Consistency-based Abductive Reasoning over Perceptual Errors of Multiple Pre-trained Models in Novel Environments

Mario Leiva[1], Noel Ngu[2], Joshua Shay Kricheli[3], Aditya Taparia[2], Ransalu Senanayake[2], Paulo Shakarian[3], Nathaniel D. Bastian[4], John Corcoran[5], and Gerardo Simari[1]

[1]DCIC, Universidad Nacional del Sur (UNS) & ICIC (UNS-CONICET), Bahía Blanca, Argentina
[2]Arizona State University, Tempe, AZ USA
[3]Syracuse University, Syracuse, NY USA
[4]United States Military Academy, West Point, NY USA
[5]Systems Planning & Analysis, Alexandria, VA USA
mario.leiva@cs.uns.edu.ar, {nngu2, ataparia, ransalu}@asu.edu, {jkrichel, pashakar}@syr.edu,
nathaniel.bastian@westpoint.edu, jack.fd.corcoran@gmail.com, gis@cs.uns.edu.ar

Abstract

The deployment of pre-trained perception models in novel environments often leads to performance degradation due to distributional shifts. Although recent artificial intelligence approaches for metacognition use logical rules to characterize and filter model errors, improving precision often comes at the cost of reduced recall. This paper addresses the hypothesis that leveraging multiple pre-trained models can mitigate this recall reduction. We formulate the challenge of identifying and managing conflicting predictions from various models as a consistency-based abduction problem, building on the idea of abductive learning (ABL) but applying it to test-time instead of training. The input predictions and the learned error detection rules derived from each model are encoded in a logic program. We then seek an abductive explanation—a subset of model predictions—that maximizes prediction coverage while ensuring the rate of logical inconsistencies (derived from domain constraints) remains below a specified threshold. We propose two algorithms for this knowledge representation task: an exact method based on Integer Programming (IP) and an efficient Heuristic Search (HS). Through extensive experiments on a simulated aerial imagery dataset featuring controlled, complex distributional shifts, we demonstrate that our abduction-based framework outperforms individual models and standard ensemble baselines, achieving, for instance, average relative improvements of approximately 13.6% in F1-score and 16.6% in accuracy across 15 diverse test datasets when compared to the best individual model. Our results validate the use of consistency-based abduction as an effective mechanism to robustly integrate knowledge from multiple imperfect models in challenging, novel scenarios.

Code — github.com/lab-v2/EDCR_PyReason_AirSim
Extended version — https://arxiv.org/abs/2505.19361

Introduction

The use of pre-trained models is very common in tasks that require perception data, such as classification and object detection in images and video (Han et al. 2021; Parisi et al. 2022). Another scenario in which differences arise is when we know we will be deploying in different environments because we are using the models that we have available—we refer to these issues as *deployment in novel environments*. As a specific example, consider emergency response, where perception models examining a disaster must contend with unforeseen environmental changes even when trained on data for a similar region. Another example is an NGO providing aid to a remote location where training data was unavailable. In both cases, we can be assured that the environment in which the perception models operate is *novel* with respect to what they were trained on.

Psychologists have shown that humans deal with novelty through metacognition (Thompson 2009) by leveraging the dual Type 1 / Type 2 processing (Evans and Stanovich 2013) (i.e., "dual process theory" (Wason and Evans 1974) popularized by (Kahneman 2012)). In particular, a collection of autonomous "Type 1" systems perceives information that may also lead to a "metacognitive cue" that triggers additional "Type 2" reasoning. Following the renewed interest in metacognitive artificial intelligence (AI) (Wei et al. 2024; Johnson et al. 2024), recent work has shown that we can learn rules that provide metacognitive cues about machine learning model failure (Kricheli et al. 2024)—that work, however, uses a single model and does not provide further reasoning resulting from the metacognitive cue. Meanwhile, work on abductive learning (Dai et al. 2019) allows for adjustment to a single machine learning model at training time based on abductive feedback. In this work, we use cues provided by logic programs modeling the failures of multiple models in an abductive framework at inference time to allow for enhanced perception in novel environments.

In this work, our working hypothesis is that by deploying more than one model we are able to at least partially address this drawback; this is the same underlying principle behind standard approaches in machine learning for developing ensembles of models, but our approach goes beyond such standard practices since we apply novel metacognitive AI techniques. In particular, as shown in Figure 1, we leverage existing rule learning techniques to derive a logic program consisting of metacognitive rules across a set of perception models, and frame the task of identifying errors across all

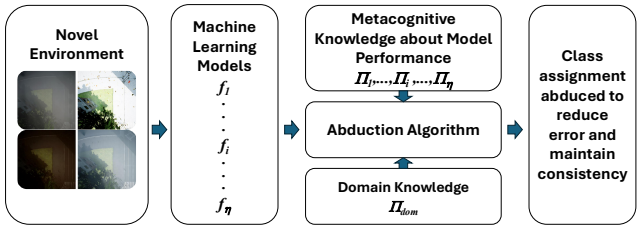

Figure 1: Overview of our Consistency-based Abductive Reasoning Approach. Here η machine learning models perceive a novel environment. Their results are considered with domain knowledge and metacognitive information about the models (learned independently and with the same training data) to abduce a set of results that exhibit consistency and reduce perceptual errors.

models as a *consistency-based abduction problem*. We then show that such error identification problems can be posed as integer programs, and provide a scalable heuristic algorithm to solve the abductive reasoning task. Noteworthy in our approach is that the logic program is created by rules learned for each perceptual model based on their training data—so there is no a priori knowledge of test data (i.e., no leakage). Further, the rules for the individual models are learned independently from each other, so we assume there is no existing knowledge of how the models perform together.

Finally, we present a thorough experimental evaluation using an extended, highly-controlled aerial imagery dataset with diverse distributions (Ngu et al. 2025). Our results demonstrate that our abductive-based reasoning approach—by effectively managing inconsistencies while maximizing predictions—achieves superior performance compared to individual models across numerous test datasets.

Related Work

This work is closely related to (Xi et al. 2024), where Error Detection and Correction rules are introduced. The main difference with our approach is that they only consider a single model. As we discuss below, here we only leverage error detection rules, though the approach can easily accommodate the use of correction rules as well.

Abductive learning (ABL) (Dai et al. 2019) also leverages abduction to reduce perception errors and, like our approach, relies on some domain knowledge (for instance, we require knowledge about consistency of predictions). However, ABL uses this to improve performance based on model training and assumes that the test environment is not entirely novel. In this work, we relax both assumptions and use abduction only at test time. We note that a follow-on study, called "ABL with new concepts" (*ABL*$_{nc}$) (Huang et al. 2023) extends ABL—in a manner similar to EDCR—with the ability to identify previously "unknown" concepts, which EDCR (Xi et al. 2024) was also shown to have. Like ABL, this approach is also focused on using abduction at model training and not inference. In this work, we do not extend the concept scheme (though approaches like ABL/*ABL*$_{nc}$, EDCR, and HDC could all potentially be helpful) but rather change the distribution of perceptual data by which such concepts are extracted. We note that early work on abduction often focused on the diagnosis of errors and faults (Poole 1989); however, to our knowledge this has not been applied to perception tasks at test time. This early work inspired our use of abduction to identify perception errors.

Recently, a concept known as "test-time training" (TTT) (Sun et al. 2020) has gained traction in the machine learning literature and has shown importance in reasoning tasks as well (Akyürek et al. 2024). In this approach, the neural model is trained in a manner such that it performs self-supervised training during test time to improve inference. While this shows ability to improve out-of-distribution results, it is also a method designed to improve model training, and was noted to have limitations based on the classes used in the self-supervised training. We note that such a TTT model could be treated as any other pre-trained model in our framework, and we can even envision different variants of TTT (e.g., with different strategies for the self-supervised portion) better working together by leveraging our results.

Finally, in (Sutor et al. 2022), the authors also explore combining a set of pre-trained models using hyperdimensional computing (HDC). However, their method relies on having a set of training samples from the same target distribution as the test set, enabling the joint training of an HDC "gluing" module. In contrast, here we assume that data from the same distribution as the test set is unavailable and that the models are trained independently. Furthermore, their method depends on the output layers of the neural networks and lacks explainability, unlike our rule-based approach.

Consistency-based Abduction Problem

Technical Preliminaries. We consider the problem of object identification for a given set of perception data Ω (Cheng and Han 2016) and assume the availability of a set η of perception models $\mathcal{F} = \{f_1, f_2, \ldots, f_\eta\}$, where each model generates a set of predictions over a shared set of m object classes $\mathcal{C} = \{c_1, ..., c_m\}$. Under the unique name assumption[1], every object $\omega \in \Omega$ identified by one or more models has at least one associated fact of the form $f_i(\omega) = c_j$, meaning that model i has identified object ω as belonging to class c_j. We will denote the set of these facts (hereafter "observations") as O.

We note that, as we are working with multiple models, there may be differences in which class is assigned a particular object. Further, some of the models may make mistakes. Hence, we introduce the predicate *accept* where for some model f_i and class c, *accept*(i, c) is true if we wish to accept model f_i's results when it returns class c. We denote the set of all acceptance atoms with $\mathcal{H}$, and typically refer to a subset of this set as a "hypothesis", denoted H. Our goal is to find a set H that meets certain criteria (to be described).

A key piece of our framework is leveraging the ability to identify metacognitive cues. As such, we assume that each model f_i has an associated logic program Π_i consisting of rules that, when fired, provide such metacognitive cues

[1]See Section 1 of the appendix in the extended version for the implementation details of how we employ this assumption.

about errors. These logic programs can be learned from the model training data as per prior work (Kricheli et al. 2024; Xi et al. 2024) and are of the following form:

$$\textit{error}(i, c, \omega) \leftarrow (f_i(\omega) = c) \wedge \textit{cue}(\omega)$$

Intuitively, if metacognitive cue is present for object ω and it was classified as class c by model f_i, then we can assume that there was an error made in the assignment. In (Kricheli et al. 2024), the metacognitive cues (referred to as "conditions" in that work) are selected from a set of candidate conditions. We include details of how we applied their learning algorithm in extended version.

In addition to each of the Π_i's, we assume a logic program $\Pi_{\textit{helper}}$ that consists of rules of the following form:

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

In other words, if we accept the results of a given class from a model and no error is reported for that model-class pair, then we can assign object ω class c. Another set of rules $\Pi_{\textit{dom}}$ specifies domain knowledge. In this work, we are primarily concerned about integrity constraints that prevent a given object from being classified with conflicting classes. In this paper, rules in $\Pi_{\textit{dom}}$ are of the form:

$$\neg \textit{assign}(c', \omega) \leftarrow \textit{assign}(c, \omega) \tag{1}$$

We use symbol Π to denote $\Pi_{\textit{dom}} \cup \Pi_{\textit{helper}} \cup (\bigcup_i \Pi_i)$. Further, leveraging simple stratification and the limited use of negations of *error* and *assign* predicates, we can implement this in a tractable monotonic logic.

Abduction Problem. We build a consistency-based abduction problem (Eiter and Gottlob 1995) whereby a set of hypotheses H is found that is consistent with a set of observations O and domain knowledge Π. We use logic programs as our representation language (specifically, the framework of (Aditya et al. 2023), though others such as Prolog and Datalog are possible). Here, H and O are expressed as atomic facts, while Π is a set of logical rules. For some universe of hypothesis atoms $\mathcal{H}$, we wish to find $H \subseteq \mathcal{H}$ such that $H \cup O \cup \Pi$ is consistent.

From a practical perspective, we may wish to allow for some (small) amount of inconsistency in the resulting output with respect to $\Pi_{\textit{dom}}$. For a given hypothesis H, we define $\textit{Inc}(H)$ as the normalized number of ground rules in $\Pi_{\textit{dom}}$ not entailed by $(H \cup O \cup \Pi) \setminus \Pi_{\textit{dom}}$. Given the rule format for $\Pi_{\textit{dom}}$ (cf. Eq. 1), computing this is trivial (doing so in other languages is left to future work). We use the symbol $\delta \in [0, 1]$ to denote an acceptable threshold for this value.

As the set of possible solutions is large, and many such solutions could be consistent with $O, \Pi \setminus \Pi_{\textit{dom}}$, we employ a notion of *parsimony* as usual with abduction (Peng and Reggia 1990). Our parsimony function will maximize the number of atoms of the form $\textit{assign}(c, \omega)$ entailed by the minimal model of $(H \cup O \cup \Pi) \setminus \Pi_{\textit{dom}}$; we denote this quantity $\textit{Pred}(H)$. The reason for maximization (as opposed to minimization) is threefold: (i) by maximizing assigned output, we minimize the suspected errors (as the models are presumably well trained); (ii) we generally seek to maximize recall; and (iii) the construction of rules $\Pi_{\textit{helper}}$ minimizes the impact of over-assignment of model results (as assignment occurs based not only on acceptance, but having at least one model perceive the object as belonging to the class and that it does so error-free). With this in mind, we can frame the following optimization problem:

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

subject to:

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

and

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

We again note that δ can be used to gauge the amount of inconsistency with domain knowledge ($\Pi_{\textit{dom}}$) as opposed to inconsistency with metacognitive knowledge ($\bigcup_i \Pi_i$). There could be variance in the amount of metacognitive cues triggered by a given Π_i, and a user may want to vary this as well. Fortunately, the approach we used for metacognitive rule learning (Kricheli et al. 2024; Xi et al. 2024) provides an intuitive hyperparameter (denoted with ϵ) that can readily be interpreted as the expected reduction in recall experienced by disregarding erroneous predictions. We examine varying ϵ as well as using a heuristic to set it automatically.

Integer Program (IP) Formulation. To solve our associated optimization, we provide an exact integer programming solution and heuristic algorithm. We first review the integer program. The goal of the IP is to find an optimal hypothesis H—represented by a set of binary decision variables—that maximizes the total number of entailed assignments ($\textit{Pred}(H)$) while ensuring the inconsistencies with our domain knowledge ($\textit{Inc}(H)$) remain below the threshold δ.

Formally, this can be expressed as

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

subject to:

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

where $A_{c,\omega}$ is a binary variable indicating whether object ω is assigned to class c, $\textit{Con}_{\omega,(c,c')}$ is a binary variable indicating a conflict between assignments c and c' for object ω.

We further define the following constants and variables: $\textit{pred}_{f,c,\omega}$ is a constant set to 1 if $(\omega, c) \in \textit{assigns}$ and $\textit{pred}_c^f(\omega) \in \Gamma^*(\Pi)$, 0 otherwise; variable $X_{\omega,f,c} \in \{0, 1\}$ indicates whether object ω is considered for model f and class c; and variable $\textit{Elim}_{f,c} \in \{0, 1\}$ indicates whether predictions from model f for class c are excluded; this last variable directly implements our choice of hypothesis H. Setting $\textit{Elim}_{f,c} = 0$ is equivalent to including the atom $\textit{accept}(f, c)$ in our hypothesis H, thereby trusting model f for class c. Conversely, setting $\textit{Elim}_{f,c} = 1$ excludes it. We can now present the set of constraints. First, for each f, c, ω we have constraints of the form:

$$X_{\omega,f,c} \leq 1 - \textit{Elim}_{f,c} \tag{2}$$

$$X_{\omega,f,c} \cdot \textit{pred}_{f,c,\omega} \leq A_{c,\omega} \tag{3}$$

Next, for each c, ω we have:

$$A_{c,\omega} \leq \sum_f X_{\omega,f,c} \cdot \textit{pred}_{f,c,\omega} \tag{4}$$

For each $\omega \in \Omega, (c, c') \in IC$:

$$A_{c,\omega} + A_{c',\omega} - 1 \leq \mathit{Con}_{\omega,(c,c')} \tag{5}$$

for each ω, we have:

$$\sum_{c \in \mathcal{C}} A_{c,\omega} \geq 1 \tag{6}$$

Finally:

$$\sum_{\omega \in \Omega} \sum_{(c,c') \in IC} \mathit{Con}_{\omega,(c,c')} \leq \delta \tag{7}$$

They respectively ensure the elimination of invalid predictions, consistency between predictions and assignments, upper bounds on assignments per object, that conflicts are adequately managed, that each object is assigned at least one class, and that the global conflict threshold holds.

The IP formulation described by Equations (3)–(8) translates into a model with a number of variables and constraints dependent on the number of unique objects (N), models ($|\mathcal{F}|$), classes ($|\mathcal{C}|$), and integrity constraints ($|IC|$). Specifically, the IP model involves decision variables for assignments ($A_{c,\omega}$), conflict indicators ($\mathit{Con}_{\omega,(c,c')}$), model-class eliminations ($\mathit{Elim}_{f,c}$), and object consideration ($X_{\omega,f,c}$). The total number of variables is primarily driven by the $N \times \mathcal{F} \times \mathcal{C}$ term (associated with $X_{\omega,f,c}$), resulting in an overall count in $O(N \cdot |\mathcal{F}| \cdot |\mathcal{C}|)$. Similarly, the number of constraints also scales in $O(N \cdot |\mathcal{F}| \cdot |\mathcal{C}|)$.

From a knowledge representation and reasoning perspective, while solving such IP instances is NP-hard in the worst case, the specific structure of our consistency-based abduction problem often lends itself to relatively efficient resolution in practice. Our IP formulation is characterized by *binary decision variables*, a *linear objective function* (maximizing valid assignments), and a set of *linear constraints*. Many of these constraints exhibit a degree of *locality* (e.g., defining conflicts $\mathit{Con}_{\omega,(c,c')}$ based on assignments $A_{c,\omega}$ for the same object ω, or linking model-class considerations $X_{\omega,f,c}$ to overall assignments). This structured nature, which demonstrated efficient performance within our specific experimental configuration, often allows for practical solutions to be found within reasonable timeframes for problems of the scale explored in our experiments.

Heuristic Search (HS). Our Heuristic Search (HS) approach is detailed in Algorithm 1. Given the set of all raw model predictions P_{raw}, an inconsistency threshold δ, and a set of EDR ϵ values to evaluate, the algorithm greedily builds a hypothesis H (represented in the algorithm as the set of final predictions S_{final}) by iterating through model-class pairs (f, c). For each pair, it evaluates all ϵ, generating a filtered prediction set P_{new} (via an implicit $\mathit{GetFilteredPreds}(f, c, \epsilon, P_{raw})$ function). It selects the P_{new} that, when added to the current solution S_{final}, maximizes the size of the resulting candidate set $S_{candidate} = S_{final} \cup P_{new}$, while ensuring that $\mathit{ComputeInconsistency}(S_{candidate}) \leq \delta$. This chosen P_{new} for the current (f, c) pair is then added to S_{final}; this is analogous to deciding whether to add the atom $\mathit{accept}(f, c)$ to the hypothesis, based on whether this addition maximizes the number of final assignments without violating the inconsistency threshold δ. The HS algorithm has a running time in $O(|\mathcal{F}| \cdot |\mathcal{C}| \cdot |E_{set}|)$, where $|\mathcal{F}|$ and $|\mathcal{C}|$ are the numbers of models and classes, and $|E_{set}|$ is the number of evaluated ϵ values—that is, the cost is polynomial with respect to these key parameters of the input. This structured, greedy method efficiently selects predictions while managing inconsistencies, rendering it suitable for large-scale problem instances.

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 $\mathit{GetFilteredPreds}(f, c, \epsilon, P_{raw})$ and $\mathit{CalcIncon}(S)$.}
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**
    $P_{new} \leftarrow \mathit{GetFilteredPreds}(f, c, \epsilon, P_{raw})$
    $S_{cand} \leftarrow S_{final} \cup P_{new}$
    if $\mathit{CalcIncon}(S_{cand}) \leq \delta$ **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}

Tie-Breaker (TB) Mechanism. To ensure deterministic class assignment per object and resolve ambiguities where multiple labels remain valid after abduction, we apply a Tie-Breaker (TB) heuristic. For any object ω with several admissible labels, TB selects the pair (ω, c) proposed by the perception model with the highest confidence. This refinement yields the IP+TB and HS+TB variants in our experiments.

Experimental Setup

For evaluating the proposed approaches, we use an extended version of the Multiple Distribution Shift – Aerial (MDS-A) dataset (Ngu et al. 2025). The MDS-A dataset was generated using the AirSim (Shah et al. 2017) simulator and designed to analyze the impact of distributional shifts (caused by varying weather conditions) on object detection models in aerial imagery. The dataset consists of images captured in random positions under various weather conditions within a city environment alongside bounding boxes with one of four categories: *pedestrians*, *vehicles*, *nature*, and *construction* assigned to each bounding box. MDS-A contains six training sets, each simulating a specific weather condition: *rain*, *snow*, *fog*, *maple leaves*, *dust*, and a *baseline* (no weather effects), and three test sets containing a complex mix of weather conditions—Figure 2 illustrates an example of how the intensity distributions of weather effects vary.

In addition to the existing test sets, we add an augmented test suite comprising 12 new test sets, which introduce a

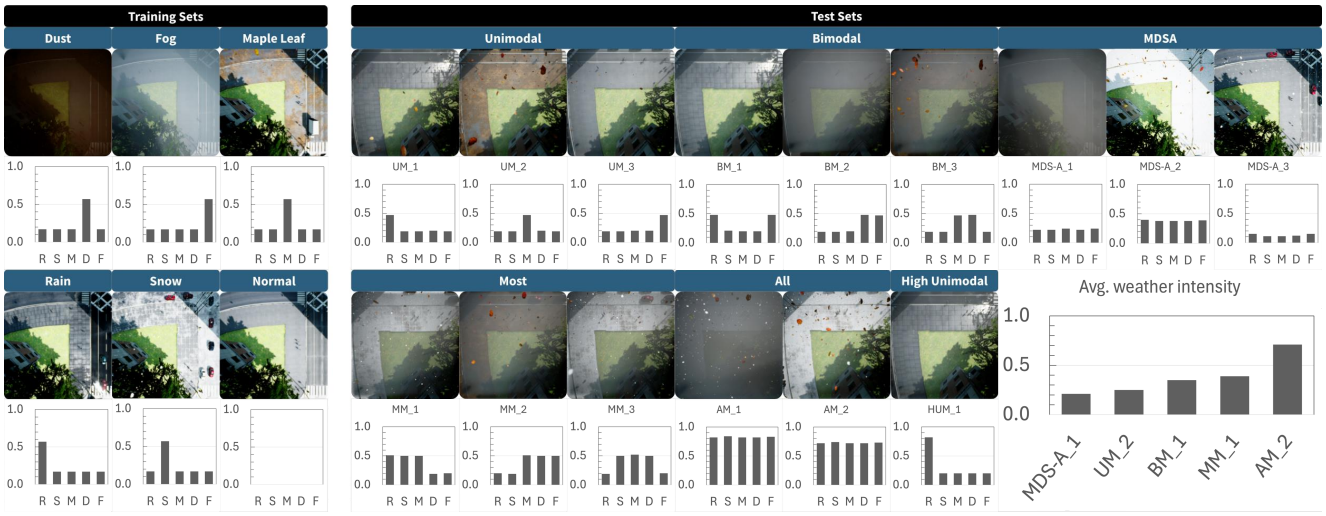

Figure 2: Images captured in the same position in AirSim under various weather conditions along with the distribution of weather conditions of the dataset that it represents. *Bottom right*: Histogram showing average intensity in selected datasets.

broader range of complex, mixed-weather conditions, further increasing the severity and diversity of distributional shifts. Each entry specifies the intensity level assigned to five different weather conditions, allowing for a systematic exploration of various distributional settings. The suite includes both homogeneous cases (where a single condition dominates) and heterogeneous scenarios involving multiple concurrent weather effects with varying intensities. The id of each dataset refers to the number of weather conditions fitted with the same intensity level: UM (unimodal), BM (bimodal), MM (most models), AM (all models), and HUM (uhigh unimodal). This design aims to evaluate the robustness of the proposed approaches under diverse and increasingly complex distribution shifts.

Six baseline object detection models were trained using the DeTR architecture (Carion et al. 2020) with a ResNet-50 backbone (He et al. 2015), each specialized in one of the weather-specific training datasets. The models were intentionally trained independently on their corresponding datasets to emphasize the effects of distributional shifts under specific weather conditions. This experimental setup provides a rigorous framework to assess the performance of both the integer programming and heuristic approaches under varying inconsistency thresholds and challenging weather-induced shifts.

Both the integer programming and heuristic approaches were evaluated using a range of inconsistency thresholds: $\delta \in [0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$. For the heuristic method, the trade-off parameter ϵ was also varied over the same range. Initially, all model-class pairs were included in the optimization process, allowing the heuristic approach to adaptively determine which combinations most effectively improve prediction performance while satisfying the inconsistency constraint. The integer programming formulation followed a similar configuration, incorporating δ as a hard constraint to limit the allowable inconsistencies. Each approach was run 50 times on each test sets.

To assess performance, we report the following metrics: *F1-score, accuracy*, and *execution time*; the latter is measured as a function of the number of objects present in each analyzed image. The results are compared against three baselines: the majority vote (MV) ensemble method (which selects the most common class prediction across models), the best-performing individual model, and the average performance of all models.

All experiments were conducted on a high-performance computing system, leveraging its advanced computational capabilities. Specifically, we used two configurations: (1) high-memory node: a Dell PowerEdge R6525 equipped with AMD EPYC 7713 64-Core processors and 2TB of RAM, and (2) GPU node: a Dell PowerEdge R7525 with AMD EPYC 7413 24-Core processors, 512GB of RAM, and three NVIDIA A30 GPUs. The logical deduction process, which applies the learned EDR to raw model outputs to identify predictions and errors, was implemented using PyReason. Subsequently, the IP solutions were implemented using the PuLP library for optimization.

Results and Discussion

We now discuss the results of the empirical evaluation of our proposed consistency-based abductive reasoning approaches for integrating predictions from multiple models in novel environments. We evaluate their effectiveness and ensure final consistency, including variants that incorporate a tie-break (TB) refinement (IP+TB and HS+TB). The key performance metrics, namely *F1-score* and *Accuracy*, are calculated across the suite of 15 test datasets, encompassing diverse distributional shifts as described in the experimental setup section. Figure 3 (left) summarizes these primary results, providing a comprehensive overview of how each method performs under various challenging conditions.

Overall Performance Analysis. Examining the F1-score and Accuracy metrics, the IP+TB method consistently

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%)

Figure 3: *Left:* Performance (F1 and Accuracy) across all test sets. Best values per test set in bold, the second-best are underlined. *Right:* Ablation Study – Performance without Tie-Breaker (TB). Values show F1-score or Accuracy for the method without TB, with the percentage difference relative to the corresponding + TB version (w.r.t. values on the left, shown in parentheses).

demonstrates superior performance, achieving the highest scores in all cases. For instance, on the challenging AM_1, AM_2, and HUM_1 test sets, characterized by significant distributional shifts, IP+TB yields notable improvements over the best-performing individual model, and significantly surpasses the standard Majority Vote (MV) ensemble, which often struggles in these complex scenarios (e.g., F1 scores of 0.21 vs. 0.18 vs. 0.05 on AM_1). The heuristic approach, HS+TB, also frequently outperforms the baselines, achieving for example the second-best F1 and Accuracy on the UM_1 test set (F1 0.61, Acc 0.44) and demonstrating strong performance on others like MDS-A_1. These results highlight the effectiveness of combining consistency-based abduction with a tie-breaking mechanism that selects the highest confidence prediction among inconsistent options. The contribution of this tie-breaker component is further analyzed in the ablation study presented below.

Environmental Analysis. To assess the robustness of our proposed methods against increasing environmental challenge, Figure 4 displays the F1-score for all 15 test datasets plotted against their average environmental intensity. Each point represents a unique test set, with different markers indicating the performance of IP+TB, HS+TB, and the baseline methods (Best Individual Model, Average Models, and Majority Vote). The average environmental intensity for each dataset (x-axis) is a normalized measure reflecting the severity of simulated conditions.

As observable in Figure 4, while there is a general trend of decreasing F1-scores for all methods as the average environmental intensity increases—indicating the inherent difficulty of operating in more severe novel environments—our IP+TB approach (represented by red diamonds) consistently achieves the highest F1-scores across the entire spectrum of intensities. In nearly all instances, IP+TB surpasses the Best Individual Model, and significantly outperforms the Average Models, Majority Vote, and our Heuristic Search (HS+TB)

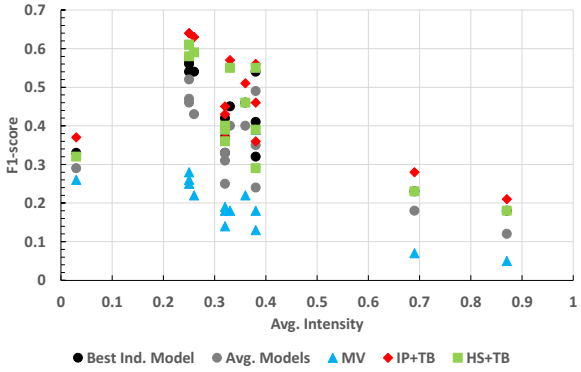

Figure 4: F1-scores for IP+TB and HS+TB vs. baselines (Best Ind. Model, Avg. Models, and Maj. Vote) across the 15 test datasets under increasing average weather intensity.

approach in terms of F1-score. This consistent superiority, irrespective of the environmental intensity level, underscores the effectiveness of IP+TB in robustly integrating model predictions and managing inconsistencies. For instance, even at higher intensity levels where all methods experience performance degradation, IP+TB maintains a clear advantage. The same analysis for Accuracy shows comparable trends, and is provided in the extended version.

These findings demonstrate that while environmental novelty impacts all approaches, our consistency-based abductive reasoning, particularly when implemented via Integer Programming (IP+TB), provides a robust performance advantage in F1-score over baseline methods and our heuristic alternative across a wide range of challenging conditions.

Tie Breaker Ablation. To evaluate the Tie-Breaker (TB) mechanism's role in ensuring consistent final predictions and its effect on performance, we carried out an ablation study comparing results with TB (Figure 3-left) and with-

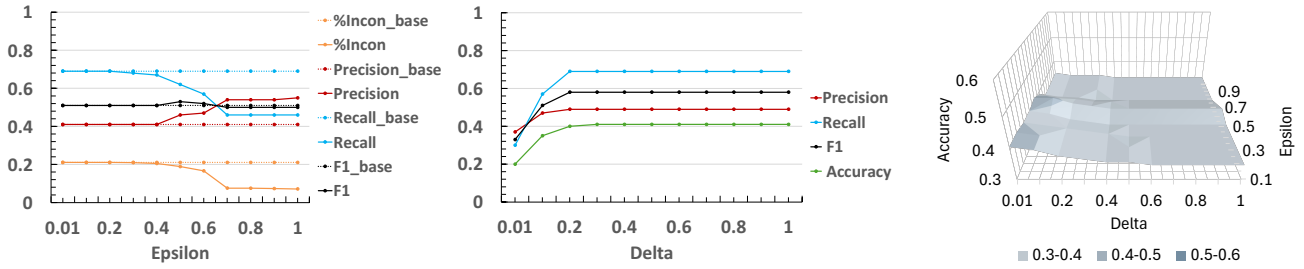

Figure 5: Hyperparameter sensitivity for the MDS-A_1 test set. *Left*: Performance metrics and inconsistency rates of the Error Detection Rule (EDR) stage across varying ϵ values. *Center*: Heuristic Search (HS+TB) performance, as a function of the δ inconsistency threshold. *Right*: IP+TB Accuracy depicted as a surface plot, varying δ and internal ϵ.

out TB (summarized in Figure 3-right, showing percentage differences). For the Integer Programming (IP) approach, removing the TB resulted in a 0.0% difference in performance across all metrics and datasets. This is a notable outcome, suggesting that for the chosen optimal δ values (typically in the range of 0.1 to 0.3, as per our sensitivity analysis), the IP optimization, in its pursuit of maximizing valid assignments inherently converged to solutions that were already fully consistent in terms of providing a single, unambiguous label per object, or where any minor ambiguities were resolvable by the TB without affecting the F1-score or Accuracy. Thus, the IP approach effectively achieved a high degree of output consistency by primarily leveraging other aspects of the formulation, such as the elimination of less reliable model-class pairs, potentially rendering the solution more consistent than strictly required by the explicit δ budget for conflicting assignments. In contrast, the Heuristic Search (HS) performance degraded consistently without the TB. As shown by the negative percentage differences in Figure 3 (right), the F1-score for HS dropped by 10% to over 17% across various datasets when the TB was removed, with similar reductions in Accuracy. We remind the reader that the tie-breaker heuristic selects the highest confidence prediction when an inconsistency is present.

EDR Rule Strictness. We also analyze the sensitivity of the initial Error Detection Rule (EDR) learning stage to its recall reduction threshold ϵ. This parameter controls the aggressiveness of filtering potentially erroneous predictions from individual models before the main abduction process. Figure 5 (left) illustrates the typical impact of varying ϵ on precision, recall, F1-score, and the inconsistency rate (i.e., conflicting predictions) for the MDS-A_1 test set.

As expected, increasing ϵ generally leads to higher precision but lower recall, demonstrating the inherent trade-off controlled by this parameter. Notably, this stricter filtering also tends to reduce the baseline level of inconsistency among the remaining predictions. This analysis highlights how tuning ϵ shapes the pool of candidate predictions subsequently processed by our IP+TB and HS+TB abduction methods aiming to maximize valid assignments under the global inconsistency constraint δ. Detailed sensitivity plots for all test sets are provided in extended version.

Hyperparameter Sensitivity. We analyzed the sensitivity of our approaches to the maximum inconsistency threshold hyperparameter δ, which applies to IP+TB and HS+TB, and the internal recall parameter ϵ used within the IP+TB optimization. Figure 5 (right) illustrates this analysis for the MDS-A_1 test set. The 3D plot of IP+TB show that the F1 score and accuracy reach their maximum value in the ranges of δ between 0.1 and 0.3, and for ϵ in 0.1 and 0.5. Similarly, the performance of HS+TB (center) stabilizes quickly as δ increases above a small initial value (e.g., 0.2), which is associated with the maximum level of inconsistency present in the initial test set (Figure 5, left); detailed sensitivity plots for all test sets are provided in the extended version.

Running Time Analysis and Complexity. A key practical difference between our approaches is computational cost. Our empirical results, illustrated in a running time plot available in the extended version, align with the theoretical complexities of the methods. As expected, our polynomial-time HS+TB approach is significantly faster than the exact IP+TB method. While the IP solver's running time is higher due to its exact nature, it remained tractable for the instances explored here. We observed that the average processing time per object for IP+TB did not increase steeply with more objects. We hypothesize that this is due to a combination of amortized solver overheads and structural dataset characteristics that allow for faster resolution. A detailed study of these issues is future work.

Conclusions and Future Work

We presented a comprehensive approach to address inconsistencies in model predictions via an abductive reasoning formulation. In future work, we plan to focus on the logic program for the deduction process to incorporate more sophisticated rules to infer alternative sets of assignments that address diverse inconsistency scenarios among model predictions. Such enhancements will be applied to new domains and datasets characterized by complex distributions. Further, we plan to refine our analysis by carrying out a more granular exploration of values of both the ϵ and δ parameters to evaluate the impact on solution robustness. Finally, optimizing runtime efficiency remains a priority to enable scalability and practical application in real-world scenarios.

Acknowledgments

This research was supported by the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement No. HR00112420370 (MCAI), the U.S. Army Combat Capabilities Development Command Army Research Laboratory under Support Agreement No. USMA 21050, and DARPA under Support Agreement No. USMA 23004. The views expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of the U.S. Military Academy, the U.S. Army, the U.S. Department of Defense, or the U.S. Government.

References

Aditya, D.; Mukherji, K.; Balasubramanian, S.; Chaudhary, A.; and Shakarian, P. 2023. PyReason: Software for Open World Temporal Logic.

Akyürek, E.; Damani, M.; Qiu, L.; Guo, H.; Kim, Y.; and Andreas, J. 2024. The Surprising Effectiveness of Test-Time Training for Abstract Reasoning. arXiv:2411.07279.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. arXiv:2005.12872.

Cheng, G.; and Han, J. 2016. A survey on object detection in optical remote sensing images. *ISPRS journal of photogrammetry and remote sensing*, 117: 11–28.

Dai, W.-Z.; Xu, Q.; Yu, Y.; and Zhou, Z.-H. 2019. Bridging machine learning and logical reasoning by abductive learning. *Advances in Neural Information Processing Systems*, 32.

Eiter, T.; and Gottlob, G. 1995. The complexity of logic-based abduction. *J. ACM*, 42(1): 3–42.

Evans, J. S. B. T.; and Stanovich, K. E. 2013. Dual-Process Theories of Higher Cognition: Advancing the Debate. *Perspectives on Psychological Science*, 8(3): 223–241. PMID: 26172965.

Han, X.; Zhang, Z.; Ding, N.; Gu, Y.; Liu, X.; Huo, Y.; Qiu, J.; Yao, Y.; Zhang, A.; Zhang, L.; et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2: 225–250.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Huang, Y.-X.; Dai, W.-Z.; Jiang, Y.; and Zhou, Z.-H. 2023. Enabling Knowledge Refinement upon New Concepts in Abductive Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7): 7928–7935.

Johnson, S. G. B.; Karimi, A.-H.; Bengio, Y.; Chater, N.; Gerstenberg, T.; Larson, K.; Levine, S.; Mitchell, M.; Rahwan, I.; Schölkopf, B.; and Grossmann, I. 2024. Imagining and building wise machines: The centrality of AI Metacognition. arXiv:2411.02478.

Kahneman, D. 2012. *Thinking, Fast and Slow*. London: Penguin. ISBN 9780141033570, 0141033576.

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 *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 3842–3846.

Ngu, N.; Taparia, A.; Simari, G. I.; Leiva, M.; Corcoran, J.; Senanayake, R.; Shakarian, P.; and Bastian, N. D. 2025. Multiple Distribution Shift – Aerial (MDS-A): A Dataset for Test-Time Error Detection and Model Adaptation. In *AAAI Spring Symposium*.

Parisi, S.; Rajeswaran, A.; Purushwalkam, S.; and Gupta, A. 2022. The unsurprising effectiveness of pre-trained vision models for control. In *international conference on machine learning*, 17359–17371. PMLR.

Peng, Y.; and Reggia, J. A. 1990. *Abductive inference models for diagnostic problem-solving*. Springer-Verlag.

Poole, D. 1989. Normality and faults in logic-based diagnosis. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'89, 1304–1310. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2017. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*.

Sun, Y.; Wang, X.; Zhuang, L.; Miller, J.; Hardt, M.; and Efros, A. A. 2020. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *ICML*.

Sutor, P.; Yuan, D.; Summers-Stay, D.; Fermuller, C.; and Aloimonos, Y. 2022. Gluing neural networks symbolically through hyperdimensional computing. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–10. IEEE.

Thompson, V. A. 2009. Dual-process theories: A metacognitive perspective. In Evans, J.; and Frankish, K., eds., *In Two Minds: Dual Processes and Beyond*, 171–195. Oxford University Press.

Wason, P.; and Evans, J. 1974. Dual processes in reasoning? *Cognition*, 3(2): 141–154.

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 Besold, T. R.; d'Avila Garcez, A.; Jiménez-Ruiz, E.; Confalonieri, R.; Madhyastha, P.; and Wagner, B., eds., *Neural-Symbolic Learning and Reasoning - 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, 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.