Probabilistic Foundations for Metacognition via Hybrid-AI

Paulo Shakarian[1], Gerardo I. Simari[2], Nathaniel D. Bastian[3]

[1]Arizona State University, Tempe, AZ USA

[2] Departament of Computer Science and Engineering, Universidad Nacional del Sur (UNS) &
Institute for Computer Science and Engineering (UNS-CONICET), Bahía Blanca, Argentina

[2]United States Military Academy, West Point, NY USA

pshak02@asu.edu, gis@cs.uns.edu.ar, nathaniel.bastian@westpoint.edu

Abstract

Metacognition is the concept of reasoning about an agent's own internal processes, and it has recently received renewed attention with respect to artificial intelligence (AI) and, more specifically, machine learning systems. This paper reviews a hybrid-AI approach known as "error detecting and correcting rules" (EDCR) that allows for the learning of rules to correct perceptual (e.g., neural) models. Additionally, we introduce a probabilistic framework that adds rigor to prior empirical studies, and we use this framework to prove results on necessary and sufficient conditions for metacognitive improvement, as well as limits to the approach. A set of future research directions is also provided.

1 Introduction

Originally a concept from developmental psychology (Flavell 1979), metacognition refers to reasoning about an agent's own internal processes (Wei et al. 2024). This idea of metacognition, considered the human brain's "self-monitoring process" (Demetriou et al. 1993), has been studied in a diverse set of fields (Li et al. 2017; Izzo, Märtens, and Pan 2019; Caesar et al. 2020). Over the years, its study has been proposed in the field of AI (Cox and Raja 2011; Cox 2005), and recent interest (Wei et al. 2024) has reinvigorated this discussion.In this paper, we review a recent, hybrid-AI style of metacognition known as "error detection and correction rules" (EDCR) that has been explored in a variety of models and use cases (see Table 1). In short, the proposal is a hybrid-AI approach in which logical rules are learned to characterize the model performance of well-trained perceptual (e.g., neural) models. This paper serves as a review of the results of these recent articles, and improves the underlying theoretical underpinnings by framing them in terms of a probabilistic argument, providing new insights into the use of hybrid-AI for metacognition. In particular, our novel theoretical framework allows us to explain some empirically observed results, as well as leads us to several new open research questions.

2 Error Detection and Correction Rules

In this framework, we are given a well-trained model, normally denoted as f, that takes some continuous input (e.g.,

Paper and use case	Base model	Source of conditions
Kricheli et al. (2024) Vision	ViT	Hier. class labels, other models
Xi et al. (2024) Trajectory classification	CNN, LRCN, LRCNa	Dom. kn, other models
Lee et al. (2024) Time series classification	CNN, RNN, Attention	Other models

Table 1: Summary of existing literature on EDCR.

a vector) and returns a set of class labels. We will use subscripts (e.g., f_i or f_1) to denote multiple models when relevant. We note that the model need not be neural (though all existing studies have focused on neural models), and in general we do not assume access to the model weights. We now introduce a small running example.

Example 2.1. *Let f_{car} be a neural model trained on data set D_{tng} such that given a sample (that we denote x) returns a subset of labels $\{ford, toyota, dodge, us, japan\}$. So, for example, for sample ω, $f_{car}(\omega) = \{dodge, us\}$.*

The work on EDCR introduces the notion of a *metacognitive condition*, which is some aspect of the environment, sensor, model, or metadata that may cause the model to be incorrect in a given circumstance. In the existing work, the metacognitive condition (which we refer to as "condition", for short) has come from domain knowledge, model output (e.g,, labels at different levels of a hierarchy), or other models (e.g., different architectures, trained on different data) – this is summarized in Table 1. This leads us to the first type of rule, the error detecting rule, which is expressed in a first order logic syntax as follows:

$$error_\alpha^i(X) \leftarrow pred_\alpha^i(X) \wedge \bigvee_{j \in DC_i} cond_j(X) \qquad (1)$$

In words, this rule states that if any of the conditions in set DC_i are met, and model i predicts class α, then the model has an erroneous detection. We note that we can replace the disjunction with a singleton and have multiple rules for a given model-class pair (i.e., using a logic program). The second type of rule is a correction rule, which has the following

syntax:

$$corr_\beta^i(X) \leftarrow \bigvee_{j,\alpha \in CC_\beta} \left(cond_j(X) \land pred_\alpha^i(X) \right) \qquad (2)$$

In words, this rule states that given a set of condition-class pairs relevant to class β (denoted CC_β), for any of those pairs (e.g., model i predicting α for the sample while condition j is true), the rule dictates that the sample should be relabeled with label β. In both Xi et al. (2024) and Lee et al. (2024), the detection step happens first (in both training and inference), allowing for samples to have some labels "erased" and only samples with erased labels will have new labels assigned by a correction rule.

Example 2.2. *Consider the model f_{car} from Example 2.1. Suppose we have a detection rule of the form:*

$$error_{toyota}^{car}(X) \leftarrow pred_{toyota}^{car}(X) \land cond_{us}(X)$$

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:

$$corr_{dodge}^{car}(X) \leftarrow cond_{us}(X) \land pred_{toyota}^{car}(X)$$

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

In Example 2.2, one could imagine having a priori knowledge that predicting $japan$ and us together would yield an inconsistency. However, in such a case, it may make more sense to define the condition as the case where $\{japan, us\} \subseteq f_{car}(x)$. In fact, in prior work, the primary knowledge engineering task is to design the conditions, not the rules, which are learned from data, primarily using combinatorial techniques. For example, in Xi et al. (2024) detection rules are learned via submodular maximization of precision while constraining recall reduction, while in Kricheli et al. (2024) detection rules are learned via submodular ratio maximization to optimize toward F1. We also note that the detection rule of Example 2.2 can also function as a constraint on class relationships, in this case meaning that us and $toyota$ cannot go together. This turns out to be useful, as the rules are already expressed in logic. This enabled Kricheli et al. (2024) to take learned detection rules and retrain a model with an LTN (Badreddine et al. 2022) loss function to gain improved model performance.

3 Probabilistic Interpretation and Results

We now present a new probabilistic interpretation of EDCR. First, we introduce a bit of notation: when the object in question is understood, we shall use the notation f_i to refer to the set of propositions produced by model f_i on the object. So, for example, $\alpha \in f_i$ means that model f_i predicted label α. We shall also use the notation gt to denote a set of ground truth labels for the object; so, the statement $\alpha \in f_i, \alpha \notin gt$ means that model f_i incorrectly predicted class α for the object. With this in mind, we can represent the precision and recall for a model as the following conditional probabilities.

$$\text{Precision:} \quad P_\alpha = \mathbf{P}(\alpha \in gt \mid \alpha \in f_i) \qquad (3)$$

$$\text{Recall:} \quad R_\alpha = \mathbf{P}(\alpha \in f_i \mid \alpha \in gt) \qquad (4)$$

We will also consider a random variable D that specifies a specific distribution. We can consider conditional probabilities with an unspecified distribution (as shown above) to be approximated from training data, while setting D to be something specific would signify being out-of-distribution: e.g., the precision for model i on class α under distribution d is written as $\mathbf{P}(\alpha \in gt \mid \alpha \in f_i, D = d)$. In what follows, we shall use M to denote the set of metacognitive conditions. We note that this framework allows us to precisely define what it means for a condition to be error detecting, and introduce a new, formal definition to that effect.

Definition 3.1 (Error Detecting). *A metacognitive condition c is* error detecting *with respect to model i, class α, and distribution d if $\mathbf{P}(\alpha \in gt \mid \alpha \in f_i, c \in M, D = d)$ is less than or equal to $\mathbf{P}(\alpha \in gt \mid \alpha \in f_i, D = d)$.*

Intuitively, a condition is error detecting if the precision of the model drops when the condition is true. The second property is distribution invariance.

Definition 3.2 (Distribution Invariance). *A metacognitive condition c is* distribution invariant *with respect to model i, class α, and set of distributions $\mathcal{D}$ if for any distributions $d \in \mathcal{D}$ it is error detecting with respect to that distribution.*

At first glance, distribution invariance seems to be a strong definition, and some may even think the set of conditions meeting that criteria would be small. However, consider conditions based on constraints among classes; e.g., inconsistent assignments in a multi-class classification problem is one such case, and the configuration of the sensor is another.

With conditions in mind, we define precision and recall after applying a metacognitive condition:

$$\text{Precision:} \quad P_\alpha^c = \mathbf{P}(\alpha \in gt \mid \alpha \in f_i, c \notin M) \qquad (5)$$

$$\text{Recall:} \quad R_\alpha^c = \mathbf{P}(\alpha \in f_i, c \notin M \mid \alpha \in gt) \qquad (6)$$

Our first result shows how much the precision changes after applying a metacognitive condition. The argument in this paper characterizes the change in precision using a probabilistic argument and, notably, the result is obtained without any assumptions of independence.

Theorem 3.1 (Metacognitive Precision Change). *The following identity holds:*

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

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

In Xi et al. (2024), an analogous result is shown. Both results suggest finding conditions that attempt to maximize the product of probabilities $\mathbf{P}(c \in M \mid \alpha \in f_i)$ and $\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M)$ is desirable for precision improvement by "erasing" labels – these correspond to support and confidence in that paper. It also turns out that this product has computationally desirable properties, as it is submodular (proven in Xi et al. (2024)). However, we point out that the result of Xi et al. (2024) did not frame the preliminaries in terms of probability, and hence it was not clear if there were latent assumptions; this probabilistic interpretation and the corresponding proof (in the appendix at https://arxiv.org/abs/2502.05398) strengthen the results,

as now they clearly do not rely on independence. Further, we obtain additional insights built on this result discussed throughout this paper, and one notable insight is that we immediately obtain a necessary and sufficient condition for obtaining improvement in precision:

$$\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M) > 1 - P_\alpha \qquad (7)$$

We can think of $1 - P_\alpha$ as the residual of the model – how much room it has to improve precision. We can also think of this result as having a practical application in determining if a condition is invariant. For example, we would likely assume that $1 - P_\alpha$ increases for out-of-distribution samples, and perhaps we could determine some ϵ where for some $D = d$ that $\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M, D = d)$ is within ϵ of confidence $\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M)$. So, for example, if we identify the confidence value by examining multiple samples, and obtain an error, we can determine up to which value of $1 - P_\alpha$ the condition is expected to be invariant. Another aspect of this result is that we can also prove that our error-detecting definition (Definition 3.1) is an equivalent condition.

Theorem 3.2 (Error Detection is Necessary and Sufficient). *Condition c is error detecting iff $P_\alpha^c \geq P_\alpha$.*

Proof. ($\rightarrow$) By way of contradiction, assume c is error detecting and $P_\alpha^c < P_\alpha$, which gives us:

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

$$\mathbf{P}(\alpha \in f_i)(\mathbf{P}(\alpha \in gt, \alpha \in f_i) -$$
$$\mathbf{P}(\alpha \in gt, \alpha \in f_i, c \in M)) <$$
$$\mathbf{P}(\alpha \in gt, \alpha \in f_i)(\mathbf{P}(\alpha \in f_i) - \mathbf{P}(\alpha \in f_i, c \in M))$$

$$\mathbf{P}(\alpha \in f_i)(-\mathbf{P}(\alpha \in gt, \alpha \in f_i, c \in M)) <$$
$$\mathbf{P}(\alpha \in gt, \alpha \in f_i)(-\mathbf{P}(\alpha \in f_i, c \in M))$$

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

However, this contradicts the definition of error detecting. The remaining part of the proof is in the appendix. $\square$

Characterization of recall. Next, we look at the change in recall. The recall prior to disregarding the classification for a prediction is $\mathbf{P}(\alpha \in f_i \mid \alpha \in gt)$. Once we apply the condition, the recall for class i becomes $\mathbf{P}(\alpha \in f_i, c \notin M \mid \alpha \in gt)$. Note that in doing so, we are essentially turning off model predictions, which can only reduce recall. We can show an equivalent value to the reduction in recall with the following result. Again, there is an analogous result in Xi et al. (2024); however our proof (in the appendix) presents a probabilistic argument and the proof clearly illustrates that no independence assumptions are made.

Theorem 3.3 (Recall Reduction). $\mathbf{P}(\alpha \in f_i \mid \alpha \in gt) - \mathbf{P}(\alpha \in f_i, c \notin M \mid \alpha \in gt) =$

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

Proof.

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

Here we see that the primary driver of recall reduction is $\mathbf{P}(\alpha \in gt \mid \alpha \in f_i, c \in M)$, which is the probability of the model obtaining the correct answer under the metacognitive condition. Likewise, the other term dependent on the conditions that impacts the reduction in recall is $\mathbf{P}(c \in M \mid \alpha \in f_i)$, which is the probability of a condition occurring with a prediction. Note that both the decrease in recall and the increase in precision trend with this quantity. In the next section, on limitations, we shall understand how this quantity can be bounded.

4 Limitations of Metacognitive Conditions

We now explore some of the limits of our approach to metacognitive improvement. Anecdotally, we noticed in prior metacognitive applications of EDCR that often detection seems easier than correction. In Xi et al. (2024) and Lee et al. (2024), correction was conducted by changing the class label resulting from a condition-class pair that led to an error. In the notation of this paper, such a precondition would be "$\alpha \in f_i, c \in M$", meaning the condition of the model predicting class α while condition c is also true. This allows us to overcome the detection-induced deficit (a consequence of Theorem 3.3), which was demonstrated empirically in Lee et al. (2024) where such metacognitive correction could ensemble rules to directly improve recall over single-model baselines. However, it was less effective in improving recall in the experiments of Xi et al. (2024), where the use case consisted of five classes. To understand why this occurs, consider a model that cannot distinguish between a set of samples, all classified under condition c. 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. However, the next most probable class, j, is lower still, and picking this would lower overall loss. As a result, without another condition or something else to distinguish these samples, the model and the metacognitive correction cannot re-assign those samples a new label that improves overall performance, while at the same time effectively identifying a case where the model had an error with a high probability. Consider the following:

Example 4.1. *Consider the scenario from Example 2.2. Suppose in cases where the error is observed (both classes toyota and us are predicted) that despite dodge being the "best" correction of class toyota we have the following:*

$$\mathbf{P}(dodge \in f_{car} \mid toyota \in f_{car}, us \in f_{car})$$
$$\leq \mathbf{P}(dodge \in f_{car} \mid toyota \in f_{car})$$

It turns out that the situation in Example 4.1 leads to a reduction in precision for the class by which the label is re-assigned. We now formalize this argument. Intuitively, if the precision for class j conditioned on reclassifying items originally classified as i under condition c is lower, then the overall precision for classification of class j will drop.

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

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

Proof. By way of contradiction, assume:

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

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

$$\frac{\mathbf{P}(\alpha \in f_i, c_\alpha \in M)}{\mathbf{P}(\beta \in f_i)} < \frac{\mathbf{P}(\beta \in gt, \alpha \in f_i, c_\alpha \in M)}{\mathbf{P}(\beta \in gt, \beta \in f_i)}$$

$$\frac{\mathbf{P}(\beta \in gt, \beta \in f_i)}{\mathbf{P}(\beta \in f_i)} < \frac{\mathbf{P}(\beta \in gt, \alpha \in f_i, c_\alpha \in M)}{\mathbf{P}(\alpha \in f_i, c_\alpha \in M)}$$

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

which contradicts the statement of the theorem. $\square$

We also note there are other potential limits on reclassification due to EDCR. Specifically, if rules are learned in a manner that allows for inconsistencies (which would be possible if rules are learned among different models independently) then proper corrective action becomes less clear. This is an active area of inquiry.

Limitations to precision improvement/recall reduction. In another new result from our analysis, we show we can bound the quantity $\mathbf{P}(c \in M \mid \alpha \in f_i)$, which as pointed out earlier can magnify or suppress the amount of change in precision, or amplify the reduction in recall based on the prevalence of the metacognitive condition. Specifically, this is bounded by $\mathbf{P}(c \in M \mid \alpha \notin gt, \alpha \in f_i)$ when c is error-detecting (shown in the next corollary). The practical application of the result, when identifying a metacognitive condition, is that we can understand the power of such a condition by only analyzing a subset of a dataset where the model was correct, which may have implications for large-scale data analytics and imbalanced classification problems. Note that we never assume $c \in M$ is independent of $\alpha \notin gt$ or $\alpha \in f_i$ – this result stems directly from the assumption that c is error detecting.

Corollary 4.1. *If c is error detecting, then:*

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

Proof. By definition of error detecting, we have:

$$\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i c \in M) \geq \mathbf{P}(\alpha \notin gt \mid \alpha \in f_i) \quad (8)$$

$$= \frac{\mathbf{P}(c \in M \mid \alpha \in f_i)\mathbf{P}(\alpha \notin gt \mid \alpha \in f_i, c \in M)}{\mathbf{P}(c \in M \mid \alpha \notin gt, \alpha \in f_i)}. \quad (9)$$

This in turn gives us:

$$1 \geq \frac{\mathbf{P}(c \in M \mid \alpha \in f_i)}{\mathbf{P}(c \in M \mid \alpha \notin gt, \alpha \in f_i)} \quad (10)$$

$$\mathbf{P}(c \in M \mid \alpha \notin gt, \alpha \in f_i) \geq \mathbf{P}(c \in M \mid \alpha \in f_i), \quad (11)$$

thus completing the proof. $\square$

5 Conclusion: Directions for Future Research

In this paper, we reviewed a hybrid-AI technique for metacognition known as EDCR, and provided a probabilistic argument that supports previous findings. This also suggests future research directions, so we end the paper with a discussion of some of these potential areas of inquiry.

Role of Domain Knowledge and Inconsistency. In Example 2.2 we showed how metacognitive conditions can be used to identify inconsistencies, and the work of Kricheli et al. (2024) demonstrates this empirically by both recovering latent constraints and using that to improve model loss via LTN. When we view conditions from the standpoint of distribution invariance, logical inconsistency clearly meets these criteria. This raises an interesting question: can consistency with domain knowledge be used for error correcting? Neurosymbolic approaches already use this concept to reduce training loss (Bizzarri et al. 2024). However, understanding how consistency can be used for metacognitive correction remains an open question.

Multi-model / Multi-modal reasoning. The results of this paper suggest that EDCR can be most effective for improving model precision while sacrificing recall (Theorems 3.1 and 3.3). The results of Lee et al. (2024) essentially leverage this property to ensemble models together, as different models can be precise and their combined recall will lead to an increase if they are identifying different phenomena. The ensembling of multi-modal models through EDCR seems to be a natural fit, as models of different modalities likely exhibit complementary capabilities. However, work like Lee et al. (2024) assumes one mode is the "base model" while the others are used to correct it. The study of EDCR (or other metacognitive methods) to ensemble two or more models without a designed "base model" remains an open question.

Online metacognition and data efficiency. Quantities such as $\mathbf{P}(\alpha \in gt \mid \alpha \in f_i, c \in M)$ and $\mathbf{P}(c \in M \mid \alpha \in f_i)$ (which are analogous to confidence and support) are typical byproducts of metacognitive rule learning under EDCR, and as seen in this paper are key quantities for validating invariance of conditions (Theorem 3.2). Likewise, an experiment in Xi et al. (2024) shows how rules can be learned from a new distribution that are effective with a small portion of data. Together, can these results suggest rapid adoption to a new distribution of data by identifying conditions on the fly (i.e., online learning of metacognitive conditions)?

Acknowledgments

References

Badreddine, S.; d'Avila Garcez, A.; Serafini, L.; and Spranger, M. 2022. Logic Tensor Networks. *Artificial Intelligence*, 303: 103649.

Bizzarri, A.; Jalaian, B.; Riguzzi, F.; and Bastian, N. D. 2024. A Neuro-Symbolic Artificial Intelligence Network Intrusion Detection System. In *Proc. ICCCN*, 1–9.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In *Proc. IEEE/CVF CVPR*, 11621–11631.

Cox, M. T. 2005. Metacognition in computation: A selected history. In *Proc. AAAI Spring Symposium: Metacognition in Computation*, 1–17.

Cox, M. T.; and Raja, A. 2011. *Metareasoning: Thinking about thinking*. MIT Press.

Demetriou, A.; Efklides, A.; Platsidou, M.; and Campbell, R. L. 1993. The architecture and dynamics of developing mind: Experiential structuralism as a frame for unifying cognitive developmental theories. *Monographs of the society for research in child development*, i–202.

Flavell, J. H. 1979. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10): 906.

Izzo, D.; Märtens, M.; and Pan, B. 2019. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics*, 3: 287–299.

Kricheli, J. S.; Vo, K.; Datta, A.; Ozgur, S.; and Shakarian, P. 2024. Error Detection and Constraint Recovery in Hierarchical Multi-Label Classification without Prior Knowledge. In *Proc. ACM CIKM*, 3842–3846. ACM.

Lee, N.; Ngu, N.; Sahdev, H. S.; Motaganahall, P.; Chowdhury, A. M. S.; Xi, B.; and Shakarian, P. 2024. Metal Price Spike Prediction via a Neurosymbolic Ensemble Approach. arXiv:2410.12785.

Li, B.-h.; Hou, B.-c.; Yu, W.-t.; Lu, X.-b.; and Yang, C.-w. 2017. Applications of artificial intelligence in intelligent manufacturing: a review. *Frontiers of Information Technology & Electronic Engineering*, 18: 86–96.

Wei, H.; Shakarian, P.; Lebiere, C.; Draper, B. A.; Krishnaswamy, N.; and Nirenburg, S. 2024. Metacognitive AI: Framework and the Case for a Neurosymbolic Approach. In *Proc. NeSy Part II*, volume 14980 of *Lecture Notes in Computer Science*, 60–67. Springer.

Xi, B.; Scaria, K.; Bavikadi, D.; and Shakarian, P. 2024. Rule-Based Error Detection and Correction to Operationalize Movement Trajectory Classification. arXiv:2308.14250.