IA hybride, vue d'ensemble et quelques applications à la classification d'images

Conférence EGC - Strasbourg 2025

Céline Hudelot
Laboratoire MICS - CentraleSupélec

30 janvier 2025

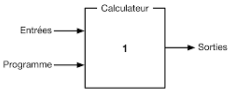
Plan

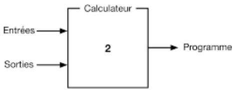
AI : two *antagonistic* approaches [a]

a. D. Cardon et al - La Revanche des neurones - https://hal.archives-ouvertes.fr/hal-02005537/document

Two different assumptions

- Human reasoning and knowledge are complex : knowledge implicitly in data.
  - Statistic or data-centric AI - Connectionist approaches - Learning from data.
  - Exploitation of the past experience represented by annotated data, building calibrated predictive models from it.
- Human reasoning can be captured, even if partially incomplete : explicit representation of knowledge (using symbols rather that statistics to represent the world).
  - Symbolic AI - Based on the modeling of logical reasoning, on formalisms for knowledge representation and reasoning.





Hybrid AI

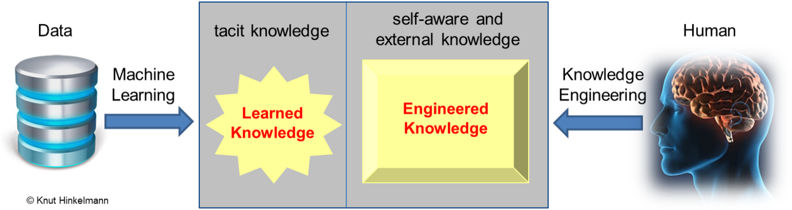


FIGURE – Source : `https://www.aaai-make.info/`

Bringing together, for added value, data-driven AI with symbolic and knowledge-oriented AI to answer their respective weaknesses.

Hybrid AI : why ?

Because :

- Lack of high level reasoning in deep-learning [Bottou,2011][a].
- Deep neural models are *black box* models that can be easily fooled.
- More to predict that what is visible or readable (the knowledge is not totally inside the data).
- For some decision-based AI systems, the rules are to be told (ethics, policies, laws...) : ⇒ need of Knowledge Representation and Reasoning.
- To apply appropriate safety standards while providing explainable outcomes guided by concepts from background knowledge : trustworthy AI and human-like cognition and decision (**learning, reasoning and collaboration**).

a. Bottou, Leon. (2011). From Machine Learning to Machine Reasoning. Computing Research Repository - CORR. 94. 10.1007/s10994-013-5335-x.

Hybrid AI

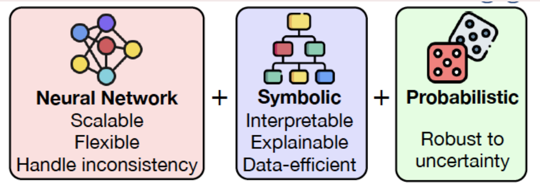
An important topic with different names and sub-fields

- **Neuro-Symbolic Artificial Intelligence** : bringing together the neural and the symbolic traditions in AI (https://people.cs.ksu.edu/~hitzler/nesy/)
  - Neural : use of artificial neural networks, or connectionist systems.
  - Symbolic : AI approaches that are based on explicit symbol manipulation.
- **Informed Machine Learning** : integrating Prior Knowledge into Learning Systems.
  - (von Rueden et al, 21) [a] Learning from an hybrid information source that consists of data and prior knowledge. The prior knowledge comes from an independent source, is given by formal representations and is explicitly integrated into the ML pipeline
- **Knowledge Reasoning meets Machine Learning**
  - KR2ML workshops (https://kr2ml.github.io/)

a. https://arxiv.org/pdf/1903.12394.pdf

Hybrid AI : Objectives

Best of both worlds perspective



Source : [Wan et al, 24] [a]

a. https://arxiv.org/pdf/2401.01040

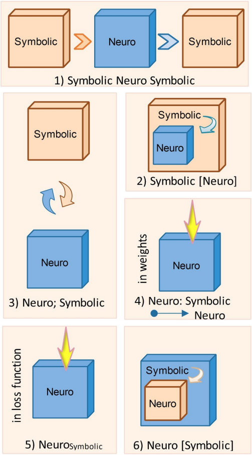
Objectives : which gains ?

- **Performance, generalization** : e.g. neuro-symbolic concept learner [Mao et al, 19], Neural-Symbolic Language Model [Demeter et al, 20]...
- **Explainability, Trust** : e.g. [Finzel et al, 2022], sdrl [Lyu et al, 2019]...
- **Frugality** : e.g. FrugalLLM : LLMs with symbolic solvers [Dutta et al, 2024]

Hybrid AI : some taxonomies

Kautz's Neurosymbolic taxonomy

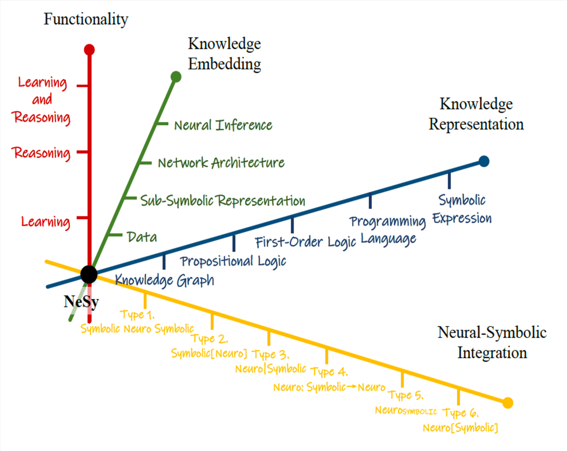
- **Symbolic Neuro Symbolic**
- **Symbolic[Neuro]**
- **Neuro ;Symbolic**
- **Neuro :Symbolic $\rightarrow$ Neuro**
- **Neuro_Symbolic**
- **Neuro[Symbolic]**



Source : [Hassan et al, 22]

Henri Kautz, The Third AI Summer : AAAI Robert S. Engelmore Memorial Lecture
Hassan et al, 22 : `https://arxiv.org/pdf/2208.00374`

Hybrid AI : some taxonomies



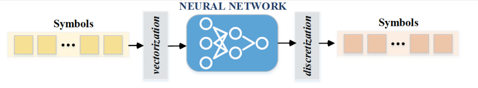
[Wang et al] Towards Data-And Knowledge-Driven AI : A Survey on Neuro-Symbolic Computing, in PAMI 2025 [1]

1. https://arxiv.org/abs/2210.15889

Hybrid AI : Kautz's Neurosymbolic taxonomy

Symbolic Neuro Symbolic

Input and output are presented in symbolic form, all the processing is neural.



Source : Wang et al, PAMI 2025

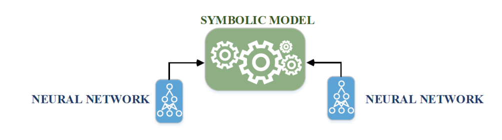
Example : Deep learning procedure for NLP

Input symbols (words) are converted to vector embeddings (Glove ; Word2VEc), processed by the neural model whose output embeddings are converted to symbols.

Hybrid AI : Kautz's Neurosymbolic taxonomy

Symbolic[Neuro] or Neuro Subroutines

Symbolic systems, where neural modules are internally used as subroutines within a comprehensive symbolic problem solver.



Source : Wang et al, PAMI 2025

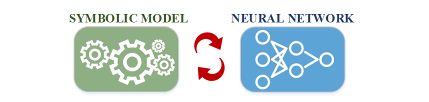
Example : Alpha Go

Monte Carlo Tree Search (symbolic solver) and NN state estimators for learning statistical patterns.

Hybrid AI : Kautz's Neurosymbolic taxonomy

Neuro | Symbolic or Neural Learning + Symbolic Solver

Neural and symbolic parts focus on different but complementary tasks in a big pipeline.



- Example : a neural network focusing on one task (e.g. object detection) interacts via its input and output with a symbolic system specialized in a complementary task (e.g. question answering).
- Many NSAI algorithms fall into this category.

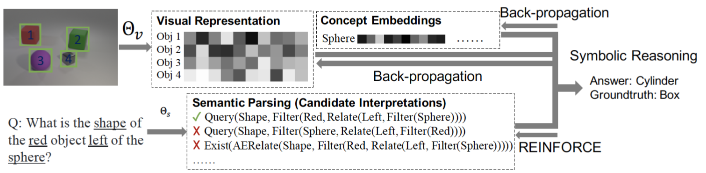
Source : Wang et al, PAMI 2025

Hybrid AI : Kautz's Neurosymbolic taxonomy

Some examples of Neuro | Symbolic approaches

Example : Neuro-symbolic concept learner

A neural perception module learns visual concepts and a symbolic reasoning module executes symbolic programs on the concept representations for question answering (Mao et al, 2019) [a]



a. https://arxiv.org/abs/1904.12584

Hybrid AI : Kautz's Neurosymbolic taxonomy

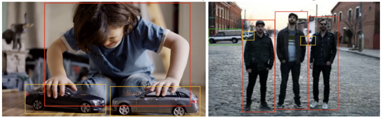
Some examples of Neuro | Symbolic approaches

Example : NeurASP : Embracing Neural Networks into Answer Set Programming

Idea : the neural network output is treated as a probability distribution over atomic facts in answer set program

4. NeurASP extends classification to context relational classification

Q: What are the cars and toy-cars in these images?



- By default, we believe person is smaller than car.

```
smaller(B,B') ← label(B)=person, label(B')=car, not ~smaller(B,B').
```

- On the other hand, there are some exceptions.

```
~smaller(B,B') ← box(B,X1,Y1,X2,Y2), box(B',X1',Y1',X2',Y2'),
                  Y2 ≤ Y2', |X1−X2|×|Y1−Y2| > |X1'−X2'|×|Y1'−Y2'|.
toy(B') ← label(B)=person, label(B')=car, smaller(B',B).
```

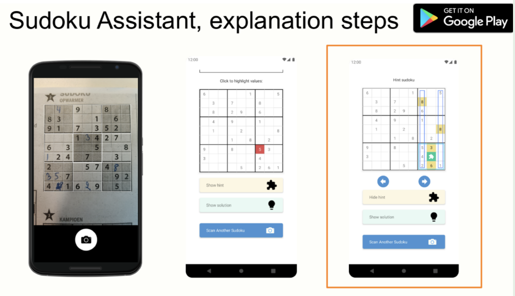
~~[Yang et al] NeurASP : Embracing Neural Networks int~~o Answer Set Programming [2]

2. https://arxiv.org/abs/2307.07700

Hybrid AI : Kautz's Neurosymbolic taxonomy

Some examples of Neuro | Symbolic approaches

Example : Tias Guns, Decision-focus learning

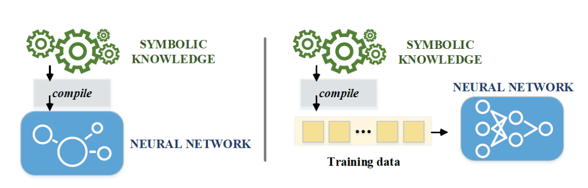


See `https://people.cs.kuleuven.be/~tias.guns/`

Hybrid AI : Kautz's Neurosymbolic taxonomy

Neuro :Symbolic → Neuro

Use of Symbolic rules into NNs to guide the learning process : symbolic knowledge is compiled into the structure of neural models.



Hybrid AI : Kautz's Neurosymbolic taxonomy

Example : GCN-based embedder

Vector based representations learning of symbolic knowledge to incorporate symbolic domain knowledge into connectionist architectures (Xie et al, 2019) [a]

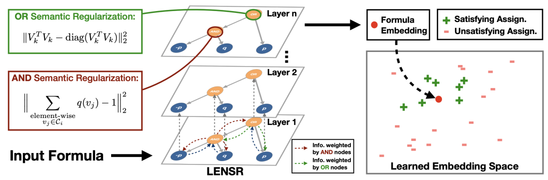


Figure 1: LENSR overview. Our GCN-based embedder projects logic graphs representing formulae or assignments onto a manifold where entailment is related to distance; satisfying assignments are closer to the associated formula. Such a space enables fast approximate entailment checks — we use this embedding space to form logic losses that regularize deep neural networks for a target task.

a. https://arxiv.org/pdf/1909.01161.pdf

Hybrid AI : Kautz's Neurosymbolic taxonomy

Examples

- Logical NNs (LNNs) : encode knowledge or domain expertise as symbolic rules (first-order logic or fuzzy logic) that act as constraints on the NN output [Riegel et al, 2020] [a]
- Deep learning for symbolic mathematics [Lample, 2019] [b], AlphaProof...
- Differentiable inductive logic programming (ILP) [Evans, 2018] [c]

---

a. https://arxiv.org/abs/2006.13155

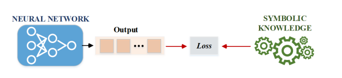
b. https://arxiv.org/abs/1912.01412

c. https://arxiv.org/abs/1711.04574

Hybrid AI : Kautz's Neurosymbolic taxonomy

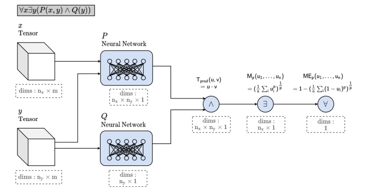
Neuro_Symbolic

Turns symbolic knowledge into additional soft-constraints in the loss function used to train DNNs



Example

Logic Tensor Networks (LTNs) ((Badreddine et al, 2022)[a]. First-order logic formulae are translated as fuzzy relations on real numbers for neural computing to allow gradient based sub-symbolic learning.

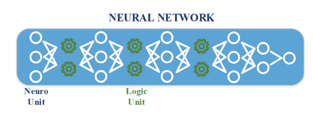


a. https://www.sciencedirect.com/science/article/abs/pii/S0004370221002009

Hybrid AI : Kautz's Neurosymbolic taxonomy

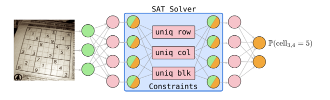
Neuro[Symbolic]

Fully-integrated system, i.e. true symbolic reasoning inside a neural engine.



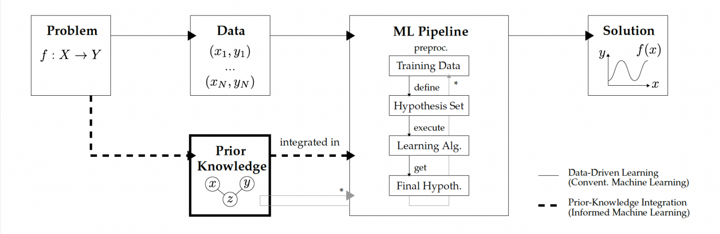
Example : Satnet

Imitating logical reasoning with tensor calculus to learn the execution of symbolic operations through neural networks.
Satnet : A layer that enables end-to-end learning of both the constraints and solutions of logic problems and a smoothed differentiable (maximum) satisfiability solver that can be integrated into the loop of deep learning systems.(Wang et al, 2019) [a]



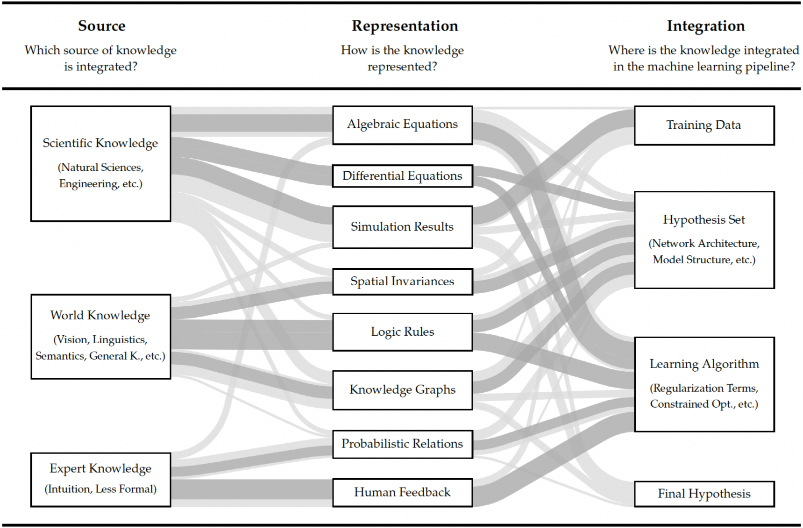
Hybrid AI (or Informed ML)

Principle (Rueden et al, 2021)



1. **Source** : Which source of knowledge is integrated ?
2. **Representation** : How is the knowledge represented ?
3. **Integration** : Where in the learning pipeline is it integrated ?
4. **Task** : What is the task ?
5. **Expected benefits** of the integration : explainability, performance, frugality ?

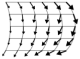
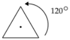
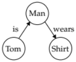
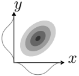
Hybrid AI (or Informed ML)



[Rueden et al, 2021]

Hybrid AI (or Informed ML)

Knowledge

Algebraic Equations	Differential Equations	Simulation Results	Spatial Invariances	Logic Rules	Knowledge Graphs	Probabilistic Relations	Human Feedback
$E = m \cdot c^2$ $v \leqslant c$	$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$ $F(x) = m \frac{d^2 x}{dt^2}$			$A \wedge B \Rightarrow C$			

[Rueden et al, 2021]

Plan

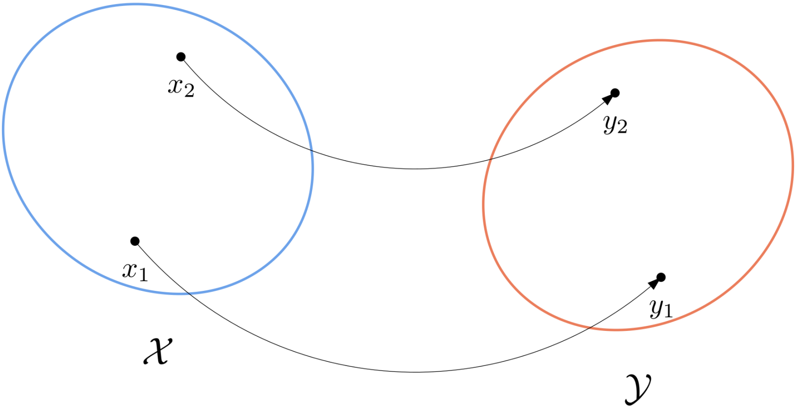
Improving neural classification with Logical Prior Knowledge

Workshop on Composite AI (CompAI),ECAI 2024

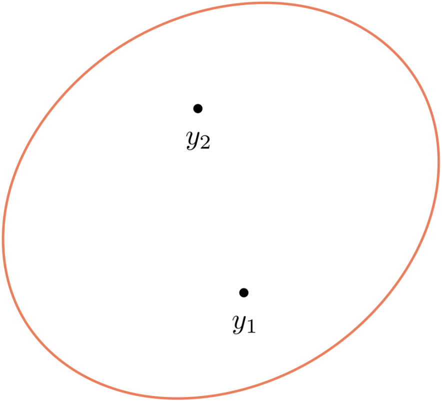
PhD thesis : Arthur Ledaguenel

Mostepha Khouadjia

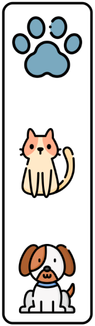
Background : Supervised learning



Background : Classification tasks

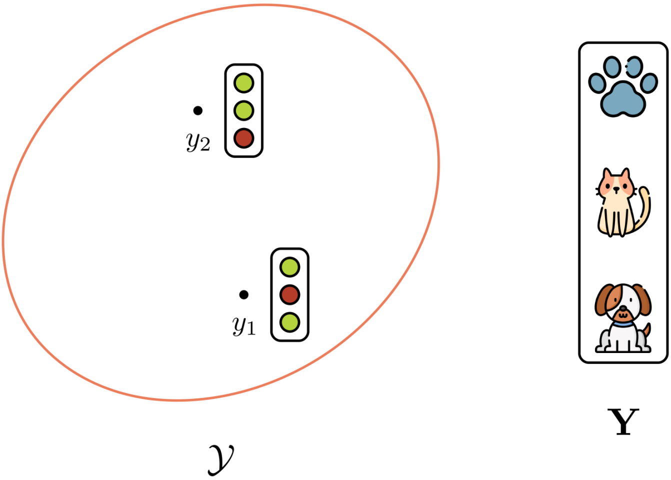


$\mathcal{Y}$



$\mathbf{Y}$

Background : Classification tasks

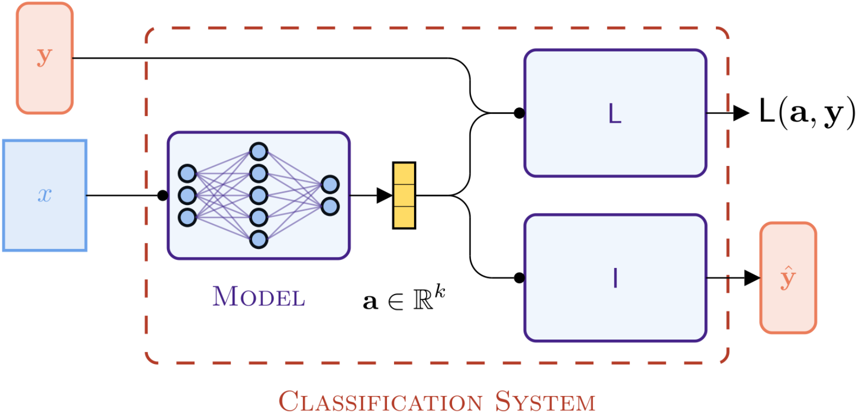


Background : Classification tasks

State

Given a finite set of variables $\mathbf{Y}$, a **state** is an element of $\mathbb{B}^{\mathbf{Y}}$, where $\mathbb{B} := \{0, 1\}$ is the set of boolean values.

Background : Neural classification system



Background : Independent multi-label classification

A neural classification system (M, L_{imc}, I_{imc}) performs **independent multi-label classification** (*imc*) iff :

$$L_{imc}(\mathbf{a}, \mathbf{y}) := -\log\left(\sum_{1\leq i\leq k} y_i \cdot p_i + (1-y_i)\cdot(1-p_i)\right)$$

$$I_{imc}(\mathbf{a}) := \mathbf{1}[\mathbf{a} \geq 0]$$

The loss corresponds to the **negative log-likelihood** of the label on independent Bernouilli variables $\mathcal{B}(p_i)_{1\leq i\leq k}$ with $p_i = s(a_i)$, where $s(\mathbf{a}) = (\frac{e^{a_j}}{1+e^{a_j}})_{1\leq j\leq k}$ is the sigmoid function.

Background : Independent multi-label classification

A neural classification system (M, L_{imc}, I_{imc}) performs **independent multi-label classification** (*imc*) iff :

$$L_{imc}(\mathbf{a}, \mathbf{y}) := -\log\left(\sum_{1\leq i\leq k} y_i \cdot p_i + (1-y_i)\cdot(1-p_i)\right)$$

$$I_{imc}(\mathbf{a}) := \mathbf{1}[\mathbf{a} \geq 0]$$

The loss corresponds to the **negative log-likelihood** of the label on independent Bernouilli variables $\mathcal{B}(p_i)_{1\leq i\leq k}$ with $p_i = s(a_i)$, where $s(\mathbf{a}) = (\frac{e^{a_j}}{1+e^{a_j}})_{1\leq j\leq k}$ is the sigmoid function.

Background : Knowledge Representation

- **Knowledge** about a **world** tells us in what **states** this world can be observed.
- In our approach : **propositional knowledge** :
  - The states correspond to subsets of a discrete set of variables $\mathbf{Y}$
  - Set of possible states is $\mathbb{B}^{\mathbf{Y}}$
  - A state y can be seen as a subset of $\mathbf{Y}$ as well as an application that maps each variable to $\mathbb{B}$
  - Knowledge tells us what combinations of variables can be observed in the world : defines a set of states that are considered valid.
  - Abstract representation of the knowledge through a **boolean function** $f : \mathbb{B}^{\mathbf{Y}} \mapsto \mathbb{B}$ that maps all states to boolean values or as a subset of $\mathbb{B}^{\mathbf{Y}}$
- **Propositional language** : concrete language to represent the knowledge

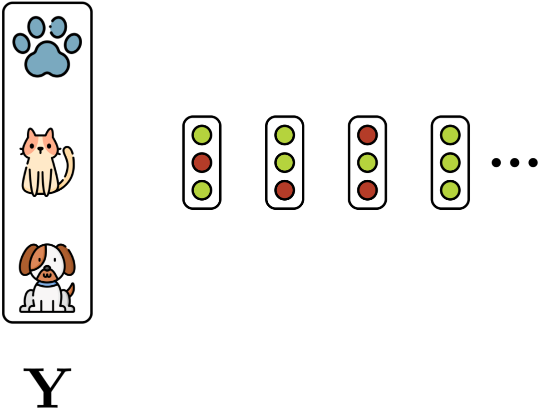
Background : Boolean functions

Boolean function

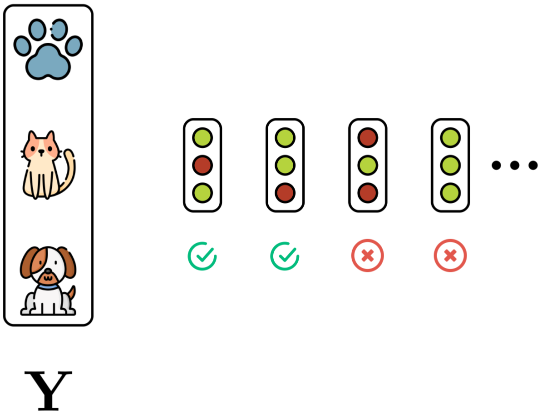
Given a finite set of variables $\mathbf{Y}$, a **boolean function** is a function $f : \mathbb{B}^{\mathbf{Y}} \mapsto \mathbb{B}$ that maps all states to boolean values.

A boolean function can also be seen as a set of states.
The set of boolean functions on $\mathbf{Y}$ is $\mathbb{B}^{\mathbb{B}^{\mathbf{Y}}}$.

Background : Boolean functions



Background : Boolean functions



Background : Propositional logic

Propositional formulas

The set of **propositional formulas** on a signature $\mathbf{Y}$, noted $\mathcal{F}_{PL}(\mathbf{Y})$, is formed inductively from variables and other formulas by using unary ($\neg$) or binary ($\vee, \wedge$) connectives :

$$\phi := \quad v \quad | \quad \neg\phi \quad | \quad \phi \wedge \varphi \quad | \quad \phi \vee \varphi,$$
$$v \in \mathbf{Y}, \phi, \varphi \in \mathcal{F}_{PL}(\mathbf{Y})$$

We simply note $\mathcal{F}_{PL}$ when the signature is clear from context.

Background : Propositional logic

Valuation

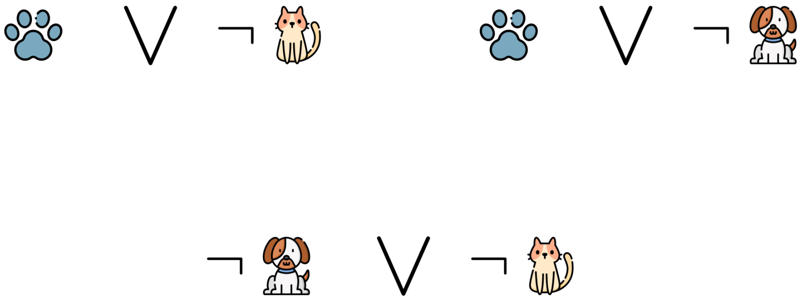
A state $\mathbf{y} \in \mathbb{B}^{\mathbf{Y}}$ inductively defines a valuation $\nu_{\mathbf{y}} \in \mathbb{B}^{\mathcal{F}_{PL}}$:

$$\forall v \in \mathbf{Y}, \nu_{\mathbf{y}}(v) = \mathbf{y}(v)$$
$$\forall \phi, \varphi \in \mathcal{F}_{PL},$$
$$\nu_{\mathbf{y}}(\neg\phi) = 1 - \nu_{\mathbf{y}}(\phi)$$
$$\nu_{\mathbf{y}}(\phi \wedge \varphi) = \nu_{\mathbf{y}}(\phi) \cdot \nu_{\mathbf{y}}(\varphi)$$
$$\nu_{\mathbf{y}}(\phi \vee \varphi) = \nu_{\mathbf{y}}(\phi) + \nu_{\mathbf{y}}(\varphi) - \nu_{\mathbf{y}}(\phi) \cdot \nu_{\mathbf{y}}(\varphi)$$

A propositional formula κ represents the boolean function f such that :

$$\forall \mathbf{y}, f(\mathbf{y}) = \nu_{\mathbf{y}}(\kappa)$$

Propositional logic



Background : Distributions

Motivation

- One Challenge of neurosymbolic AI : **bridge the gap** between the discrete nature of logic and the continuous nature of neural networks.
- **Probabilistic reasoning** can provide the **interface** between these two realms by allowing us to reason about uncertain facts.
- The ingredients :
  - A probability distribution on a set of boolean variables $\mathbf{Y}$
  - To define internal operations between distributions, like multiplication, we extend this definition to un-normalized distributions.
  - The **mode** of a distribution is its most probable state
  - A standard distribution is the **exponential probability distribution**, which is parameterized by a vector of logits a, one for each variable in $\mathbf{Y}$.

Background : Probabilistic Reasoning

When belief about random variables is expressed through a probability distribution and new information is collected in the form of evidence (i.e., a partial assignment of the variables), we are interested in two things :

- computing the probability of such evidence
- updating our beliefs using Bayes' rules by conditioning the distribution on the evidence.

Background : Probabilistic Reasoning

Probabilistic reasoning allows to perform **the same operations with logical knowledge in place of evidence**.

- Probability distribution $\mathcal{P}$ on variables $\mathbf{Y}$
- A satisfiable theory κ from a propositional language.
- Computing $\mathcal{P}(\kappa|\mathbf{a})$ is a **counting** problem called **Probabilistic Query Evaluation** (PQE).
- Computing the mode of $\mathcal{P}(\cdot|\mathbf{a}, \kappa)$ is an **optimization** problem called **Most Probable Explanation** (MPE).

Solving these probabilistic reasoning problems is at the core of many neurosymbolic techniques

Background : Distributions and Probabilistic Reasoning

Distribution

A joint probability distribution for a finite set of binary variables $\mathbf{Y}$ is an application :

$$\mathcal{P} : \mathbb{B}^{\mathbf{Y}} \mapsto \mathbb{R}^{+} \quad \text{such that} \quad \sum_{\mathbf{y} \in \mathbb{B}^{\mathbf{Y}}} \mathcal{P}(\mathbf{y}) = 1$$

To allow internal operations between distributions (multiplication) we also define (un-normalized) distributions on $\mathbf{Y}$:

$$\mathcal{E} : \mathbb{B}^{\mathbf{Y}} \mapsto \mathbb{R}^{+}$$

The **null distribution** is the application that maps all states to 0. A **boolean function** on $\mathbf{Y}$ is a distribution on $\mathbf{Y}$ that maps all states to $\mathbb{B}$.

Background : Distributions

Partition function

The **partition function** Z maps each distribution to its sum, ie :

$$Z : \mathcal{E} \mapsto \sum_{\mathbf{y} \in \mathbb{B}^{\mathbf{Y}}} \mathcal{E}(\mathbf{y})$$

We note $\overline{\mathcal{E}} := \frac{\mathcal{E}}{Z(\mathcal{E})}$ the normalized distribution (when $\mathcal{E}$ is non-null).

Background : Exponential Distributions

Exponential Distribution

Given activation scores $\mathbf{a} := (a_i, ..., a_k) \in \mathbb{R}^k$, one can define the **exponential distribution** :

$$\mathcal{E}(\cdot|\mathbf{a}) : \mathbb{B}^{\mathbf{Y}} \to [0,1], \mathbf{y} \mapsto \prod_{1 \leq i \leq k} e^{a_i \cdot y_i}$$

The **independent multi-label probability distribution** is then :

$$\mathcal{P}(\cdot|\mathbf{a}) = \overline{\mathcal{E}(\cdot|\mathbf{a})}$$

The independent multi-label probability distribution is the joint distribution of independent Bernouilli variables $\mathcal{B}(p_i)_{1 \leq i \leq k}$ with $p_i = s(a_i)$, where $s(\mathbf{a}) = (\frac{e^{a_j}}{1+e^{a_j}})_{1 \leq j \leq k}$ is the sigmoid function.

Background : Distributions

Independent multi-label probability distribution

$$L_{imc}(\mathbf{a}, \mathbf{y}) = -\log(\mathcal{P}(\mathbf{y}|\mathbf{a}))$$

$$I_{imc}(\mathbf{a}) = \underset{\mathbf{y} \in \mathbb{B}^k}{\arg\max} \mathcal{P}(\mathbf{y}|\mathbf{a})$$

Background : Probabilistic reasoning

Assume a finite set of variables $\mathbf{Y}$, a probability distribution $\mathcal{P}$ and a propositional formula $\kappa \in \mathcal{F}_{PL}$ representing a boolean function f.

Probabilistic reasoning

The **probability** of κ under $\mathcal{P}$ is :

$$\mathcal{P}(\kappa) := Z(\mathcal{P} \cdot f) = \sum_{\mathbf{y} \in \mathbb{B}^{\mathbf{Y}}} \mathcal{P}(\mathbf{y}) \cdot f(\mathbf{y}) \tag{1}$$

The distribution $\mathcal{P}$ **conditioned on** κ, noted $\mathcal{P}(\cdot|\kappa)$, is :

$$\mathcal{P}(\cdot|\kappa) := \overline{\mathcal{P} \cdot f} \tag{2}$$

Background : Probabilistic reasoning

Assume a finite set of variables **Y**, a probability distribution $\mathcal{P}$ and a propositional formula $\kappa \in \mathcal{F}_{PL}$ representing a boolean function f.

Probabilistic reasoning

The **probability** of κ under $\mathcal{P}$ is :

$$\mathcal{P}(\kappa) := Z(\mathcal{P} \cdot f) = \sum_{\mathbf{y} \in \mathbb{B}^{\mathbf{Y}}} \mathcal{P}(\mathbf{y}) \cdot f(\mathbf{y}) \tag{1}$$

The distribution $\mathcal{P}$ **conditioned on** κ, noted $\mathcal{P}(\cdot|\kappa)$, is :

$$\mathcal{P}(\cdot|\kappa) := \overline{\mathcal{P} \cdot f} \tag{2}$$

Background : Probabilistic reasoning

By convention, we note :

$$\mathcal{P}(\kappa|\mathbf{a}) := Z(\mathcal{P}(\cdot|\mathbf{a}) \cdot f)$$

$$\mathcal{P}(\cdot|\mathbf{a}, \kappa) := \frac{\mathcal{P}(\cdot|\mathbf{a}) \cdot f}{\mathcal{P}(T|\mathbf{a})}$$

Probabilistic Query Evaluation

Computing $\mathcal{P}(\kappa|\mathbf{a})$ is a **counting** problem called **Probabilistic Query Evaluation** (PQE).

Most Probable Explanation

Computing the mode of $\mathcal{P}(\cdot|\mathbf{a}, \kappa)$ is an **optimization** problem called **Most Probable Explanation** (MPE).

Task : Informed classification

Prior knowledge is available about a classification task.
Goal : improve our neural classification system by integrating this knowledge into its design.

Informed domain

An **informed** classification domain $\mathcal{D} := (\mathcal{X}, \mathcal{Y}, \kappa)$ is composed of :

- an **input domain** $\mathcal{X}$
- an **output domain** $\mathcal{Y} := \mathbb{B}^{\mathbf{Y}}$
- a **satisfiable formula** $\kappa \in \mathcal{F}^*_{PL}(\mathbf{Y})$

Dataset

A supervised **dataset** on an informed domain $\mathcal{D} := (\mathcal{X}, \mathcal{Y}, \kappa)$ is a collection of pairs $D := (x^i, \mathbf{y}^i)_{1 \leq i \leq d} \in (\mathcal{X} \times \mathcal{Y})^d$ where :

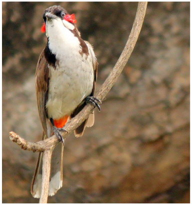
$$\forall 1 \leq i \leq d, \mathbf{y}^i \models \kappa$$

Task : Informed classification

Many informed classification tasks

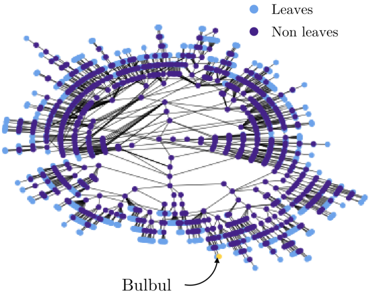
- **Categorical classification** : one and only one output variable is true for a given input sample.The sigmoid layer is replaced by a softmax layer and the variable with the maximum score is predicted.
- **Hierarchical classification** : hierarchical knowledge on the variables.
- Propositional knowledge can be used to define very diverse output spaces
  - Sudoku solutions.
  - Simple paths in a graph.
  - Preference rankings.
  - Matchings in a graph.

Task : Hierarchical classification



Bulbul

$\mathbf{y} \in \{0,1\}^{1860}$



$$\kappa_H := \left(\bigwedge_{(i,j) \in E_h} Y_i \vee \neg Y_j \right) \wedge \left(\bigwedge_{(i,j) \in E_e} (\neg Y_i \vee \neg Y_j) \right) \tag{3}$$

the first part ensures that a son node cannot be *true* if its father node is not and the second part prevents two mutually exclusive nodes to be *true* simultaneously.

Task

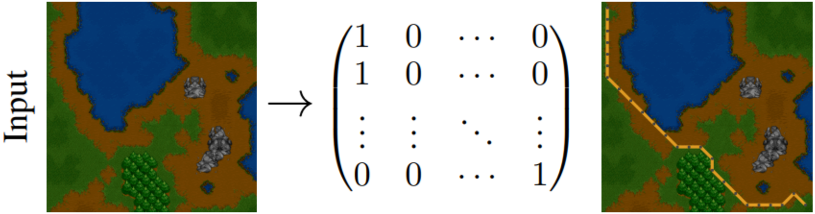


FIGURE – Warcraft shortest path instance from [Pogancic, 2019]

[Pogancic, 2019] Differentiation of Blackbox Combinatorial Solvers [3]

3. `https://openreview.net/forum?id=BkevoJSYPB`

Task

FIGURE – MNIST Sudoku instance from [Augustine, 22]

[Augustine, 22] Visual Sudoku Puzzle Classification : A Suite of Collective Neuro-Symbolic Tasks [4]

4. https://ceur-ws.org/Vol-3212/paper2.pdf

Neurosymbolic techniques

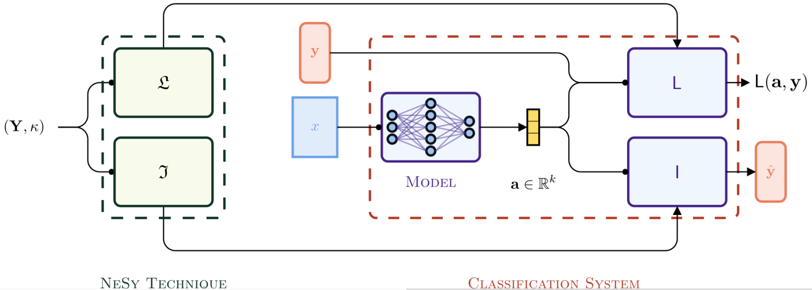
The purpose of a neurosymbolic technique is to automatically derive appropriate loss and inference modules from prior knowledge.

Neurosymbolic technique

A (model agnostic) **neurosymbolic technique** is $T := (L, I)$ such that for any finite set of variables $\mathbf{Y}$ and formula $\kappa \in \mathcal{F}_{PL}(\mathbf{Y})$:

$$L(\mathbf{Y}, \kappa) := L : \mathbb{R}^k \times \mathcal{Y} \mapsto \mathbb{R}^+$$

$$I(\mathbf{Y}, \kappa) := I : \mathbb{R}^k \mapsto \mathcal{Y}$$



Semantic regularization

Principle : Use the probability of the prior knowledge based on output scores of the model as a **regularization** term.

Semantic regularization (with coefficient $\lambda > 0$) is $T_r^\lambda := (L_r^\lambda, I_r^\lambda)$ for any finite set of variables $\mathbf{Y}$ and formula $\kappa \in \mathcal{F}_{PL}^*(\mathbf{Y})$:

$$L_r^\lambda(\mathbf{Y}, \kappa) : (\mathbf{a}, \mathbf{y}) \rightarrow L_{imc}(\mathbf{a}, \mathbf{y}) - \lambda. \log(\mathcal{P}(\kappa|\mathbf{a})) \tag{4}$$

$$I_r^\lambda(\mathbf{Y}, \kappa) : \mathbf{a} \rightarrow I_{imc}(\mathbf{a}) \tag{5}$$

Introduced for propositional logic in [Xu, 2018] , inspired by fuzzy regularization techniques [Diligenti, 2017 ; Marra, 2019 ; Badreddine,2022]

Semantic conditioning

Idea : Following the previous probabilistic interpretation, a natural way to integrate prior knowledge κ into a neural classification system is to condition the distribution $\mathcal{P}(.|M(x,\theta))$ on κ.

Semantic conditioning is $T_{sc} := (L_{sc}, I_{sc})$ such that for any finite set of variables $\mathbf{Y}$ and formula $\kappa \in \mathcal{F}^*_{PL}(\mathbf{Y})$:

$$L_{sc}(\mathbf{Y},\kappa) : (\mathbf{a},\mathbf{y}) \to -\log(\mathcal{P}(\mathbf{y}|\mathbf{a},\kappa)) \tag{6}$$

$$I_{sc}(\mathbf{Y},\kappa) : \mathbf{a} \to \underset{\mathbf{y}\in\mathbb{B}^{\mathbf{Y}}}{} \mathcal{P}(\mathbf{y}|\mathbf{a},\kappa) \tag{7}$$

Introduced under different forms for HEX-graph constraints [Deng,2014], boolean circuits [Ahmed,2022], ASP programs [Yang,2020], Prolog programs [Manhaeve,2021].

Semantic conditioning at inference

Idea : applies conditioning only in the inference module (i.e., infers the most probable state that satisfies prior knowledge) while retaining the standard negative log-likelihood loss.

Semantic conditioning at inference for any abstract logic $\mathcal{L} := (\mathcal{T}, s)$ is $T_{sc} := (L_{sc}, I_{sc})$ such that for any finite set of variables $\mathbf{Y}$ and theory $\kappa \in \mathcal{F}^*_{PL}(\mathbf{Y})$:

$$L_{sci}(\mathbf{Y}, \kappa) : (\mathbf{a}, \mathbf{y}) \rightarrow L_{imc}(\mathbf{a}, \mathbf{y}) \tag{8}$$

$$I_{sci}(\mathbf{Y}, \kappa) : \mathbf{a} \rightarrow \underset{\mathbf{y}\in\mathbb{B}^{\mathbf{Y}}}{} \mathcal{P}(\mathbf{y}|\mathbf{a}, \kappa) \tag{9}$$

Introduced in [Ledaguenel,2024]

Advantages : Properties

The propose framework enables to analyze specific properties of neurosymbolic techniques such that

- **Syntactic invariance** : equivalent formulas produce identical loss and inference modules.
- **Consistency** : the inference module can only produce outputs that satisfy the prior knowledge.

Experimental evaluation

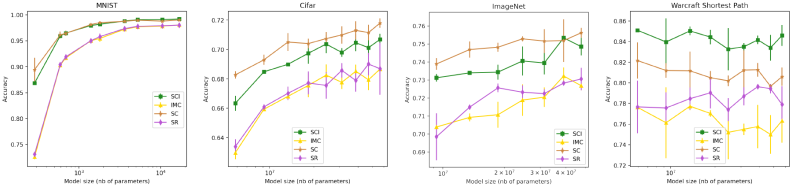
Empirical evaluation of the impact of neurosymbolic techniques on four informed classification tasks :

- A categorical task : MNIST dataset
- Two hierarchical tasks : Cifar-100 and ImageNet
- A simple path prediction task : Warcraft Shortest Path (WSP) dataset

A multiscale evaluation

- Most papers in the field evaluate the benefits of their neurosymbolic technique on a single neural network architecture.
- For each task, we select a single architectural design that can be **scaled to various sizes** and compared the performance of the neurosymbolic techniques against an **uninformed baseline**.
- Metrix : **exact accuracy** : the share of instances which are well classified on all labels.

Results


MNIST
Cifar
ImageNet
Warcraft Shortest Path
Accuracy
Model size (nb of parameters)
SCI
IMC
SC
SR

Results

Observations

- **Observation 1.** Semantic conditioning and semantic conditioning at inference outperform semantic regularization and independent multi-label classification across tasks and model scales.
- **Observation 2.** Except for the larger networks on Warcraft Shortest Path, semantic regularization brings little benefits in terms of accuracy compared to independent multi-label classification.
- **Observation 3.** On MNIST, Cifar and ImageNet, semantic conditioning at inference retains most of the performance gains (about 75%) of semantic conditioning, despite only integrating knowledge during inference. It even outperforms semantic conditioning on Warcraft Shortest Path.
- **Observation 4.**Accuracy gains of semantic conditioning at inference tend to decrease and converge towards a significantly positive value as the accuracy of the neural network increases.

Computational complexity

- Are these techniques tractable in the general case ?
  $\rightarrow$ No
- For which fragments can we implement neurosymbolic techniques tractably ?
  $\rightarrow$ tractable fragments
  $\rightarrow$ semi-tractable fragments
- In case of intractability, can we approximate efficiently ?

Openings

- Are there neurosymbolic techniques for semi-supervised learning ?
  $\rightarrow$ semantic loss [Xu,2018]
  $\rightarrow$ neurosymbolic entropy regularization [Ahmed,2022]
- Can you extend beyond classification ?
  $\rightarrow$ object detection ?
  $\rightarrow$ scene graph generation ?
  $\rightarrow$ text generation ?

We currently extend to conformal prediction.

Plan

INTERPRETABLE IMAGE CLASSIFICATION THROUGH AN ARGUMENTATIVE DIALOG BETWEEN ENCODERS – ECAI, 2024

PhD thesis: Dao Thauvin

Stéphane Herbin

Wassila Ouerdane

Interpretable image classification through an argumentative dialog between encoders

Objective

Explainability

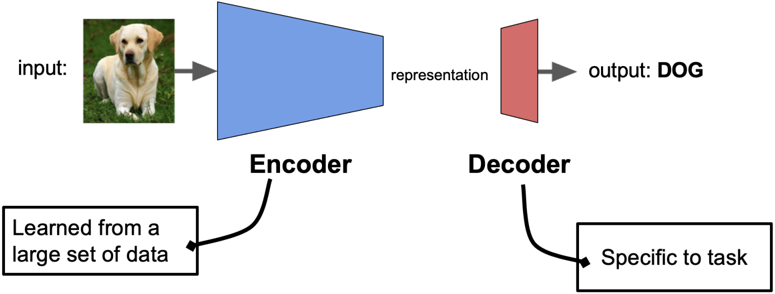
How

Human-like cognition : collective decision and argumentation

Nesy

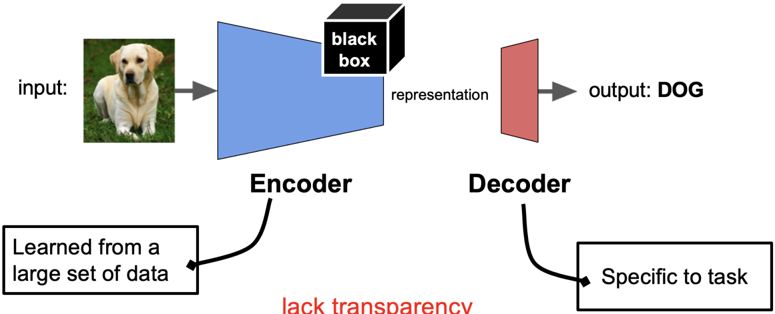
Hybridation of argumentation-based dialogue with image classification with deep models.

Image classification : state-of-the art



Encoder: DINO [Caron et al. 2021] / DINOv2 [Oquab et al. 2023] / CLIP [Radford et al. 2021]

Image classification : state-of-the art



lack transparency
=> loss of confidence

How to improve the model transparency?

Image classification with interpretability

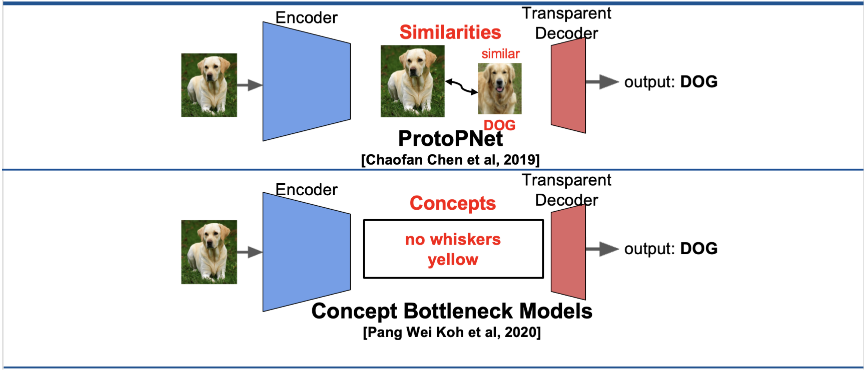
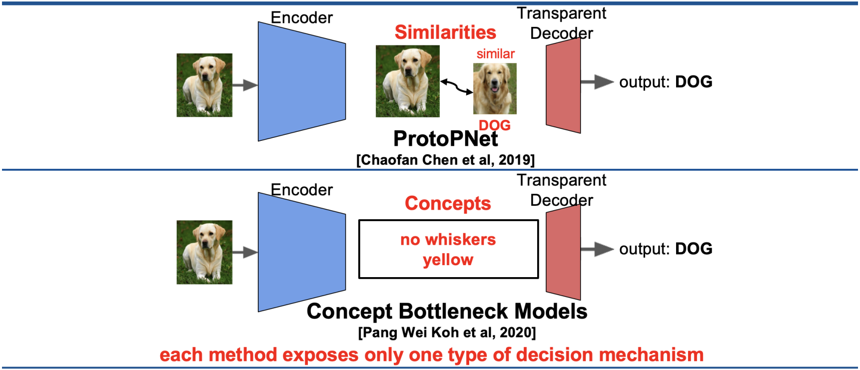
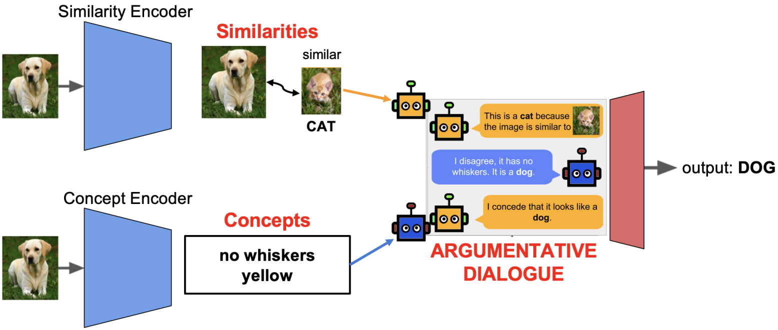


Image classification with interpretability

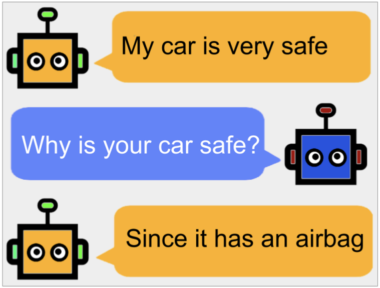


Our proposition : combine concepts and similarity



Argumentation-based Dialog [Black et al, 2021]

Argumentative Dialogue: exchanges between several agents of **arguments** (reasons) for or against some matter.



Interest: close to humans

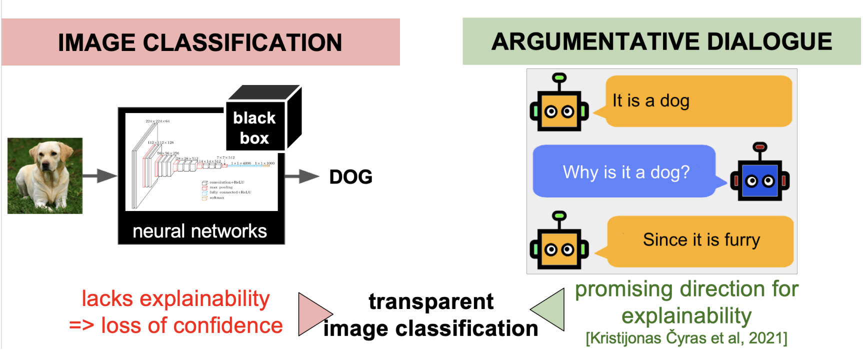
Role of argumentative dialogue

Expose Propositions & Arguments
≠
Predict the class

Black et al. Argumentation-based Dialogue. Handbook of Formal Argumentation, Volume 2, College Publications, 2021 [5]

5. `https://hal.science/hal-03429859v1`

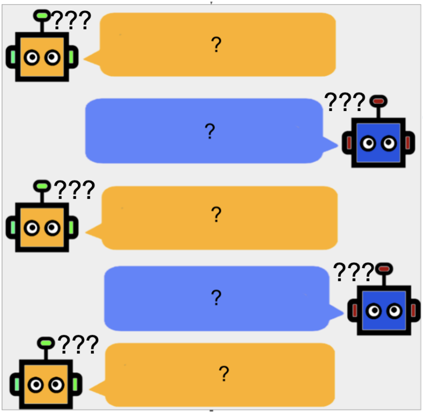
A bridge between two domains



See K Čyras, Argumentative XAI : A Survey [6]

6. https://arxiv.org/abs/2105.11266

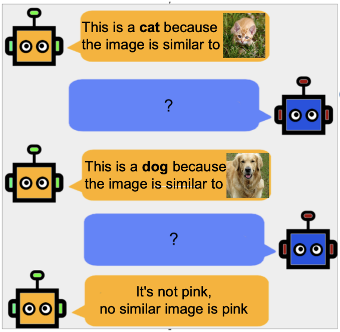
Formalize a dialog for image classification

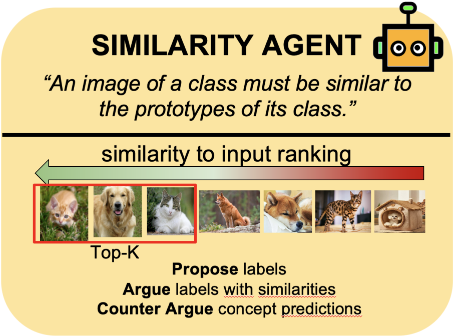


Specify:
- agents' knowledge
- agents' roles
- exchange rules

Expected Properties:
- Collaborative
- Efficient
- Interpretable
- Computable

Two role : similarity agent





Logical formalization

Dialogue

(1) **P**: *I propose the label* DOG *for* x.
(2) **A**: *Why is* x *of class* DOG?
(3) **P**: *Since* p_4 *is similar to* x *and* p_4 *is of class* DOG, *so* x *is a* DOG.

Speech Acts

PROPOSE (x_is_y)
WHY-PROPOSE (x_is_y)
ARGUE(Ψ,ϕ)
DROP-PROPOSE(x_is_y)
CONCEDE(ϕ)

Visual Information (Literals)

label ***DOG*** *for x*
x_is_DOG
p_4 is similar to x
p_4_is_sim_to_x
p_4 is of class ***DOG***
p_4_is_DOG
p_4 has the concept ***WHISKERS***
p_4_has_WHISKERS

Arguments (Logical Rules)

p_is_sim_to_x ∧ p_is_y → x_is_y
p_is_y ∧ ¬(p_has_t) ∧ x_has_t → ¬(x_is_y)

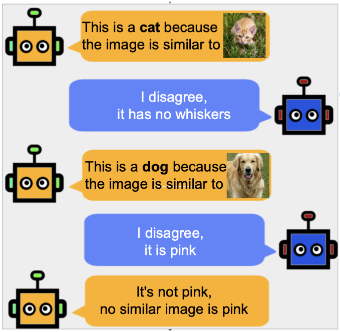
Dialogue Protocol

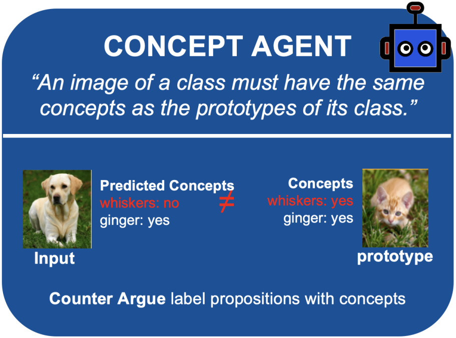
Speech acts	Attacks	Surrenders
PROPOSE (x_is_y) **(1)**	WHY-PROPOSE (x_is_y) ARGUE(Ψ, $\neg(x\_is\_y)$)	
WHY-PROPOSE (x_is_y) **(2)**	ARGUE(Ψ, x_is_y)	DROP-PROPOSE(x_is_y)
ARGUE(Ψ,ϕ) **(3)**	ARGUE(Ψ',ϕ') where $\phi' = \neg\phi$ or $\neg\phi' \in \Psi$	CONCEDE(ϕ)
DROP-PROPOSE(x_is_y)		
CONCEDE(ϕ)		

Formal representation of the dialogue

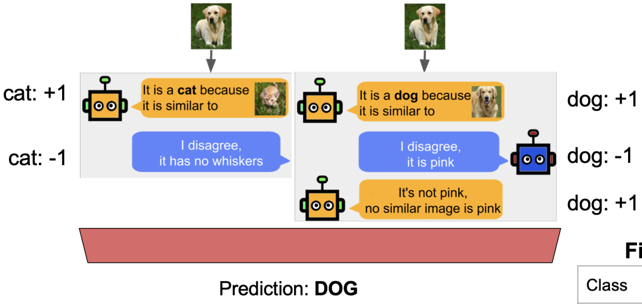
(1) Propose(x_is_DOG)
(2) Why-Propose(x_is_DOG)
(3) Argue(p_4_is_sim_to_x∧x_is_DOG→x_is_DOG, x_is_DOG)

Two role : concept agent





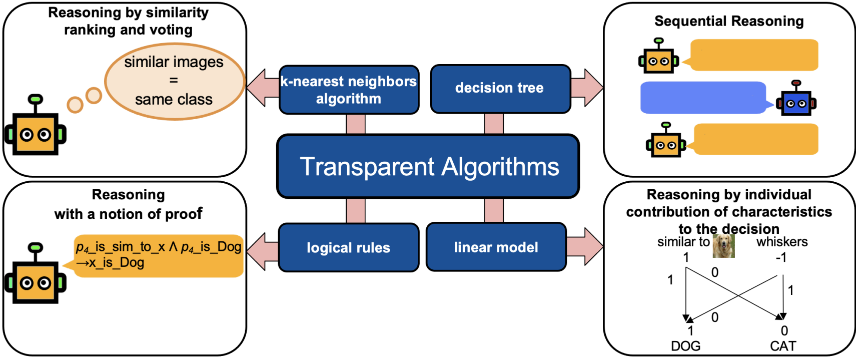
Class prediction



Final Voting Score

Class	Cat	Dog
	0	1

Results : Expressivity of our model

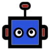


Implementation

CUB 200 dataset
[Wah et al. 2011]
number of classes: 200
number of concepts: 312

Similarity Encoder: DINO/DINOv2
[Caron et al. 2021]/[Oquab et al. 2023]

Flowers 102 dataset
[Nilsback et al. 2008]
number of classes: 200
number of concepts: 0
=> generate concepts [Han et al. 2023]



Concept Encoder: CLIP
[Radford et al. 2021]

Example of dialogue

Predicted Label: Warbling Vireo

(1) $\mathcal{P}$: I propose that x is of label Philadelphia Vireo.
(2) $\mathcal{A}$: Why x is of label Philadelphia Vireo?
(3) $\mathcal{P}$: x is of label Philadelphia Vireo because x is similar to prototype 4576, prototype 4576 is of label Philadelphia Vireo.
(4) $\mathcal{A}$: x is not of label Philadelphia Vireo because x has not the attribute yellow throat color, prototype 4576 has the attribute yellow throat color, prototype 4576 is of label Philadelphia Vireo.
(5) $\mathcal{P}$: x is of label Philadelphia Vireo because x is similar to prototype 4578, prototype 4578 is of label Philadelphia Vireo.
(6) $\mathcal{A}$: Ok, x is of label Philadelphia Vireo
(7) $\mathcal{P}$: I propose that x is of label Warbling Vireo.
(8) $\mathcal{A}$: Why x is of label Warbling Vireo?
(9) $\mathcal{P}$: x is of label Warbling Vireo because x is similar to prototype 4616, prototype 4616 is of label Warbling Vireo.
(10) $\mathcal{A}$: Ok, x is of label Warbling Vireo
(11) $\mathcal{P}$: x is of label Warbling Vireo because x is similar to prototype 4635, prototype 4635 is of label Warbling Vireo.
(12) $\mathcal{A}$: Ok, x is of label Warbling Vireo

Experimental results

Method	Accuracy CUB	Accuracy Flowers 102
K-NN (DINO)	68.72%	80.92%
Ours (CLIP+DINO)	**70.29%**	**83.67%**
K-NN (DINOv2)	86.65%	**99.67%**
Ours (CLIP+DINOv2)	**86.69%**	**99.67%**
Concept Bottleneck Models	80.1%	/
ProtoPNet	80.2%	/

Results:

- Better performance than a K-NN with the similarity encoder
- The difference of performance depends on the similarity encoder
- Better performance than classical transparent methods

Transparent & Accurate

Conclusion

An hybrid dialogue-based approach

- **Collaboration** : Combines and explicits similarities and concepts
- **Interpretable** : Makes the synthesis of the 4 classical transparent methods
- **Efficient** : Allows a more reliable classification by a collaboration between 2 agents

Future works

- Apply and adapt the method to new tasks and new datasets
- Properly evaluate its explainability
- Take advantage of its transparency to let users correct the model itself