y) \;=\; x \subseteq y \wedge \mathrm{Full}(y)$$

The pairs of *same class* full instances are defined as

$$\mathrm{SameClass}(x, y) \;=\; \mathrm{Full}(x) \wedge \mathrm{Full}(y) \wedge \big(\mathrm{Pos}(x) \leftrightarrow \mathrm{Pos}(y)\big)$$

Minimal Sufficient Reasons

The binary predicate *sufficient reason* is defined as

$$\mathrm{SR}(x, y) \ = \ \mathrm{Full}(y) \ \wedge \ x \subseteq y \ \wedge \ \forall z\,(\mathrm{FullComp}(x, z) \ \rightarrow \ \mathrm{SameClass}(y, z)\big)$$

The binary predicate *minimal* sufficient reason is

$$\mathrm{mSR}(x, y) \ = \ \mathrm{SR}(x, y) \ \wedge \ \forall z\left(\mathrm{SR}(z, y) \wedge z \subseteq x \ \rightarrow \ z = x\right)$$

Two Issues with FOIL

Expressiveness

FOIL is not capable of expressing some useful explainability notions used in practice

Complexity

The evaluation complexity of FOIL is prohibitively expensive even over simple models

First Issue: Expressiveness

Theorem (Arenas, B., Bustamante, Caraball, Subercaseaux 2024)

There is no formula $\phi(x, y)$ in FOIL that checks whether x is a minimum sufficient reason for y over the class of decision trees

Intuitively, FOIL cannot compare the cardinalities of sets of features

Second Issue: Complexity

Theorem (Arenas, B., Bustamante, Caraball, Subercaseaux 2024)

For every $k \geq 1$, there exists a sentence ϕ_k in FOIL for which the evaluation problem over the class of decision trees is Σ_k^{P}-complete

Conflicting Requirements

Design a logic that, at the same time, can express useful notions of explainability often found in practice, and can be evaluated “efficiently” over decision trees

What "efficient" means in this case?

Formulas that can be evaluated as
Boolean combinations of NP languages

This is unavoidable since even evaluating
minimum sufficient reason over decision trees is coNP-complete
(B., Monet, Pérez, Subsercaseaux, 2020)

On the positive side, it allows us to use SAT solvers technology
to evaluate formulas in the language

The Logic DT-FOIL

This logic is specifically tailored for decision trees

It is composed of three layers:

Atomic formulas that can only compare
syntactic properties of partial inputs
(can be evaluated in PTIME)

Guarded formulas that allow quantification over partial inputs that
represent nodes and positive leaves of decision trees
(can be evaluated in PTIME)

DT-FOIL formulas, which are Boolean combinations of
quantified guarded formulas
(correspond to Boolean combinations of NP languages)

Atomic Formulas

These formulas correspond to FO over the following syntactic relations:

Binary relation $\subseteq$ that contains all pairs $(\bar{e}, \bar{e}')$ of partial inputs such that $\bar{e}'$ is more complete than $\bar{e}$

Binary relation $\preceq$ that contains all pairs $(\bar{e}, \bar{e}')$ of partial inputs such that # of undefined features in $\bar{e}$ is no larger than in $\bar{e}'$ (this allows to compare cardinalities of sets of features)

Guarded Formulas

These are recursively defined as follows:

(1) Atomic formulas are guarded formulas

(2) Guarded formulas are closed under Boolean combinations

(3) If ϕ is a guarded formula, then

$$\exists x(\mathrm{Node}(x) \wedge \phi) \qquad \exists x(\mathrm{PosLeaf}(x) \wedge \phi)$$

are also guarded formulas

$\mathrm{Node}(x)$ holds for
those partial instances that correspond to nodes of the decision tree

$\mathrm{PosLeaf}(x)$ holds for
those partial instances that correspond to positive leaves of the decision tree

DT-FOIL Formulas

These are recursively defined as follows:

(1) Guarded formulas are DT-FOIL formulas

(2) DT-FOIL formulas are closed under Boolean combinations

(3) If ϕ is a guarded formula, then

$$\exists x_1 \cdots \exists x_l \phi$$

is a DT-FOIL formula

Expressiveness of DT-FOIL

DT-FOIL is capable of expressing many
useful explainability properties often found in practice

An Example

The following atomic formula checks whether the features defined in both partial inputs x and y have the same value:

$$\mathrm{Cons}(x, y) \;=\; \exists z(x \subseteq z \wedge y \subseteq z)$$

The following guarded formula checks if partial instance x is a leaf:

$$\mathrm{Leaf}(x) \;=\; \mathrm{Node}(x) \wedge \forall y(\mathrm{Node}(y) \wedge x \subseteq y \rightarrow y = x)$$

The following guarded formula checks whether all "completions" of a partial instance x are positive:

$$\mathrm{AllPos}(x) \;=\; \forall y(\mathrm{Node}(y) \rightarrow (\mathrm{Leaf}(y) \wedge \mathrm{Cons}(x, y)) \rightarrow \mathrm{PosLeaf}(y))$$

An Example

The following guarded formula checks whether x is a sufficient reason for y:

$$\mathrm{SR}(x, y) \;=\; x \subseteq y \land \mathrm{Full}(y) \land (\mathrm{Pos}(x) \to \mathrm{AllPos}(x)) \land (\neg \mathrm{Pos}(x) \to \mathrm{AllNeg}(x))$$

The following DT-FOIL formula checks whether x is a minimum sufficient reason for y:

$$\mathrm{SR}(x, y) \land \forall z (\mathrm{SR}(z, y) \to x \preceq z)$$

Complexity of DT-FOIL

Theorem (Arenas, B., Bustamante, Caraball, Subercaseaux 2024)

Let ϕ be a fixed DT-FOIL formula.

The problem of evaluating ϕ over decision trees can be solved by a constant number of calls to an NP oracle

THANKS