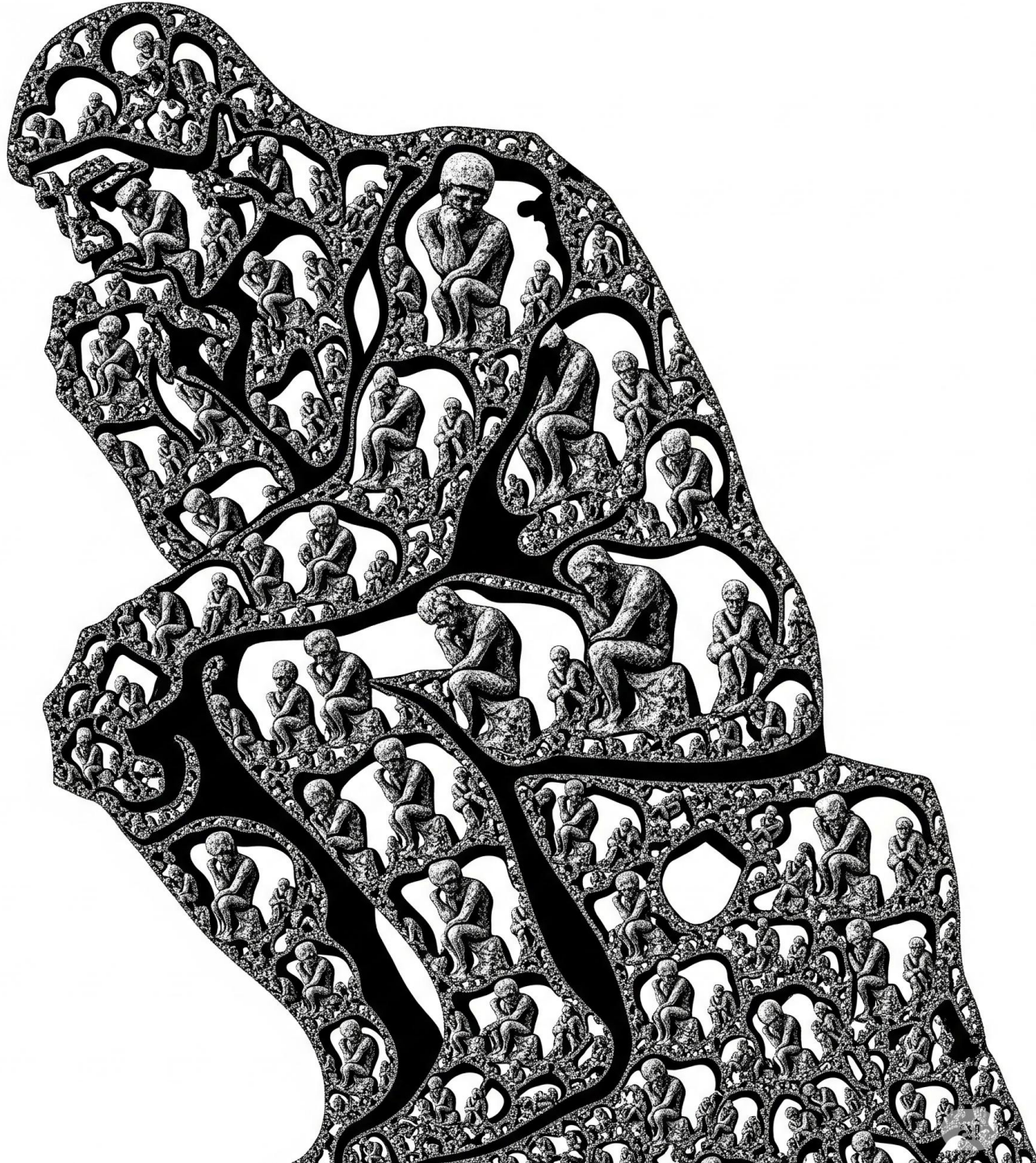
Syracuse University

Artificial Metacognition: A Semantic Approach

2026 IEEE ICSC Keynote

Paulo Shakarian
K.G. Tan Endowed Professor of AI
Director, Leibniz Lab

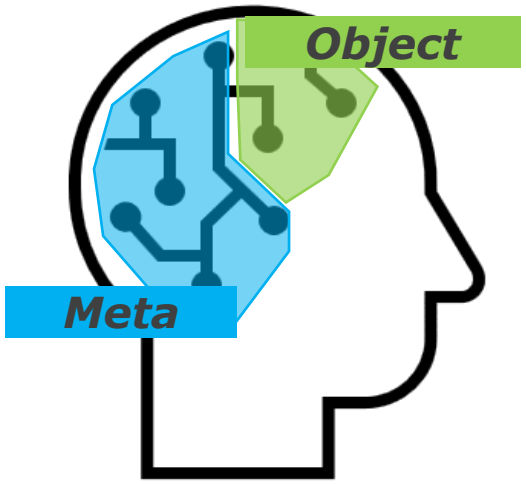
Leibniz Lab



Metacognition:

Awareness and understanding of one's own thought process.

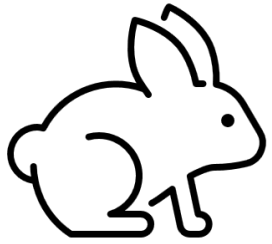
Object and Meta-level reasoning (Nelson and Narens, 1994)



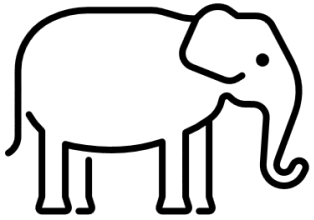
Object-Level *processes include tasks such as perception, learning, reasoning, and planning*

Meta-Level *processes monitor and assess the object-level processes*

Dual Process Theory: Thinking Fast and Thinking Slow



System 1



System 2

***System 1**: fast, rapidly interprets sensor input*

***System 2**: slow, reaches conclusions based on inference*

- A 1975 study by Wasson and Evans popularized the notion of dual process theory in psychology
- Popularized by Nobel Lauret Daniel Kahneman's 2011 book "Thinking Fast and Thinking Slow"
- But the theory was criticized by what separates the to systems – different studies identified different correlates for System 1 and System 2

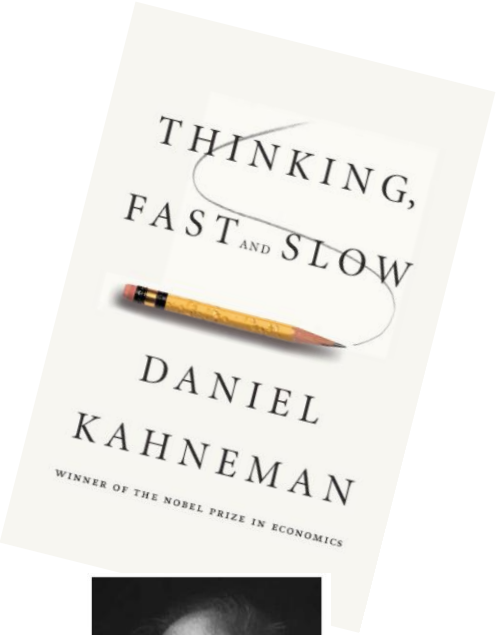
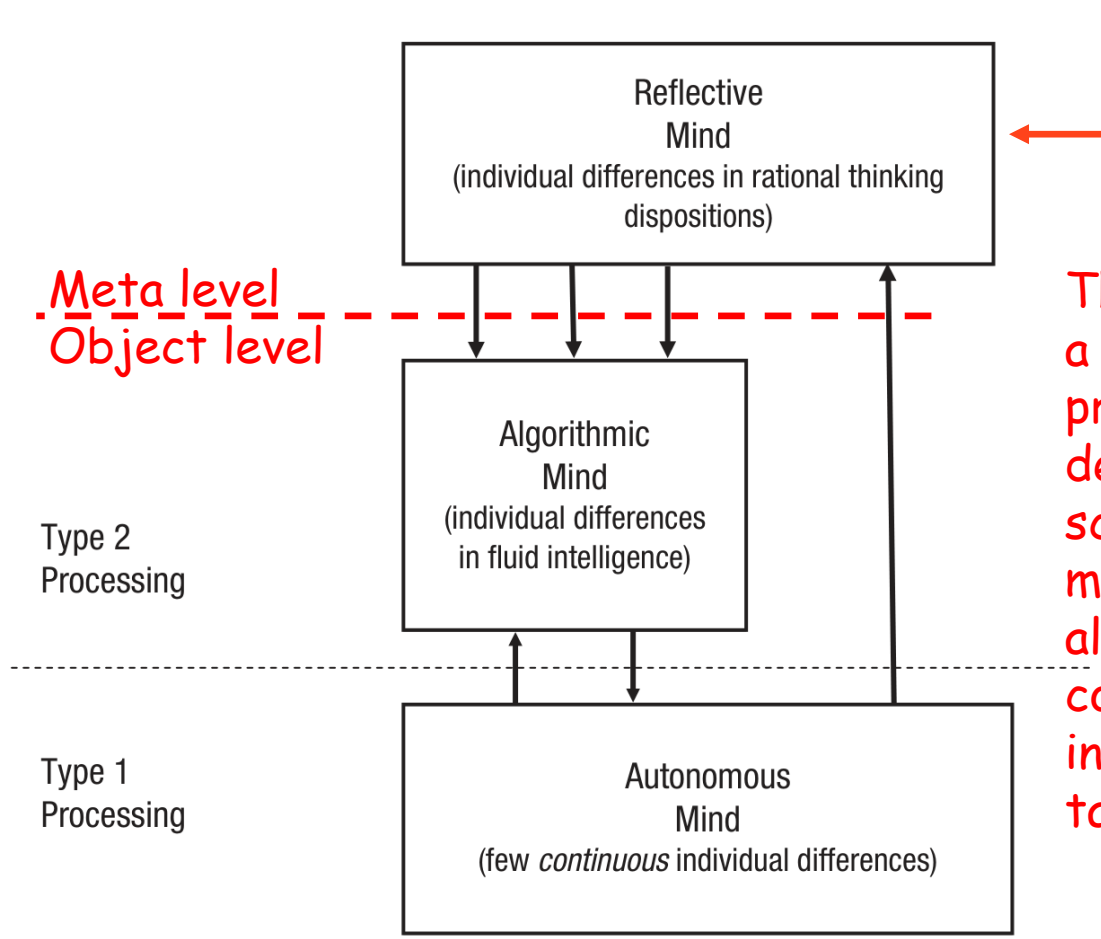


Photo from the Nobel Foundation archive.

Human Cognition: A Dual Process

An occasional "slow thinking" is used when we feel that the autonomous processes are incorrect

Humans have "fast thinking" from a collection of autonomous processes



The reflective mind is a metacognitive process. It is determining when scarce working memory should be allocated for more computationally intensive "Type 2" tasks.

Figure from Evans and Stanovich 2013

What initiates Metacognitive processes

Flavell (1979) proposed “automatic” and “deliberate” metacognitive processes:

1. **Automatic metacognition** entails the emergence of metacognitive “cues” -- heuristics that indicate provide information about the quality of the cognitive action.

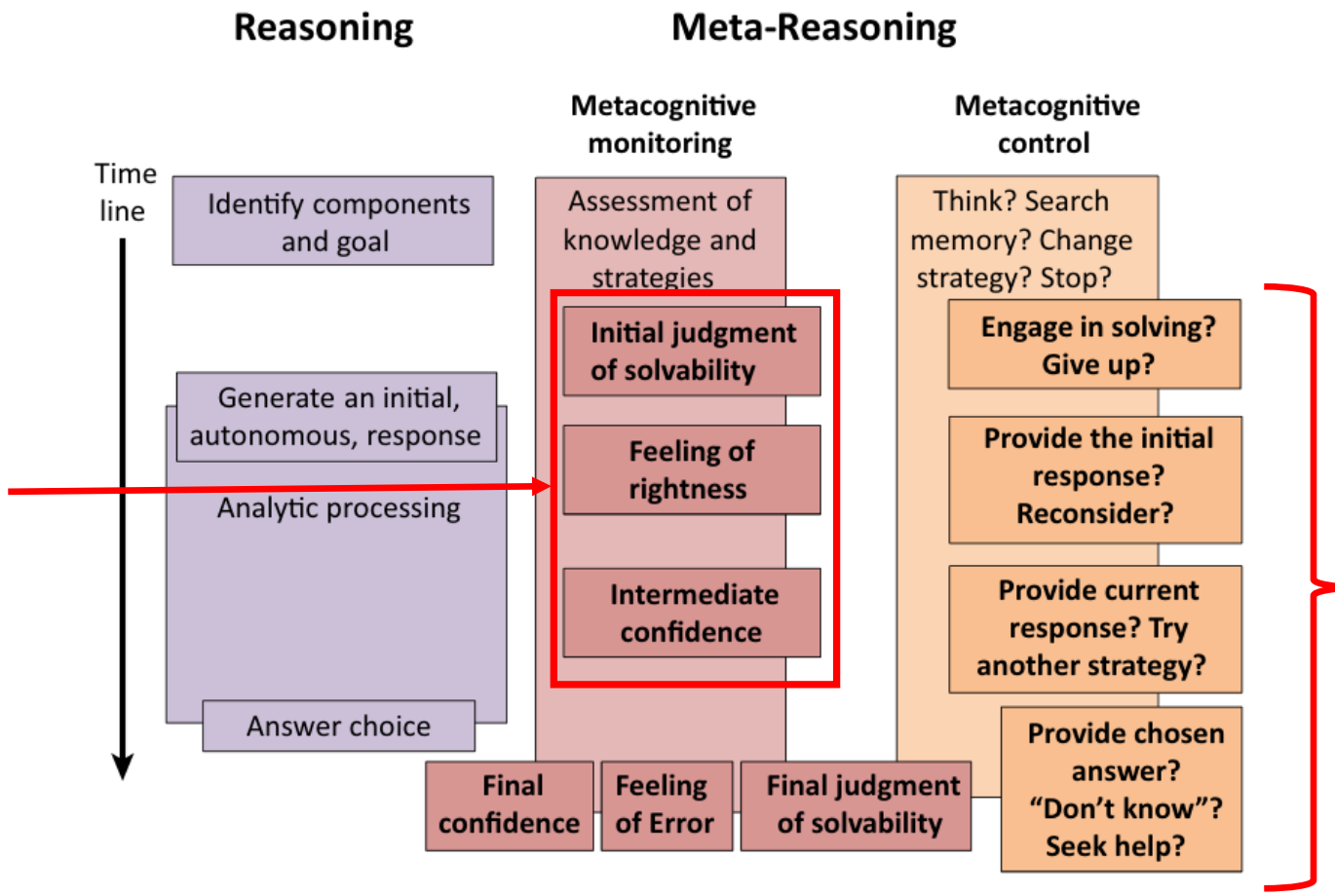
2. **Deliberate metacognition** serves various purposes with principle activities include:

- Communication of cognitive state (Shea et al. 2014)
- Seeking help (Undorf, Livneh, and Ackerman 2021)
- Regulation of time investment (Ackerman 2013; Ackerman and Undorf 2017; Toplak et al., 2014)

Figure from Evans and Stanovich 2013

Metacognitive Monitoring and Control

Humans use various "metacognitve cues" to self-assess reasoning processes



These cues allow for regulation of cognitive resources

Figure from Ackerman and Thomson 2017

Computational Criterion for Metacognition

Toward Computational Instantiation: TRAP: Criterion for Artificial Metacognition

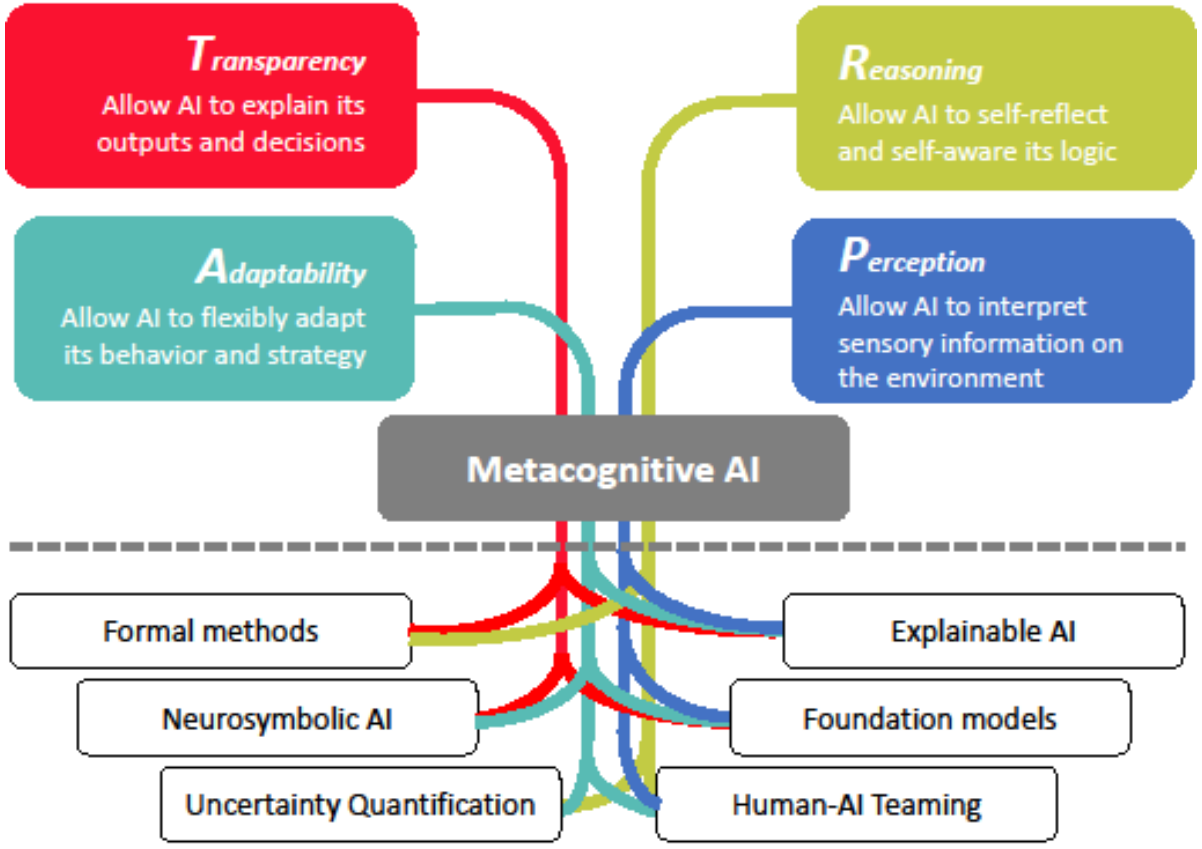
TRAP (Transparency, Reasoning, Adaptability, and Perception) form a criteria for an artificial metacognitive (AM) system.

Transparency

Adaptability

Reasoning

Perception
Can the AM interpret perceptual information to improve results



Wei et al., 2024

Mapping Computational Desiderata to Cognitive Psychology Concepts

Design considerations

- Perception
  - Metacognitive monitoring / automatic metacognition
  - Developing metacognitive cues
  - Easily computable, triggers other processes
- Reasoning
  - Metacognitive control / deliberate metacognition
  - Correct initial outputs
  - Regulate computational resources

Desired capabilities

- Adaptability
  - A capability that arises out of metacognitive control / deliberate metacognition
- Transparency
  - Communication of cognitive state

Techniques to Employ Metacognitive Monitoring

1. Detect an error state

- Train a model to detect if the model has an error
- Examples include ML introspection (Daftry et al., 2016), NASR (Cornelio et al., 2022), and abductive learning with new concepts (Huang et al., 2023)

2. Use an alternative model for the same task

- Use results of another model trained on the same task. The results of those models act as metacognitive cues (e.g., feature for an error detection model) (Lee et al., 2024)
- Recent work has looked at multiple models and does not require a "lead model" (Leiva et al., 2026)

Techniques to Employ Metacognitive Monitoring

3. Critique models

- Common technique with metacognition for LLM's (Shinn et al. 2023; Xiong et al. 2025; Yang et al. 2025)
- Recent techniques such as Critic-V (Zhang et al. 2025) were trained on degraded LLM reasoning paths
- Such approaches can also inform correction due to their natural-language syntax

4. Consistency-based approaches

- Use a verifier to determine if the results are internally-consistent
- This has been demonstrated with human-created requirements (Yang, Neary, and Topcu 2024)
- But has also been demonstrated with respect to logical rules learned from training data (Krichelli et al., 2024)

Example Metacognitive Architecutres

1. **Detect an error state**
2. **Use an alternative model for the same task**
3. **Critique models**
4. **Consistency-based approaches**

Error Detection Rules: A strategy toward a metacognitive wrapper

Can we derive rules on when to disregard a machine learning result and attempt to make a correction?

Simple example.
Assume multi-label classification problem where samples can have a subset of labels (automotive example).
$\{ford, toyota, dodge, us, japan\}$

So, for a given sample, we have a subset of labels.

$$f_{car}(\omega) = \{dodge, us\}$$

model sample returned set of labels

Example detection and correction rules and intuitions.

$$error^{car}_{toyota}(X) \leftarrow pred^{car}_{toyota}(X) \wedge cond_{us}(X)$$

Here we have a single condition $cond_{us}$, and we define it to be true for sample x when $us \in f_{car}(x)$. This is an example of using a class label from a different level of the hierarchy, as done in (Kricheli et al. 2024). Likewise, we can imagine a detection rule:

$$corr^{car}_{dodge}(X) \leftarrow cond_{us}(X) \wedge pred^{car}_{toyota}(X)$$

Here, if the condition-class pair $cond_{us}$ and toyota are true, then the sample should be re-labeled as dodge.

Shakarian et al., AAAI-MAKE 2025.

Probabilistic Interpretation

Probabilistic interpretation. Here we express key metrics as conditional probabilities. Specifically, we are interested in precision and recall change when metacognitive conditions are present.

Defining precision and recall.

$$Precision: P_\alpha = \mathbf{P}(\alpha \in gt \mid \alpha \in f_i)$$
$$Recall: R_\alpha = \mathbf{P}(\alpha \in f_i \mid \alpha \in gt)$$

Precision and recall when a condition is present.

$$Precision: P_\alpha^c = \mathbf{P}(\alpha \in gt \mid \alpha \in f_i, c \notin M)$$
$$Recall: R_\alpha^c = \mathbf{P}(\alpha \in f_i, c \notin M \mid \alpha \in gt)$$

Shakarian et al., AAAI-MAKE 2025.

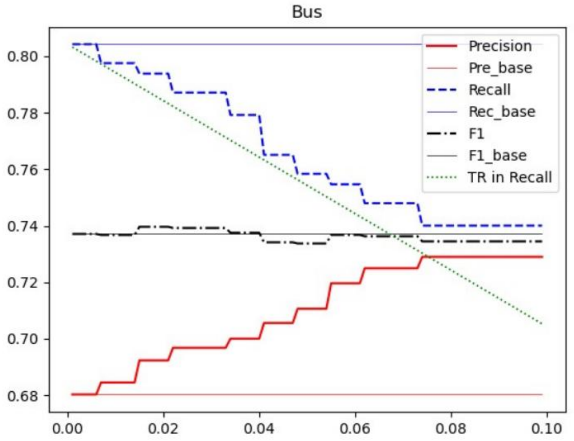
Necessary and Sufficient Condition for Error Correction

Prior Empirical Observation

Identifying conditions by maximizing the following product (support times confidence) subject to the below quantity leads to error

$$\mathbf{P}(c \in M \mid \alpha \in f_i) \times \mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M)$$

Example results for precision improvement for trajectory classification (Xi et al., '25)



Characterization of Precision

Theorem 3.1 (Metacognitive Precision Change).

$$P_\alpha^c - P_\alpha = K \times \left(\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M) - (1 - P_\alpha)\right)$$

$$where\ K = \frac{\mathbf{P}(c \in M \mid \alpha \in f_i)}{\mathbf{P}(c \notin M \mid \alpha \in f_i)}$$

- Validates empirical findings
- Proven without independence assumptions
- Gives rise to necessary and sufficient condition for precision improvement: the probability of error given the condition must be greater than the false discovery rate

$$\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M) > 1 - P_\alpha$$

- Provides for necessary and sufficient conditions for error detecting conditions (Theorem 3.2 in the paper)

Shakarian et al., AAAI-MAKE 2025.

Limits of Reclassification

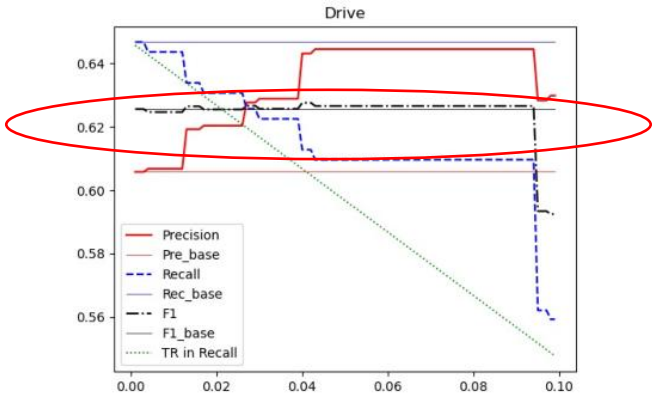
Prior Empirical Observation

We were able to obtain significantly better F1 improvement in binary classification tasks when compared to multi-class when label correction was included.

Single-class example (Lee et al. '24)

Magnesium			
Model Variant	Precision	Recall	F1
CNN (1)	0.52	0.65	0.58
RNN (4)	0.20	0.99	0.33
CNN (2)	**0.86**	0.17	0.28
CNN (1) (EDCR)	0.53 (+3.19%)	0.79 (+21.74%)	**0.64** (+10.67%)
RNN (4) (EDCR)	0.20 (+0.87%)	**1.00** (+1.43%)	0.33 (+0.96%)
CNN (2) (EDCR)	**0.86** (0.0%)	0.17 (+0.0%)	0.28 (0.0%)

Multi-class example (Xi et al. '25)



Limits of Reclassification

Theorem 4.2 (Limits of Reclassification). *If* $\mathbf{P}(\beta \in gt \mid \alpha \in f_i, c_\alpha \in M) \leq \mathbf{P}(\beta \in gt \mid \beta \in f_i)$ *then*

$$\mathbf{P}(\beta \in gt \mid \beta \in f_i) \geq \mathbf{P}(\beta \in gt \mid \beta \in f_i \vee (\alpha \in f_i, c_\alpha \in M)$$

Essentially, this is a formal argument for the following:

- A well-trained model assigns class *i*, the most probable class by training
- But the probability of *i* being correct is lower than the average precision for predictions of class *i*.
  - Which can be identified in error detection
- However, the next most probable class, *j*, is lower still, and picking this would lower overall loss.
- This is because both the model and the conditions, those samples cannot be re-assigned in a manner to improve performance.

Shakarian et al., AAAI-MAKE 2025.

Example Metacognitive Architecutres

1. **Detect an error state**

2. **Use an alternative model for the same task**

3. **Critique models**

4. **Consistency-based approaches**

Problem Setup

Multiple models

- Have error detection rules learned from training data
- No information on how the models can work together

Models are deployed (tested) in a novel environment

Shown on the left are snapshots of an aerial image dataset used to train the models.

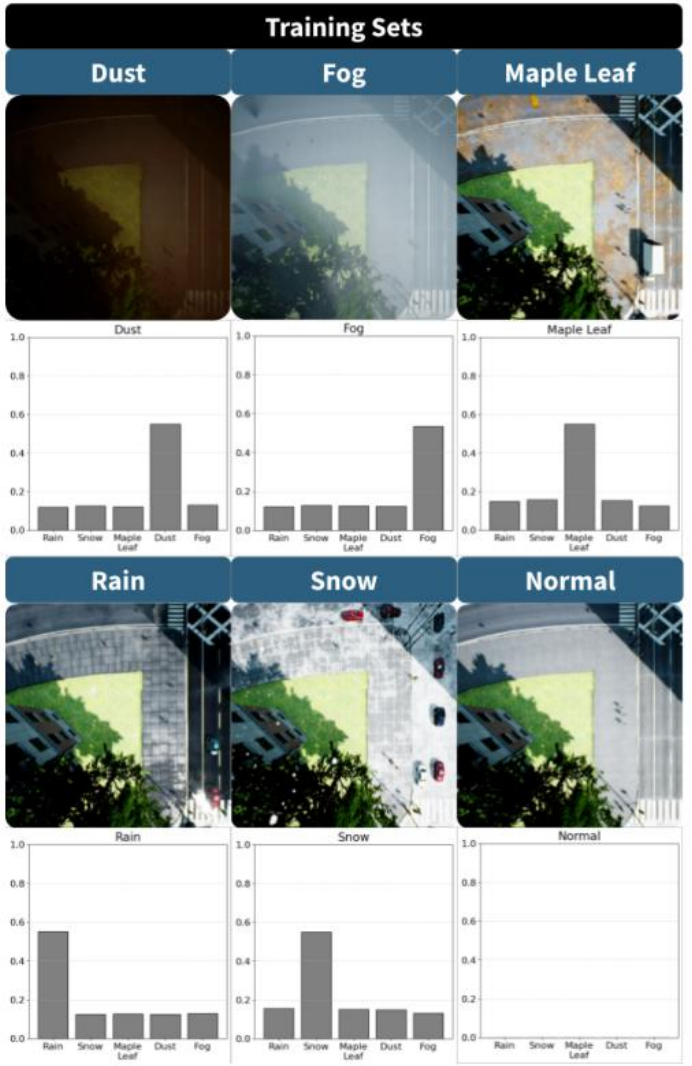


Figure from Leiva et al., AAAI 2026

Test Environments

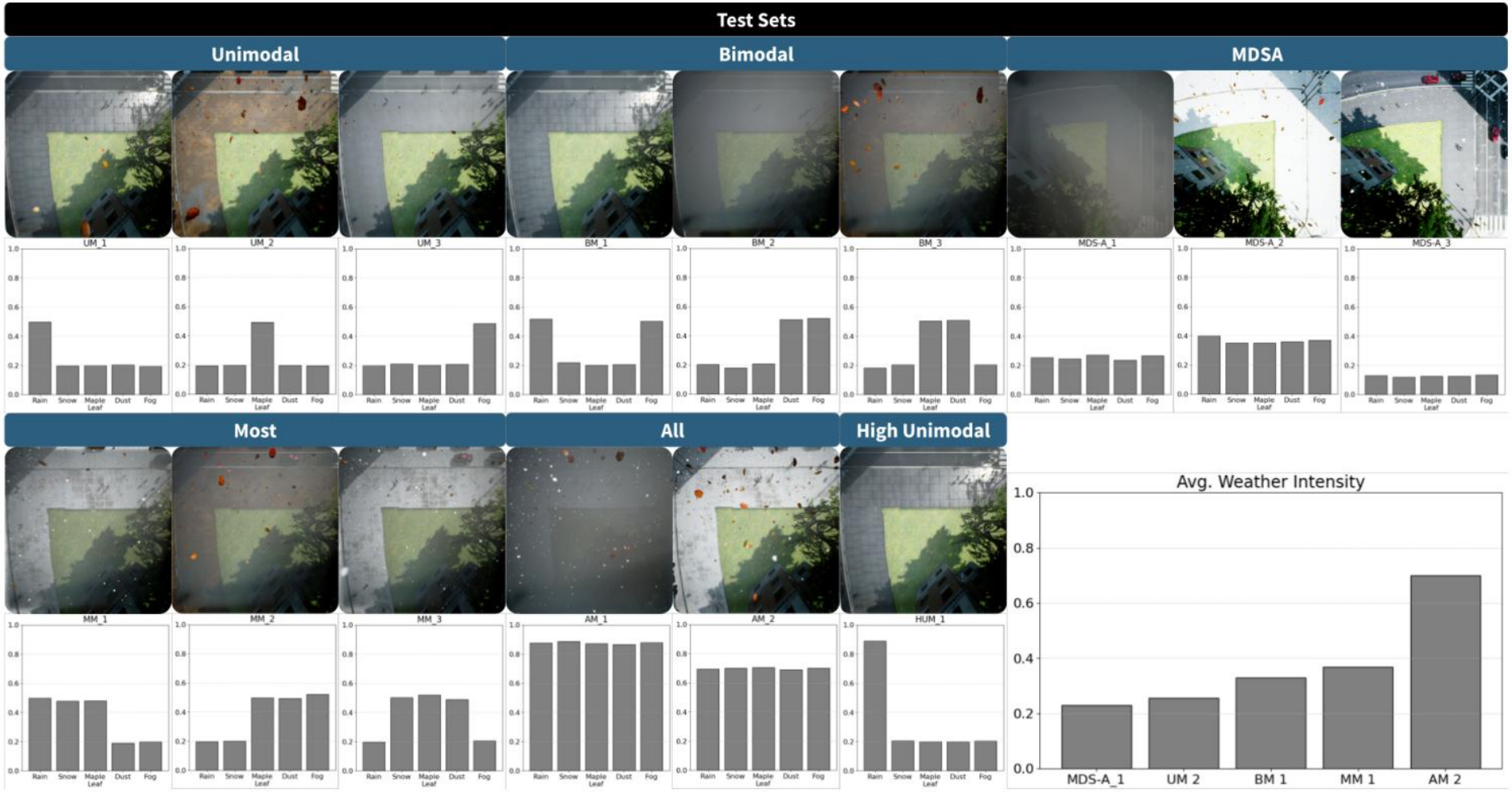


Figure from Leiva et al., AAAI 2026

Approach

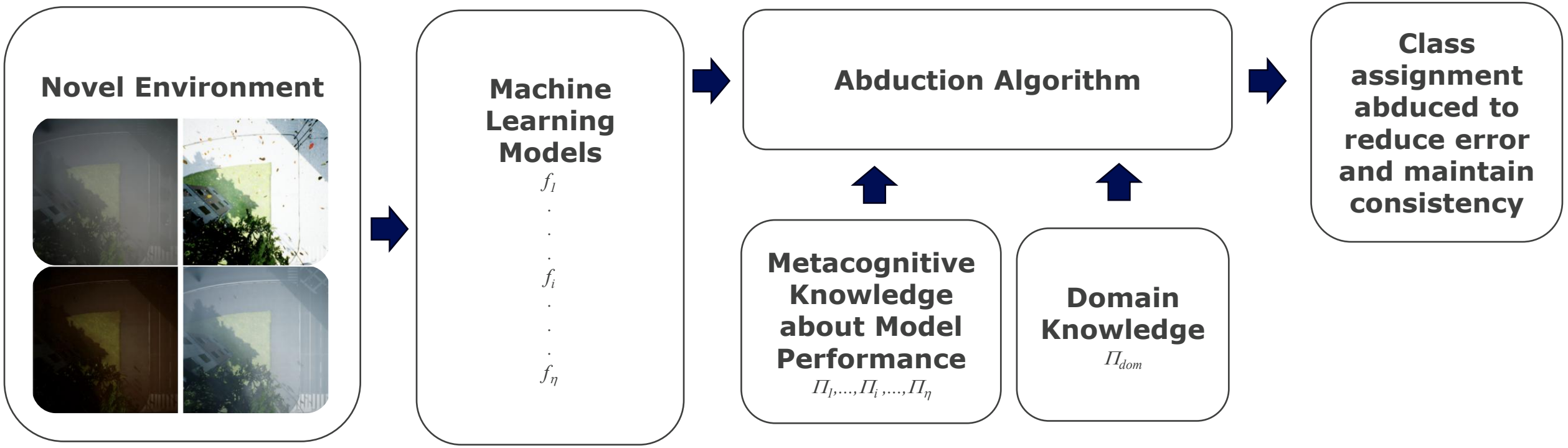


Figure from Leiva et al., AAAI 2026

Abduction Problem

- *accept(i,c)* is an atom that is true if we accept model f_i's classification of class *c*
- A set of atoms formed with predicate *accept* is a hypothesis and denoted *H*
- We use the following rule to *assign* samples to a given class based on *accept* and the error detection rules

$$assign(c, \omega) \leftarrow \neg error(i, c, \omega) \wedge (f_i(\omega) = c) \wedge accept(i, c)$$

- Pred(H) is the number of assignments that occur due to a given hypothesis
- Inc(H) is the number of inconsistencies that occur due to a given hypothesis

$$\max_{H \in \mathcal{H}} \; Pred(H)$$

subject to:

$$Inc(H) \leq \delta, \quad \delta \in [0, 1]$$

and

$$(H \cup O \cup \Pi) \setminus \Pi_{dom} \quad \text{is consistent}$$

Formalism from Leiva et al., AAAI 2026

Algorithmic Approach

EXACT APPROACH: INTEGER PROGRAM

$$\max \sum_{\omega \in \Omega} \sum_{c \in \mathcal{C}} A_{c,\omega},$$

subject to:

$$\sum_{\omega \in \Omega} \sum_{(c,c') \in IC} Con_{\omega,(c,c')} \leq \delta,$$

$$X_{\omega,f,c} \leq 1 - Elim_{f,c}$$

$$X_{\omega,f,c} \cdot pred_{f,c,\omega} \leq A_{c,\omega}$$

$$A_{c,\omega} \leq \sum_{f} X_{\omega,f,c} \cdot pred_{f,c,\omega}$$

$$A_{c,\omega} + A_{c',\omega} - 1 \leq Con_{\omega,(c,c')}$$

$$\sum_{c \in \mathcal{C}} A_{c,\omega} \geq 1$$

$$\sum_{\omega \in \Omega} \sum_{(c,c') \in IC} Con_{\omega,(c,c')} \leq \delta$$

APPROXIMAT APPROACH: HEURISTIC SEARCH

Algorithm 1: Heuristic Search (HS) for Prediction Optimization

Input:
  P_{raw} (Set of all raw prediction tuples (o, l, f, c))
  δ (Maximum allowed inconsistency for S_{final})
  E_{set} (Set of EDR ϵ thresholds to evaluate)
  {Implicit: Sets $\mathcal{F}$ (models), $\mathcal{C}$ (classes); Functions $GetFilteredPreds(f, c, \epsilon, P_{raw})$ and $CalcIncon(S)$.}
Output: S_{final} (Optimized set of prediction tuples (o, l))
$S_{final} \leftarrow \emptyset$
for each model $f \in \mathcal{F}$ and class $c \in \mathcal{C}$ **do**
  $P_{best\_add} \leftarrow \emptyset$ {Best predictions from current (f, c) to add}
  $n_{current\_max} \leftarrow |S_{final}|$ {Max size of $S_{final} \cup P_{new}$}
  for each $\epsilon \in E_{set}$ **do**
    $P_{new} \leftarrow GetFilteredPreds(f, c, \epsilon, P_{raw})$
    $S_{cand} \leftarrow S_{final} \cup P_{new}$
    if $CalcIncon(S_{cand}) \leq \delta$ **and** $|S_{cand}| > n_{current\_max}$ **then**
      $P_{best\_add} \leftarrow P_{new}$
      $n_{current\_max} \leftarrow |S_{cand}|$
    end if
  end for
  if $P_{best\_add} \neq \emptyset$ **then**
    $S_{final} \leftarrow S_{final} \cup P_{best\_add}$
  end if
end for
return S_{final}

Also, a confidence-based tie breaker heuristic was employed in both approaches.

Formalism from Leiva et al., AAAI 2026

Algorithmic Approach

Test Set	Best		Avg.		MV		IP+TB		HS+TB	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
MDS-A_1	0.57	0.40	0.52	0.36	0.28	0.34	**0.58**	**0.41**	**0.58**	**0.41**
MDS-A_2	0.33	0.20	0.29	0.17	0.26	**0.22**	**0.37**	**0.22**	0.32	0.19
MDS-A_3	0.54	0.37	0.49	0.33	0.39	0.29	**0.56**	**0.39**	0.55	0.38
UM_1	0.54	0.37	0.47	0.31	0.26	0.23	**0.64**	**0.47**	0.61	0.44
UM_2	0.56	0.38	0.46	0.31	0.25	0.22	**0.64**	**0.47**	0.61	0.44
UM_3	0.54	0.37	0.43	0.28	0.22	0.19	**0.63**	**0.46**	0.59	0.42
BM_1	0.42	0.27	0.33	0.20	0.19	0.16	**0.45**	**0.29**	0.39	0.24
BM_2	0.33	0.20	0.25	0.15	0.14	0.12	**0.37**	**0.23**	0.36	0.22
BM_3	0.37	0.23	0.31	0.19	0.18	0.16	**0.43**	**0.27**	0.40	0.25
MM_1	0.46	0.30	0.40	0.25	0.22	0.21	**0.51**	**0.34**	0.46	0.30
MM_2	0.32	0.19	0.24	0.14	0.13	0.10	**0.36**	**0.22**	0.29	0.17
MM_3	0.41	0.26	0.35	0.22	0.18	0.16	**0.46**	**0.30**	0.39	0.24
AM_1	0.18	0.10	0.12	0.07	0.05	0.04	**0.21**	**0.11**	0.18	0.10
AM_2	0.23	0.13	0.18	0.10	0.07	0.06	**0.28**	**0.16**	0.23	0.13
HUM_1	0.45	0.29	0.40	0.25	0.18	0.17	**0.57**	**0.40**	0.55	0.38

Test Set	IP (No TB)		HS (No TB)	
	F1 (% Diff)	Acc (% Diff)	F1 (% Diff)	Acc (% Diff)
MDS-A_1	0.58 (0.0)	0.41 (0.0)	0.52 (-10.3%)	0.35 (-14.6%)
MDS-A_2	0.37 (0.0)	0.22 (0.0)	0.27 (-15.6%)	0.16 (-16.7%)
MDS-A_3	0.56 (0.0)	0.39 (0.0)	0.49 (-10.9%)	0.32 (-15.8%)
UM_1	0.64 (0.0)	0.47 (0.0)	0.53 (-13.1%)	0.36 (-18.2%)
UM_2	0.64 (0.0)	0.47 (0.0)	0.52 (-14.1%)	0.35 (-18.8%)
UM_3	0.63 (0.0)	0.46 (0.0)	0.52 (-11.9%)	0.35 (-16.7%)
BM_1	0.45 (0.0)	0.29 (0.0)	0.34 (-11.1%)	0.20 (-16.7%)
BM_2	0.37 (0.0)	0.23 (0.0)	0.31 (-13.5%)	0.19 (-13.6%)
BM_3	0.43 (0.0)	0.27 (0.0)	0.34 (-15.0%)	0.20 (-20.0%)
MM_1	0.51 (0.0)	0.34 (0.0)	0.38 (-15.7%)	0.24 (-20.0%)
MM_2	0.36 (0.0)	0.22 (0.0)	0.25 (-13.8%)	0.14 (-17.6%)
MM_3	0.46 (0.0)	0.30 (0.0)	0.33 (-15.4%)	0.20 (-16.7%)
AM_1	0.21 (0.0)	0.11 (0.0)	0.15 (-16.7%)	0.08 (-20.0%)
AM_2	0.28 (0.0)	0.16 (0.0)	0.19 (-17.4%)	0.11 (-15.4%)
HUM_1	0.57 (0.0)	0.40 (0.0)	0.48 (-12.7%)	0.32 (-15.8%)

Consistently the best performing approach.

Table from Leiva et al., AAAI 2026

Example Metacognitive Architecutres

1. **Detect an error state**

2. **Use an alternative model for the same task**

3. **Critique models**

4. **Consistency-based approaches**

Example Problems

Geometric

Question: The second angle (A_2) of a triangle is double the first (A_1). The third angle (A_3) is 40° less than the first (A_1). Find the three angles.
Equations:
A_2 = 2 * A_1
A_3 = A_1 - 40
A_1 + A_2 + A_3 = 180

Age

Question: Dad is 25 years elder than his 10 year-old son. Mom is 3 years younger than Dad . What age is Mom?
Equations:
age_son = 10
age_dad - age_son = 25
age_dad - age_mom = 3

Numeric

Question: The sum of three consecutive even integers is 246. What are the integers?
Equations:
b = a + 2
c = b + 2
a + b + c = 246

Rate: work and time

Question: Karl can clean a room in 4 hours. His son Becky can clean it in 12 hours. How long would it take if they clean it together?
Equations:
rate_karl * 4 = 1
rate_becky * 12 = 1
(rate_karl + rate_becky) * time_together = 1

Chemistry

Question: Sally and Terry blended a coffee mix that sells for $2.50 by mixing two types of coffee. If they used 40 mL of a coffee that costs $3.00, how much of another coffee costing $1.50 did they mix with the first?
Equations:
price_mix = 2.5
price_first = 3
price_another = 1.5
amount_first = 40
amount_mix = amount_first + amount_another
amount_mix * price_mix = amount_first * price_first + amount_another + price_another

Rate: speed, distance and time

Question: Terry leaves his house riding a bike at 20 km/h. Sally leaves 6 h later on a scooter to catch up with him travelling at 80 km/h. How long will it take her to catch up with him?
Equations:
speed_terry = 20
speed_sally = 80
time_shally = time_terry - 6
time_shally * speed_sally = time_terry * speed_terry

Monetary

Question: Doug and Becky sold 41 tickets for an event. Tickets for children cost \$1.50 and tickets for adults cost $2.00. Total receipts for the event were $73.50. How many of each type of ticket was sold?
Equations:
ticket_adults + ticket_children = 41
price_children = 1.5
price_adults = 2
price_children * ticket_children + price_adults * ticket_adults = 73.5

Figures from Yang et al., AAAI-FS 2025

EDCIM: Rule Learning

Algorithm 1: DetRuleLearn (Xi et al. 2023)

Input: Recall reduction threshold ε, Condition set C
Output: Subset of conditions C'
$C' := \emptyset$
$C^* := \{c \in C \; s.t. \; NEG_{\{c\}} \leq \varepsilon \cdot N \}$
while $C^* \neq \emptyset$ **do**
  $c_{best} = \arg\max_{c \in C^*} POS_{C' \cup \{c\}}$
  Add c_{best} to DC_i
  $C^* := \{c \in C \setminus C' \; s.t. \; NEG_{C' \cup \{c\}} \leq \varepsilon \cdot N \}$
end while
return C'

Train/Test Split	ACC Gain
0.1/0.9	0.115
0.2/0.8	0.111
0.3/0.7	0.111
0.4/0.6	0.109
0.5/0.5	0.106
0.6/0.4	0.108
0.7/0.3	0.104
0.8/0.2	0.104
0.9/0.1	0.104

- Input includes candidate conditions
- Algorithm finds conditions that lead to error correction

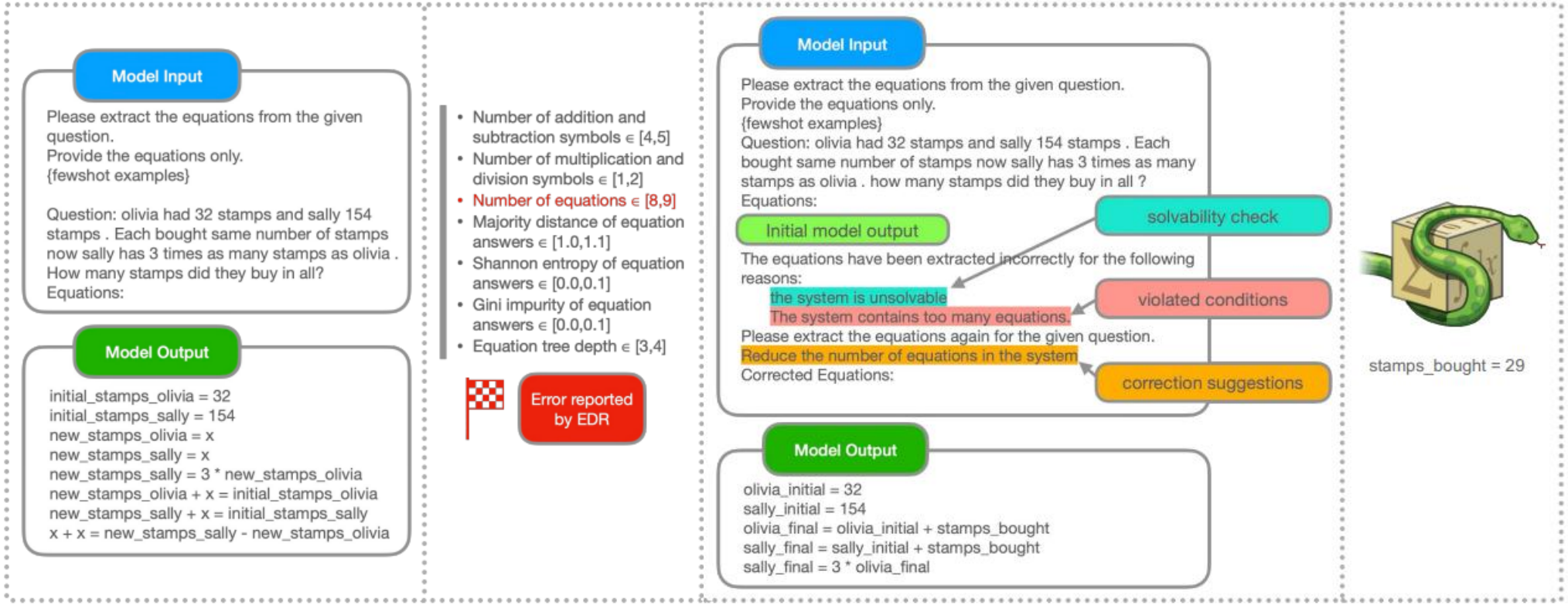
Figures from Yang et al., AAAI-FS 2025

Example Critic Model: EDCIM (Yang et al., 2025)

Problem: convert a math word problem into formal equations via LLM

Metacognitive cues suggest possible errors

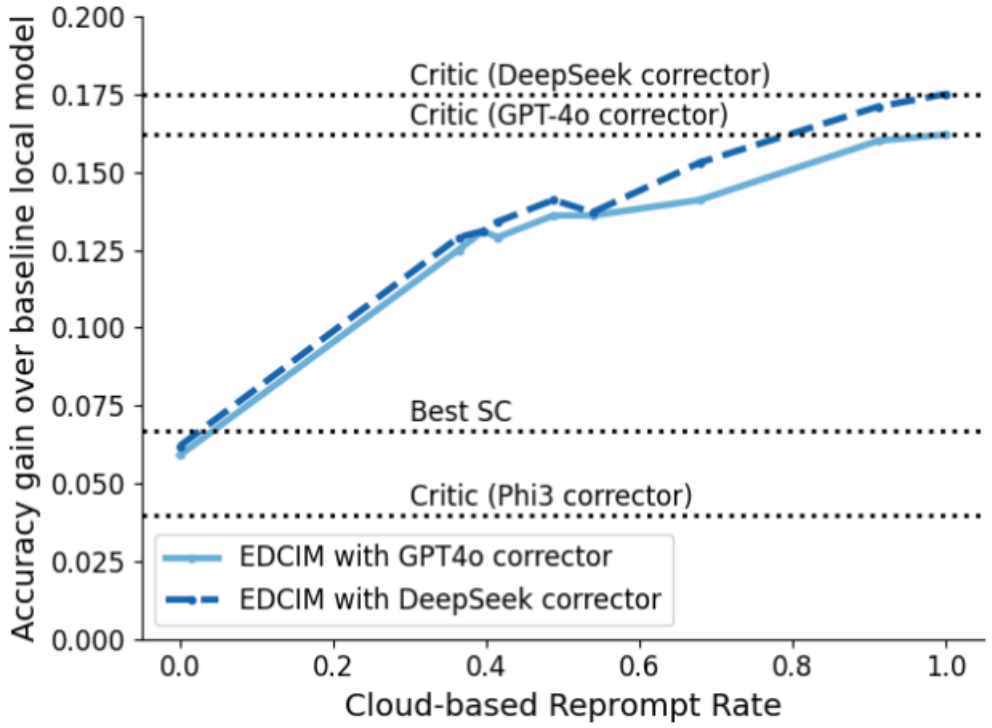
Metacognitive control uses the cues to call a more powerful LLM to improve performance



Figures from Yang et al., AAAI-FS 2025

From Metacognitive Rules to Corrective Suggestions

This, in turn, allows us to trade-off performance for queries to a more powerful model.



Type	Triggered Conditions	Correction Suggestions
Low	The system may be under-specified with too few equations	Add more equations to fully describe the problem
High	The system contains too many equations	Reduce the number of equations in the system
Low	Too few variables may miss important aspects	Consider introducing additional variables if needed
High	Too many variables were used	Use fewer variables to simplify the equations
Low	Very few constants might miss numeric details	Include more relevant numerical values
High	Too many constants were used	Use fewer numeric constants
Low	Lack of addition operations might indicate under-formed expressions	Use addition operations where needed to complete relationships
High	Too many addition operations were used	Simplify the equations by reducing additions
Low	Too few multiplications might under-represent relationships	Include multiplication to model proportional or product-based relations
High	Too many multiplication operations were used	Reduce the number of components in each equation
Low	Shallow equation depth might miss logical structure	Use more structured nesting to reflect problem hierarchy
High	Equations are deeply nested	Simplify by reducing nesting in expressions
Low	Equations are overly simple	Add more structure to better represent the problem
High	Equations are structurally complex with many elements	Reduce the number of components in each equation
Low	Too few leaf nodes may indicate underdeveloped equations	Use more complete expressions with relevant terms
High	Too many terminal nodes in expression trees	Reduce terminal terms for clarity
Low	The responses are overly uniform	Encourage more variation to explore diverse interpretations
High	The responses are highly diverse	Focus on extracting more consistent equations
Low	Very low variation detected	Consider encouraging alternative formulations
High	There is significant variation among responses	Promote more consistent equation structures
Low	Strong consensus detected Still, double-check for correctness	Still, double-check for correctness despite agreement
High	The majority answer is not clearly supported	Refine equations to better align with consensus

Figures from Yang et al., AAAI-FS 2025

Ablations

Method	Answer Generator	Error Corrector	DRAW-1k ACC	DRAW-1k # calls per sample local	DRAW-1k # calls per sample cloud	GSM-8k ACC	GSM-8k # calls per sample local	GSM-8k # calls per sample cloud
LLM only	GPT4o	-	91.6	-	1	84.4	-	1
LLM only	DeepSeek	-	92.4	-	1	86.2	-	1
LLM only	Phi3	-	75.2	1	-	52.2	1	-
SC	Phi3	-	78.8	10	-	60.4	10	-
SC+Solv.	Phi3	-	81.9	10	-	61.7	10	-
CRITIC	Phi3	Phi3	79.2	2	-	64	2	-
	Phi3	GPT4o	91.4	1	1	83.8	1	1
	Phi3	DeepSeek	92.7	1	1	84.6	1	1
EDCIM	Phi3	Phi3	78.8	10.36	-	66.2	10.43	-
	Phi3	GPT4o	85.7	10	0.36	74.4	10	0.43
	Phi3	DeepSeek	87.8	10	0.36	75.0	10	0.43

Figures from Yang et al., AAAI-FS 2025

Example Metacognitive Architecutres

1. **Detect an error state**

2. **Use an alternative model for the same task**

3. **Critique models**

4. **Consistency-based approaches**

Error Detection and Constraint Recovery in Hierarchical Multi-Label Classification

Kricheli et al., 2024 (CIKM)

Logical Framework

Key idea: learn rules to identify conditions when an image classifier would produce an erroneous classification.

$$error_y(X) \leftarrow assign_y(X) \wedge \bigvee_{cond \in DC_y} cond(X).$$

If sample *X* is assigned class *y* and one of several conditions in set DC_y is true for sample *X*, then the model produced an error.

In hierarchical multi-class problem, we can use the complementary class as a condition.

This can also allow us to recover constraints between classes.

Detection

Combinatorial algorithm allows for the learning of the set of a conditions for each class by maximizing the F1 of the error.

Algorithm 1 RatioDetRuleLearn

Require: Class $y \in \mathcal{Y}_g$ of a granularity g and its per-granularity condition set C_g

Ensure: Non-empty subset of conditions $\emptyset \subset \hat{DC}_y \subseteq C_g$

$DC_y^0 \leftarrow \emptyset, C_y \leftarrow C_g, i \leftarrow 0$

while $C_y \neq \emptyset$ **do**

$c_{best} \in \text{argmin}_{c \in C_y} \frac{BOD^{\mathcal{T}}_{DC_y^i \cup \{c\}} - BOD^{\mathcal{T}}_{DC_y^i}}{POS^{\mathcal{T}}_{DC_y^i \cup \{c\}} - POS^{\mathcal{T}}_{DC_y^i}}$

$DC_y^{i+1} \leftarrow DC_y^i \cup \{c_{best}\}$

$C_y \leftarrow \left\{ c \in C_y \,\middle|\, POS^{\mathcal{T}}_{DC_y^{i+1} \cup \{c\}} > POS^{\mathcal{T}}_{DC_y^{i+1}} \right\}$

$i \leftarrow i + 1$

end while

$\hat{i} \in \text{argmin}_i \frac{BOD^{\mathcal{T}}_{DC_y^i} + FP_y^{\mathcal{T}}}{POS^{\mathcal{T}}_{DC_y^i}}$

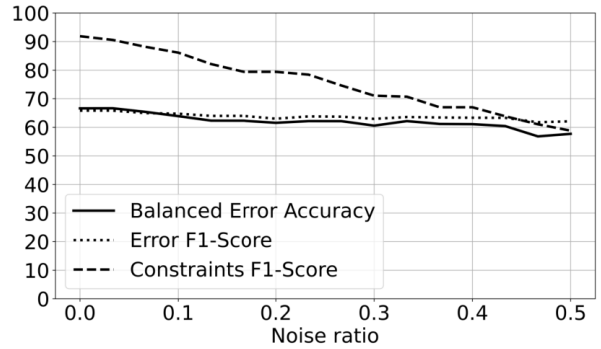
return $\hat{DC}_y \leftarrow DC_y^{\hat{i}}$

Accurate Detection (Vision)

Dataset	Method	Balanced Error Acc.	Error f1
Military Vehicles	Binary NN	80.10%	80.18%
	DetRuleLearn [32]	83.45%	82.62%
	f-EDR (ours)	**84.08%**	**83.17%**
ImageNet50	Binary NN	72.85%	68.96%
	DetRuleLearn [32]	80.92%	72.78%
	f-EDR (ours)	**84.26%**	**77.78%**
OpenImage36	Binary NN	64.80%	63.65%
	DetRuleLearn [32]	59.87%	46.46%
	f-EDR (ours)	**66.63%**	**65.83%**

Correction

Error detection rules reliably recover relationships among classes (OpenImage results shown).



We could use this error information to retrain the model and improve performance.

Dataset	Method	Fine-Grain Acc.	Coarse-Grain Acc.	Inconsistency
OpenImage36	VIT b_16	57.68%	90.15%	3.02% (362/12002)
	f-EDR + LTN (ours)	**60.11%**	**91.21%**	**1.73%** (208/12002)

Avenues for Inquiry and Closing Comments

Areas for Further Iqnuiry

Metacognitive Control

- Can we regulate the use of computational resources in a non ad-hoc manner?
- Becoming increasingly important with massive requirements for power and compute
- Techniques like EDCIM (Yang et al., 2025) have shown this is possible, but it is mainly a side-effect of correction as opposed to reasoning about available compute

Areas for Further Iqnuiry

Artificial Metacognition to Mimic Human Metacognitive Processes

- Recently proposed extension to the common model of cognition (Laird et al., 2025)
- Proposed extensions for ACT-R have also been recently proposed (Lebiere et al., 2025)
- Hyperdimensional computing, a brain-inspired paradigm has also been shown to provide evidence of metacognitive abilities.

Hyperdimensional vector "gluing" was shown to be able to combine hypervectors based on metacognitive error information.

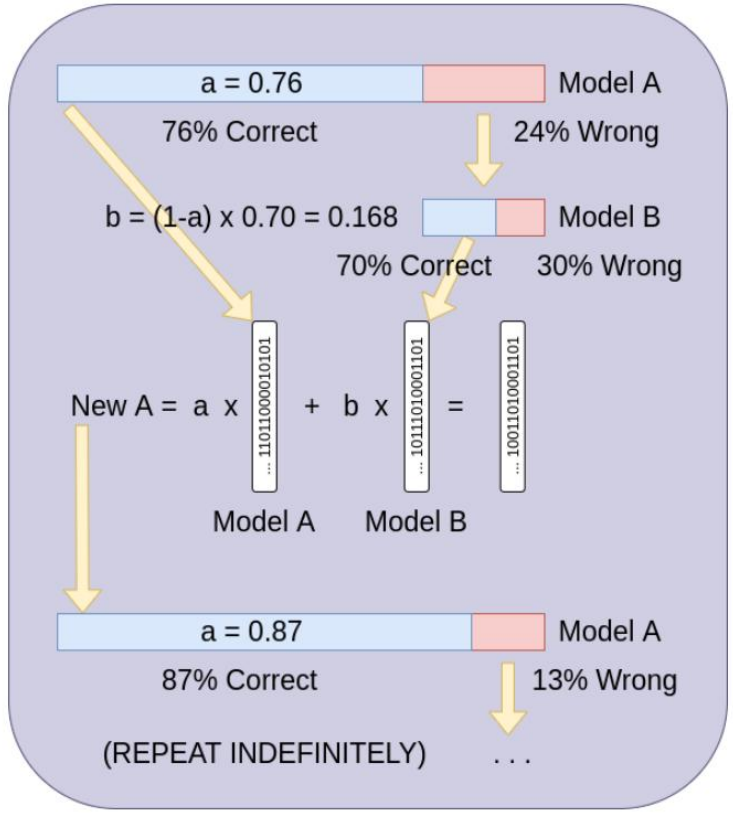


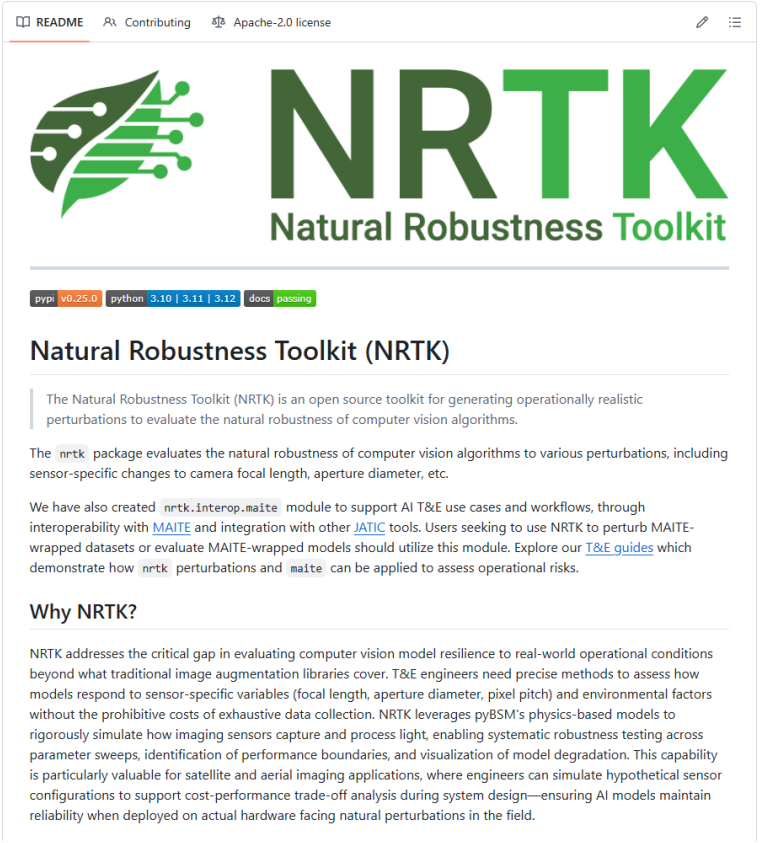
Figure from Sutor et al., IJCNN 2022.

Areas for Further Iqnuiry

Need for Datasets/Benchmarks

- Benchmarks for Artifical Metacognition are nascent
  - Benchmark: Multiple Distribution Shift – Aerial (MDS-A) (Ngu et al. 2025)
  - Toolkit for building benchmarks: Natural Robustness Toolkit (NRTK) (Kitware 2025)
- Evaluation is also a challenge (Lanus and Freeman 2025)

KitWare's NRTK can lay the potential groundwork for improved evaluation of artificial metacognition.



Past Metacognition Workshops

METACOG-23, Nov. 2023 @ Scottsdale, AZ
ARO-Sponsored
25 interdisciplinary participants representing
16 universities and companies

METACOG-23 (Nov. 2023)

This event was sponsored by the Army Research Office held on Nov. 13-15, 2023 in Scottsdale, AZ.

As artificial intelligence become more prevalent in military systems, improved characterization of such systems will, in-turn, become important to ensure that such systems are safe and reliable in supporting the warfighter. However, while AI systems, often using supervised machine learning or reinforcement learning, have provided excellent results for a variety of applications, the reasons behind their failure modes or anomalous behavior are generally not well understood. The idea of metacognition, reasoning about an AI system itself, is a key avenue to understanding the behavior and performance of machine learning systems. Recently, a variety of methodologies have been explored in the literature, which including stress testing of robotic systems [1], model introspection [2], model certification [3], and performance prediction [4]. Moreover, researchers across multiple disciplines including computer science, control theory, mechanical engineering, human factors, and business schools have explored these problems from different angles. The objectives of the workshop are as follows:

- Create a taxonomy of various approaches to metacognition of AI systems
- Understand the requirements for various metacognitive approaches
- Summarize recent results obtained in the study of AI metacognition
- Enumerate current applications for which AI metacognitive techniques have been applied
- Understand the relationship between AI metacognition and human operators

Specific topics to be covered include, but are not limited to:

- Explainable performance prediction of black-box AI systems
- Stress testing of reinforcement learning systems
- How can metacognition be used to increase trust in AI systems by the operator
- Applications of AI metacognition to robotic and vision systems

Christian Lebiere	Carnegie Mellon University	An architectural approach to metacognition
Sergei Nirenburg	Rensselaer Polytechnic Institute	Mutual Trust in Human-AI Teams Relies on Metacognition
Hua Wei	Arizona State University	Trustworthy Decision Making in the Real World through Uncertainty Reasoning
Ufuk Topcu	University of Texas	Multi-Modal, Pre-Trained Models in Verifiable Sequential Decision-Making
Visar Berisha	Arizona State University	A Theoretically-Grounded Framework for Assured ML in High-Stakes Domains
Chandan Reddy	Virginia Tech	Bridging Symbolic and Numeric Paradigms: Unified Neuro-symbolic Models for Mathematical Understanding and Generation
Paulo Shakarian	Arizona State University	Metacognitive AI through Error Detection and Correction Rules

METACOG-25, May 2025 @ Alexandria, VA (SDM)
SSCI-Sponsored,
18 interdisciplinary participants representing
15 universities and companies

Home / METACOG-25

METACOG-25

The Second Workshop on Metacognitive Prediction of AI Behavior

Second Workshop on Metacognitive Prediction of AI Behavior

Held at SDM-2025: https://www.siam.org/conferences-events/siam-conferences/sdm25/

Video release form: https://neurosymbolic.asu.edu/wp-content/uploads/sites/28/2025/03/Video-Consent-and-Release-Form.pdf

Date: May 1, 2025 10:00am-3:30pm
Location: Alexandria, VA (The Westin Alexandria Old Town Hotel, room Edison C)
Part of SIAM Data Mining 2025 (SDM-25) (SDM occurs May 1-3, 2025 - full schedule here)

Thanks to our sponsor, SSCI for their support of this event.

Main Keynote: Andrea Stocco
What is "meta" in metacognition? Insights from brain's cognitive architecture

Andrea Stocco is a computational cognitive neuroscientist from Friuli, Italy. Dr. Stocco earned his Ph.D. from the University of Trieste and completed postdoctoral work at Carnegie Mellon University. He is now an associate

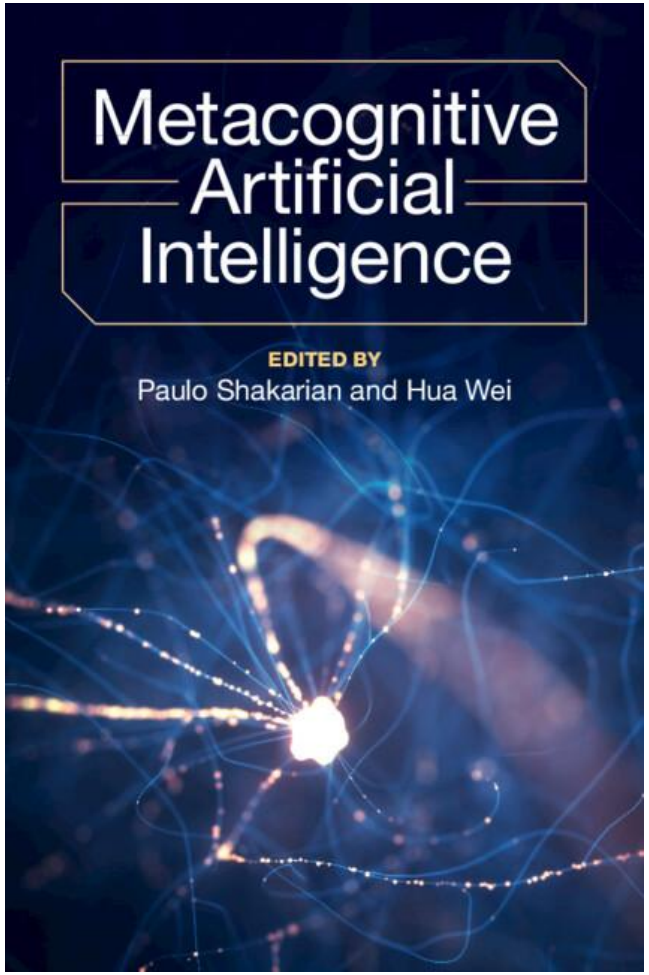
Archives
- July 2025

Categories
- Uncategorized

Metacognition Resources

Resource page:
https://metacognition.syracuse.edu

Edited volume by Cambridge University Press



Syracuse University

Learn more ←

Paulo Shakarian

K.G. Tan Endowed Professor of AI
Director, Leibniz Lab

Leibniz Lab

pashakar@syr.edu
https://leibniz.syracuse.edu