Concise and Logically Consistent Conformal Sets for Neuro-Symbolic Concept-Based Models

Samuele Bortolotti[*]
DISI, University of Trento
samuele.bortolotti@unitn.it

Emanuele Marconato[†]
University of Copenhagen
emanuele.marconato@math.ku.dk

Andrea Pugnana
DISI, University of Trento
andrea.pugnana@unitn.it

Andrea Passerini
DISI, University of Trento
andrea.passerini@unitn.it

Stefano Teso
DISI & CiMeC, University of Trento
stefano.teso@unitn.it

Abstract

Neuro-Symbolic Concept-based Models (NeSy-CBMs) are a family of architectures that integrate neural networks with symbolic reasoning for enhanced reliability in high-stakes applications. They work by first extracting high-level concepts from the input and then inferring a task label from these compatibly with given logical constraints. Yet, their label and concept predictions can be overconfident, making it difficult for stakeholders to gauge when the model decisions can be trusted. We address this issue by integrating ideas from Conformal Prediction (CP), a framework providing rigorous, distribution-free coverage guarantees. We formalize three desiderata – consistency, coverage, and conciseness – that any conformal method for NeSy-CBMs should satisfy, and show that existing approaches fall short of at least one. We then introduce COCOCO, a post-hoc framework that conformalizes concepts and labels jointly and reconciles them via *a single deduction–abduction revision step*. COCOCO satisfies all three desiderata, retains distribution-free coverage, is robust to imperfect knowledge and supports user-specified size budgets. Our experiments on 9 data sets highlight how COCOCO compares favorably against competitors and natural baselines in terms of performance and set size.

1 Introduction

Neuro-Symbolic AI (NeSy) architectures integrate neural networks with symbolic reasoning, and as such are prime candidates for designing reliable-by-construction AI systems [1, 2]. We focus on Neuro-Symbolic Concept-based Models (***NeSy-CBMs***), a general family of NeSy classifiers that distinguish between the neural and reasoning steps at the architectural level [3, 4, 5, 6]. Given a low-level input, a NeSy-CBM first distills it into a set of high-level ***concepts*** using a neural backbone, and then applies differentiable logic reasoning to infer a ***task prediction*** that complies to constraints we wish to hold. E.g., if tasked with computing the sum of two MNIST digits [7], it would first predict the corresponding digits (the concepts) and then apply the rule of addition (the constraint) [8].

Despite being geared for reliability, NeSy-CBMs can still be overconfident [9, 10], making their predictive uncertainty unreliable, see Fig. 1 (Left). This hinders their application to high-stakes tasks,

[*]Corresponding author.
[†]Work done while at DISI, University of Trento.

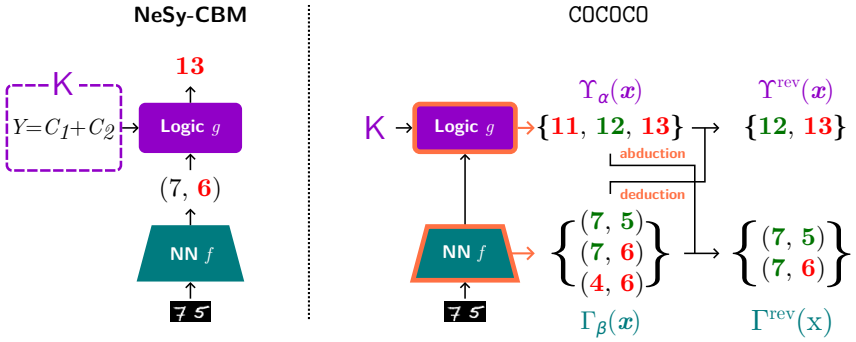


Figure 1: **Left**: the task and concept predictions of NeSy-CBMs can make confident mistakes (e.g., 6 and 13, in **red**), making it difficult to gauge the (un)reliability of their predictions. **Right**: COCOCO enables any NeSy-CBM to output task- and concept-level prediction sets with coverage guarantees for any desired size, while ensuring their mutual consistency via an abduction-deduction refinement step. E.g., in the picture the task label 11 (and the concept label $(4, 6)$) is removed because it is not logically valid, leaving the correct outputs (in **green**) untouched with high probability.

where one often requires machine learning models to be aware of their own mistakes [11]. Moreover, even if a few NeSy approaches correct overconfidence in a principled manner (e.g., [9, 10]), their predicted probabilities can be difficult to grasp for practitioners and end-users [12].

We address these limitations with ***conformal prediction*** (CP), a general framework for constructing predictors that, rather than producing a point-wise probability estimate, output a set of plausible alternatives with finite-sample and distribution-free ***coverage guarantees*** [13]. Specifically, we distill three desiderata, i.e., ***consistency***, ***coverage***, and ***conciseness***, that conformal NeSy methods ought to satisfy, and introduce COCOCO (COnformality, COnsistency, COnciseness), a novel and general conformal approach that can turn any NeSy-CBM into a full-fledged conformal NeSy predictor satisfying these desiderata. In a nutshell, COCOCO implements conformal prediction at the concept and task level and combines them through a post-processing step that ensures the resulting prediction sets are logically consistent with each other. Moreover, we show how to leverage *e-values* [14, 15, 16, 17] to ensure the prediction sets [18, 19, 20, 21] attain maximal coverage for any given size budget while remaining consistent with prior knowledge. Our extensive experiments on eight image and text datasets and two NeSy CBMs show that COCOCO achieves high coverage while keeping the size of the prediction sets under control, even in the presence of noisy, incorrect, or underspecified knowledge.

Our Contributions. We *i*) formulate three natural desiderata, i.e., consistency, coverage and conciseness, that any conformal method for NeSy-CBMs ought to satisfy; *ii*) design COCOCO, a post-hoc framework that applies CP jointly at the concept and task levels and reconciles the two via a *single deductionabduction revision step*; *iii*) prove COCOCO satisfies all three desiderata: consistency by construction, distribution-free coverage whose lower bound degrades linearly under imperfect knowledge, and optimal conciseness; *iv*) show that COCOCO enables user-specified size budgets with guarantees; *v*) evaluate COCOCO on 9 NeSy data sets with different characteristics, highlighting the promise of COCOCO in terms of consistency, coverage and set size compared to natural competitors.

2 Preliminaries

Notation. We write scalar constants x in lower-case, random variables X in upper case, vector constants and variables in bold typeface ($\mathbf{x}$ and $\mathbf{X}$), and sets $\mathcal{X}$, Υ in upper-case italics/greek. Given a function $f : \mathcal{X} \to \mathcal{Y}$, we indicate the image of a set $\mathcal{X}' \subseteq \mathcal{X}$ as $f(\mathcal{X}')$ and the pre-image of a set $\mathcal{Y}' \subseteq \mathcal{Y}$ as $f^{-1}(\mathcal{Y}')$. We write $\Delta_{\mathcal{S}}$ for the simplex of distributions over $\mathcal{S}$ and $[n]$ for $\{1, \dots n\}$.

Neuro-Symbolic AI. NeSy approaches differ wildly in terms of aim and architecture [1, 2]. We focus on Neuro-Symbolic Concept-based Models (***NeSy-CBMs*** for short), a family of classifiers designed for reliability [3, 4, 6]. All NeSy-CBMs extract intermediate, high-level *concepts*[3] from the input and infer predictions by *reasoning* over the concepts compatibly with user-specified prior

[3]Also known as *neural predicates* [22].

knowledge. The latter is encoded as a logic formula K. A NeSy-CBM [23, 24, 6, 5] maps inputs $\mathbf{x} \in \mathcal{X}$ to k concepts $\mathbf{c} = (c_1, \ldots, c_k) \in \mathcal{C}$ and then to labels $\mathbf{y} \in \mathcal{Y}$ using two components. The ***concept extractor*** $f : \mathcal{X} \to \Delta_{\mathcal{C}}$ is a neural network mapping the input $\mathbf{x} \in \mathcal{X}$ to a distribution over concepts $p_\theta(\mathbf{C} \mid \mathbf{x})$. The ***inference layer*** $g : \mathcal{C} \to \Delta_{\mathcal{Y}}$ maps concept vectors to a distribution over labels $p(\mathbf{Y} \mid \mathbf{c}, \mathsf{K})$ according to given and fixed prior knowledge K. With slight abuse of notation, we write $(g \circ f)$ to indicate the conditional distribution $p(\mathbf{Y} \mid \mathbf{x}; \mathsf{K})$ obtained by passing the concept probabilities output by f to the inference layer g. In practice, how these are combined depends on the specific NeSy-CBM [6]; we describe the two we use experimentally below, after a running example.

Example 1. *In* `MN-Add` *[8], the goal is to predict the sum of two* `MNIST` *[7] digits. Here, the concepts* $\mathbf{c} \in \mathcal{C} := \{0, \ldots, 9\}^2$ *represent the two digits and the label* $y \in \mathcal{Y} := \{0, \ldots, 18\}$ *is their sum. The knowledge* $\mathsf{K} = (y = c_1 + c_2)$ *specifies*[4] *that* y *should be the arithmetic sum of the two concepts. E.g., given an input* $\mathbf{x} =$  *a NeSy-CBM applies* f *to infer a distribution over digits and then* g *to obtain one over sums, typically assigning sums that violate* K *(e.g.,* $\mathbf{c} = (7, 5)$*,* $y = 11$*) lower probability.*

We focus on two representative NeSy-CBMs. ***DeepProbLog*** (`DPL`) [8] models the inference layer g using probabilistic logic semantics [25]. This guarantees that the predictive distribution $p_\theta(\mathbf{Y} \mid \mathbf{c})$ always allocates zero mass to labels that violate the prior knowledge K, e.g., incorrect sums. `DPL` infers hard predictions by solving the MAP problem $\operatorname{argmax}_{\mathbf{y}} p_\theta(\mathbf{y} \mid \mathbf{c})$, using knowledge compilation [26, 27] for efficiency. ***LogicTensorNetworks*** (`LTN`) [28, 29] implement the inference layer using fuzzy logic [30, 31], scoring labels by their fuzzy degree of satisfaction. Both architectures learn the parameters θ via end-to-end training, i.e., given a training set $\mathcal{D}_{\text{tr}} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}$ sampled from a ground-truth data distribution $p^*(\mathbf{X}, \mathbf{C}, \mathbf{Y})$, `DPL` maximizes the likelihood of the training data, while `LTN` maximizes the fuzzy degree of satisfaction of the prior knowledge K evaluated over the ground-truth training data $\mathbf{y}_i^*$. When concept annotations are available, they can be leveraged during training by combining the original loss with an average cross-entropy term over the concepts.

Concept independence and reasoning shortcuts. We highlight two key points. First, it is common practice to implement the concept extractor f by stacking per-concept neural classifiers [32, 8, 28, 6]. This means the distribution $p_\theta(\mathbf{C} \mid \mathbf{x})$ factorizes as $\prod_i p_\theta(C_i \mid \mathbf{x})$, that is, the predicted concepts are ***conditionally independent***: $C_i \perp\!\!\!\perp C_j \mid \mathbf{X}$ for all $i \neq j$ [10, 33]. Second, typically ground-truth concept annotations $\mathbf{c}_i^*$ are not used during training. As a result, NeSy-CBMs often solve the task by learning concepts that do not match human semantics, a phenomenon known as ***reasoning shortcuts*** (RSs) [24, 6]. Models affected by such RSs output confident but wrong concept predictions even when their label accuracy is high, and thus suffer from impaired interpretability and reliability [9].

Conformal Prediction (CP) is a general framework for constructing set-valued predictions with *finite-sample* and *distribution-free coverage guarantees* [13]. Such guarantees are especially relevant in decision support systems [34, 35, 36], as well as in high-stakes applications where human judgment is needed [37, 38] or where statistical reliability assurances are required [39].

Given a predictor $p_\theta(\mathbf{Y} \mid \mathbf{X})$ and a calibration dataset $\mathcal{D}_{\text{cal}} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^n$ of exchangeable examples, CP maps a test input $\mathbf{x}_i \in \mathcal{X}$ to a prediction set $\Upsilon_\alpha(\mathbf{x}_i) \subseteq \mathcal{Y}$ such that:

$$\Pr(\mathbf{y}_i^* \in \Upsilon_\alpha(\mathbf{x}_i)) \geq 1 - \alpha \tag{1}$$

CP relies on a nonconformity score $s(\mathbf{x}, \mathbf{y})$ that quantifies how atypical a candidate output $\mathbf{y}$ is for an input $\mathbf{x}$, e.g., a typical choice being $1 - p(\mathbf{y}^* \mid \mathbf{x})$ [40]. Let $s_i = s(\mathbf{x}_i, \mathbf{y}_i^*)$ for $i \in [n]$ be the nonconformity score of each calibration example, and let $q_{1-\alpha}$ be the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of $\{s_i\}_{i=1}^n$. The factor $(n+1)/n$ is the standard finite-sample correction that treats the test point as exchangeable with the calibration set, and it ensures the exact finite-sample guarantee Eq. (1) [40, 41]. Then, the conformal prediction set for a test input $\mathbf{x}_i$ can be built as:

$$\Upsilon_\alpha(\mathbf{x}_i) = \{\mathbf{y} \in \mathcal{Y} : s(\mathbf{x}_i, \mathbf{y}) \leq q_{1-\alpha}\} \tag{2}$$

3 Building Blocks of Conformal NeSy-CBMs

NeSy-CBMs are geared for reliability in high-stakes applications [6, 42, 43], where it is essential that predictions satisfy domain-specific constraints K [8, 44, 45, 46, 47, 48, 49, 50]. Yet, their predictive uncertainty is both unreliable and difficult to understand. In fact, they are often miscalibrated [9, 10, 11], meaning stakeholders cannot use their uncertainty to assess whether to trust their predictions

[4] This constraint can be naturally written in propositional and first-order logic [8].

[9, 10]. Moreover, predictive uncertainty is difficult for users to understand in the first place [12]. Both issues are problematic, especially for NeSy-CBMs affected by reasoning shortcuts, in which case users ought to be made aware that the learned concepts are not human aligned, yet these are often predicted wrong with confidence [6, 24].

This calls for a different strategy that can simultaneously handle uncertainty while providing a user-friendly interface for reliability assessment. This is where CP comes into play, as it offers a principled, distribution-free remedy that does not involve communicating probabilities to users. We outline the benefits we wish to obtain from such a combination by formalizing three desiderata that *any* NeSy conformal approach ought to satisfy:

- **D1** *Consistency*: predicted concepts and labels must be logically consistent with the knowledge K.
- **D2** *Coverage*: prediction sets must enjoy distribution-free, finite-sample marginal coverage guarantees on *both* labels and concepts, so as to ensure stakeholders can trust the prediction sets.
- **D3** *Conciseness*: prediction sets should be as small as possible *without sacrificing* **D1**–**D2**, i.e., they should retain only alternatives that are supported both by the conformal sets and by the knowledge K. Indeed, large sets are uninterpretable and operationally useless [35].

In Section 4, we will introduce `COCOCO`, a novel NeSy CP strategy that *does* complies with **D1**–**D3**. Yet, integrating CP into NeSy-CBMs without compromising any desideratum is non-trivial. In fact, as we will show in Section 4.1, any "***one-sided***" CP strategy that independently targets either the concept extractor or the inference layer is insufficient to satisfy all desiderata.

Inference layer operators. Before proceeding, we introduce two operators that all methods we will discuss build on, together with their coverage properties. We denote with $g^{\dagger}$ the argmax of the inference layer g and its preimage as $g^{-\dagger}$, that is:

$$g^{\dagger}(\mathbf{c}) := \mathrm{argmax}_{\mathbf{y}\in\mathcal{Y}}\, g(\mathbf{c}) \subseteq \mathcal{Y}, \qquad g^{-\dagger}(\mathbf{y}) := \{\mathbf{c}\in\mathcal{C} : \mathbf{y}\in g^{\dagger}(\mathbf{c})\} \tag{3}$$

Intuitively, $g^{\dagger}$ ***deduces*** a (subset of) label(s) from given concepts, while $g^{-\dagger}$ ***abduces*** a (subset of) concepts from given label(s). Two properties of $g^{\dagger}$ play a central role in our coverage guarantees:

Definition 3.1 (Deductive & Abductive Soundness)**.** Let $p^*(\mathbf{x},\mathbf{c},\mathbf{y})$ be the joint data distribution. A NeSy-CBM (f, g) satisfies *deductive soundness* if, for every $(\mathbf{c}^*,\mathbf{y}^*)\in\mathrm{supp}(p^*(\mathbf{x},\mathbf{c},\mathbf{y}))$, $g^{\dagger}(\mathbf{c}^*) = \mathbf{y}^*$. It satisfies *abductive soundness* if, for every $(\mathbf{c}^*,\mathbf{y}^*)\in\mathrm{supp}(p^*(\mathbf{x},\mathbf{c},\mathbf{y}))$, $\mathbf{c}^*\in g^{-\dagger}(\mathbf{y}^*)$.

Deductive and abductive soundness depend on both the prior knowledge K and on how the inference layer g implements it. In particular, they hold when: (*i*) the knowledge is ***correct and complete***, meaning that it allows the ground-truth labels and concepts to be retrieved from one another without errors (as in `MN-Add`, see Example 1), and (*ii*) the inference layer g ***preserves this property***. The latter holds for both `DPL` and `LTN`[5], and in general for most NeSy-CBMs [45, 51, 47]. However, these properties may not hold when the knowledge is incomplete or erroneous, as for some of the datasets used in our experiments (Section 6), or when the inference layer is learned from data [52, 53, 54, 55].

Label and concept sets via standard CP. Given a calibration set $\mathcal{D}_{\mathbf{Y}} = \{(\mathbf{x}_i,\mathbf{y}_i^*)\}$ of exchangeable examples and a miscoverage level $0<\alpha<1$, we can obtain a ***label set*** $\Upsilon_{\alpha}(\mathbf{x})\subseteq\mathcal{Y}$ satisfying:

$$\Pr(\mathbf{y}^*\in\Upsilon_{\alpha}(\mathbf{x}))\geq 1-\alpha \tag{4}$$

simply by applying CP (Eq. (2)) to the NeSy-CBM label distribution $p_{\theta}(\mathbf{Y}\mid\mathbf{x},\mathsf{K})$. Doing the same for the concept extractor f is not as straightforward, in that doing so requires concept annotations $\mathbf{c}^*$ on the entire concept vector, which are seldom available [56, 57]. We work around this issue by exploiting the conditional independence of concepts. Specifically, for each concept C_j we collect a *per-concept* calibration set $\mathcal{D}_{C_j} = \{(\mathbf{x}_i, c_{ij}^*)\}$, build a conformal set $\Gamma^j_{\beta_j}(\mathbf{x})$ with miscoverage level β_j, and use the latter to construct a ***product concept set*** as

$$\mathbf{\Gamma}^{\times}(\mathbf{x}) = \bigtimes_{j=1}^{k}\Gamma^j_{\beta_j}(\mathbf{x}) = \Gamma^1_{\beta_1}(\mathbf{x})\times\cdots\times\Gamma^k_{\beta_k}(\mathbf{x}). \tag{5}$$

Notably, the product set inherits coverage guarantees, as shown by the following Proposition:

[5]We specifically refer to versions of `LTN` for which this property holds, namely, those that use t-(co)norms [31].

Proposition 3.1 (Coverage of $\mathbf{\Gamma}^{\times}(\mathbf{x})$)**.** *For any NeSy-CBM (f, g) and $0 < \beta_j < 1$, let $\beta = \sum_{j=1}^{k} \beta_j$. If each $\Gamma^j_{\beta_j}(\mathbf{x})$ satisfies $\Pr(\mathbf{c}^*_j \in \Gamma^j_{\beta_j}(\mathbf{x})) \geq 1 - \beta_j$, then for every $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*) \in \mathrm{supp}(p^*(\mathbf{x}, \mathbf{c}, \mathbf{y}))$:*

$$\Pr(\mathbf{c}^* \in \mathbf{\Gamma}^{\times}(\mathbf{x})) \geq 1 - \beta \tag{6}$$

All proofs are reported in Section B due to space constraints. Eq. (6) implies that, in order to ensure $\mathbf{\Gamma}^{\times}(\mathbf{x})$ meets joint coverage $1 - \beta$, it is sufficient to choose $\beta_j = \beta/k$ for all j. One downside is that, as k grows, this allocates vanishing miscoverage to each per-concept set $\Gamma^j_{\beta_j}(\mathbf{x})$, inflating them and $\mathbf{\Gamma}^{\times}(\mathbf{x})$ as a result. We will see in Section 4 that e-values avoid this issue entirely.

In what follows, we use $\Gamma_\beta(\mathbf{x})$ to denote any concept set satisfying $\Pr(\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})) \geq 1 - \beta$. In our default construction, this is $\mathbf{\Gamma}^{\times}(\mathbf{x})$, but our method applies to any concept set with this guarantee, including those built using a calibration set over the entire concept vector $\mathbf{c}^*$.

Abductive and deductive operators. With these notions in place, we can map between $\Upsilon_\alpha(\mathbf{x})$ and $\Gamma_\beta(\mathbf{x})$ using the inference layer operators introduced in Eq. (3), that is:

$$\Gamma^{\mathsf{ab}}(\mathbf{x}) := g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})), \qquad \Upsilon^{\mathsf{de}}(\mathbf{x}) := g^{\dagger}(\Gamma_\beta(\mathbf{x})). \tag{7}$$

$\Gamma^{\mathsf{ab}}(\mathbf{x})$ collects all concept vectors that logically entail at least one label in $\Upsilon_\alpha(\mathbf{x})$ via *abduction* [58, 59, 60]; $\Upsilon^{\mathsf{de}}(\mathbf{x})$ collects all labels derivable from concepts in $\Gamma_\beta(\mathbf{x})$ via *deduction* [57]. These operators transfer coverage guarantees across levels, as shown by the following propositions:

Proposition 3.2 (Coverage of $\Gamma^{\mathsf{ab}}(\mathbf{x})$)**.** *For any NeSy CBM (f, g), let $0 < \alpha < 1$. If $\Upsilon_\alpha(\mathbf{x})$ is a label conformal set Eq. (4) and $\Gamma^{\mathsf{ab}}(\mathbf{x})$ is the set constructed as Eq. (7), then for every $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*) \in \mathrm{supp}(p^*(\mathbf{x}, \mathbf{c}, \mathbf{y}))$:*

$$(1 - \alpha)\delta_{\mathrm{ab}} \leq \Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})) \leq \alpha + \delta_{\mathrm{ab}} \tag{8}$$

where $\delta_{\mathrm{ab}} := \Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) \mid \mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x}))$ is the probability that abduction recovers the ground-truth concepts $\mathbf{c}^*$ provided the label set $\Upsilon_\alpha(\mathbf{x})$ contains the correct label $\mathbf{y}^*$. Moreover, if the NeSy-CBM satisfies abductive soundness, the lower bound tightens to $\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})) \geq 1 - \alpha$.

Proposition 3.3 (Coverage of $\Upsilon^{\mathsf{de}}(\mathbf{x})$)**.** *For any NeSy CBM (f, g), let $0 < \beta < 1$. If $\Gamma_\beta(\mathbf{x})$ is a concept conformal set Eq. (6) and $\Upsilon^{\mathsf{de}}(\mathbf{x})$ is the set constructed as Eq. (7). Then, for every $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*) \in \mathrm{supp}(p^*(\mathbf{x}, \mathbf{c}, \mathbf{y}))$:*

$$(1 - \beta)\delta_{\mathrm{de}} \leq \Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})) \leq \beta + \delta_{\mathrm{de}} \tag{9}$$

where $\delta_{\mathrm{de}} := \Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x}) \mid \mathbf{c}^* \in \Gamma_\beta(\mathbf{x}))$ measures how likely it is that deduction recovers the true label $\mathbf{y}^*$ given that the concept set $\Gamma_\beta(\mathbf{x})$ contains the correct concepts $\mathbf{c}^*$. Moreover, under deductive soundness, the bound tightens to $\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})) \geq 1 - \beta$.

Next, we show that $\Gamma^{\mathsf{ab}}(\mathbf{x})$ or $\Upsilon^{\mathsf{de}}(\mathbf{x})$, while enjoying coverage guarantees, are insufficient to construct a NeSy CP strategy satisfying **D1**–**D3** when used in isolation, while combining them through a revision step, as done by COCOCO, does.

4 Consistency, Conformality and Conciseness

We begin by introducing COCOCO (COnformality, COnsistency, COnciseness), a post-hoc framework for turning any NeSy-CBM into a conformal predictor satisfying all three desiderata **D1**–**D3**. The idea is as follows: starting from a label set $\Upsilon_\alpha(\mathbf{x})$ and a concept set $\Gamma_\beta(\mathbf{x})$, we make them mutually consistent by intersecting the label set (resp. concept set) with $g^{\dagger}$ applied to the concept set (resp. $g^{-\dagger}$ to the label set). We show that this requires only one revision step to converge.

Joint Revision Step. Given $\Upsilon_\alpha(\mathbf{x})$ satisfying Eq. (4) and $\Gamma_\beta(\mathbf{x})$ satisfying Eq. (6), COCOCO computes:

$$\Upsilon^{\mathsf{rev}}(\mathbf{x}) := \Upsilon_\alpha(\mathbf{x}) \cap g^{\dagger}(\Gamma_\beta(\mathbf{x})), \qquad \Gamma^{\mathsf{rev}}(\mathbf{x}) := \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})). \tag{10}$$

In other words, $\Gamma^{\mathsf{rev}}(\mathbf{x})$ retains only the concepts in $\Gamma_\beta(\mathbf{x})$ that logically entail some label in $\Upsilon_\alpha(\mathbf{x})$, and $\Upsilon^{\mathsf{rev}}(\mathbf{x})$ retains only the labels in $\Upsilon_\alpha(\mathbf{x})$ that are entailed by some concept in $\Gamma_\beta(\mathbf{x})$. We highlight two key structural properties: a *single* application of Eq. (10) is enough to reconcile the two sets, and the resulting pair is *maximally informative* among all consistent revisions of $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$. Both properties are made formal below.

Proposition 4.1 (One-step consistency and optimality of COCOCO)**.** *Let* $R(\Gamma, \Upsilon) := \big(\Gamma \cap g^{-\dagger}(\Upsilon),\ \Upsilon \cap g^{\dagger}(\Gamma)\big)$ *denote the joint revision operator. Then, for any* $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$*:*

1. (One-step fixed point) $R\big(R(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))\big) = R(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$*, so iterating the revision is unnecessary;*
2. (Optimality) $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x})) = R(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$ *is the* largest *pair* $(\Gamma', \Upsilon') \subseteq \Gamma_\beta(\mathbf{x}) \times \Upsilon_\alpha(\mathbf{x})$ *such that* $\Gamma' \subseteq g^{-\dagger}(\Upsilon')$ *and* $\Upsilon' \subseteq g^{\dagger}(\Gamma')$*, i.e., each label* $\mathbf{y} \in \Upsilon'$ *is represented by at least a concept* $\mathbf{c} \in \Gamma'$*, and vice-versa, according to the inference layer* g*, making it the largest mutually consistent subset of* $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$.

Notably, Proposition 4.1 shows that a single deduction-abduction pass is sufficient to simultaneously achieve consistency (**D1**) and optimal conciseness (**D3**) given $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$. Note that "largest" in Proposition 4.1 and "as small as possible" in **D3** are two sides of the same coin: COCOCO removes *all and only* the elements of $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$ that are mutually inconsistent under K, i.e., it performs the *minimal deletion* required to attain **D1**. Any further shrinkage would discard elements that are still supported by both the conformal sets and the knowledge hindering **D2**. Next, we show that this joint revision preserves coverage guarantees (**D2**).

Proposition 4.2 (Marginal coverage of COCOCO)**.** *For any NeSy CBM* (f, g)*, let* $0 < \alpha < 1$ *and* $0 < \beta < 1$*. If* $\Gamma_\beta(\mathbf{x})$ *is a concept conformal set Eq. (6) and* $\Upsilon_\alpha(\mathbf{x})$ *is a label conformal set Eq. (4), then, for every* $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*) \in \mathrm{supp}(p^*(\mathbf{x}, \mathbf{c}, \mathbf{y}))$*, we have*

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) \geq (1-\alpha)\delta_{\mathrm{ab}} - \beta + \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x})), \tag{11}$$

$$\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{rev}}(\mathbf{x})) \geq (1-\beta)\delta_{\mathrm{de}} - \alpha + \Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\mathsf{de}}(\mathbf{x}))\,, \tag{12}$$

where $\Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x}))$ *and* $\Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\mathsf{de}}(\mathbf{x}))$ *are the probabilities that the ground-truths are not covered by* neither *the concept/label conformal set* nor *the abductive/deductive conformal set. Moreover, if the NeSy predictor satisfies both deductive soundness and abductive soundness (Definition 3.1), the bounds tighten, with* $\delta_{\mathrm{ab}} = 1 = \delta_{\mathrm{de}}$.

Under soundness ($\delta_{\mathrm{ab}} = \delta_{\mathrm{de}} = 1$), COCOCO retains coverage at least $1 - \alpha - \beta$ on both sides, plus a non-negative joint-failure correction, i.e., $\Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x}))$ or $\Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\mathsf{de}}(\mathbf{x}))$. This correction term is empirically estimable from a held-out set and is small in practice (≤ 0.08 across our datasets, see Section E). When soundness degrades, e.g., on incomplete-knowledge datasets like RIV10 the bounds worsen according to the quality of the inference layer (δ).

User-specified size budgets. Standard CP fixes α, β a priori and yields the corresponding set sizes. In practice, specifically in decision-support systems [34, 38, 35], it is often more useful to fix a *size budget* ("at most 5 candidate concepts") and report the corresponding coverage, thereby avoiding overwhelming practitioners with a range of choices. As shown by Gauthier et al. [20], this can be achieved by replacing standard CP with *conformal e-predictions* [18, 19, 20, 21], which use *e-values* [14, 15, 16, 17] in place of quantiles. E-values bring two benefits to COCOCO. First, we can provably build a conformal set over concepts *while avoiding Bonferroni's correction* (Corollary B.4), meaning that the resulting concept-set guarantee does *not* degrade with the number of concepts k (Proposition B.9). Second, e-values allow data-dependent miscoverage levels $\tilde{\alpha}, \tilde{\beta}$ to be chosen *after* observing calibration scores, in particular to satisfy user-specified size budgets while retaining a coverage guarantee *in expectation* [17, 18]. We denote this variant COCOCO*, and prove its joint-coverage guarantee in Section B.10. Details about e-values and COCOCO* are left to Section C.3.

Runtime. A key advantage of COCOCO is that the joint revision operator reaches a fixed point in a single step (Proposition 4.1), so the additional overhead amounts to computing the abductive and deductive images once, i.e., $\Gamma^{\mathsf{ab}}(\mathbf{x}) = g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$ and $\Upsilon^{\mathsf{de}}(\mathbf{x}) = g^{\dagger}(\Gamma_\beta(\mathbf{x}))$. The main computational bottleneck lies in constructing the concept set $\mathbf{\Gamma}^{\times}(\mathbf{x})$, which scales exponentially in the number of concepts k due to the cross-product structure. This issue is mitigated when full concept annotations are available or if COCOCO* is employed due to *e-values*, in which cases $\Gamma_\beta(\mathbf{x})$ can be obtained without carrying out any set product. In practice, we find that the runtime of COCOCO does not exceed $1.6\times$ that of standard DPL and LTN, which is acceptable in high-stakes settings where reliability is crucial. A detailed runtime analysis is deferred to Section E.7 due to space constraints.

Table 1: **Summary of guarantees** provided by different NeSy conformal strategies.

Method	Consistency (**D1**)		Coverage (**D2**)		Conciseness (**D3**)	
	Task	Concepts	Task $\mathbf{Y}$	Concepts $\mathbf{C}$	Task $\mathbf{Y}$	Concepts $\mathbf{C}$
Task Only (`TO`) [61]	✗	–	✓ (4)	✗	✗	–
Task plus Abduction (`TAb`)	✓	✓	✓ (4)	✓ (3.2)	✗	✗
Concepts Only (`CO`)	–	✗	✗	✓ (3.1)	–	✗
Concepts plus Deduction (`CDe`)	✓	✓	✓ (3.3)	✓ (3.1)	✗	✗
Joint (`RPB`) [57]	✓	✗	✓	✓	✓	✗
`COCOCO`	✓ (B.7)	✓ (B.7)	✓ (4.2)	✓ (4.2)	✓ (B.8, 6.2)	✓ (B.8, 6.2)

4.1 One-Sided Strategies do not Suffice

The joint revision operator $R(\Gamma, \Upsilon) = (\Gamma \cap g^{-\dagger}(\Upsilon),\ \Upsilon \cap g^{\dagger}(\Gamma))$ admits four *one-sided* restrictions, obtained by conformalizing only one side and either bypassing the symbolic layer or applying $g^{\dagger}/g^{-\dagger}$ unilaterally. We briefly discuss them and their properties (summarized in Table 1):

- Task only (`TO`) applies CP to the labels but not to the concepts, meaning it violates coverage **D2**; it also violates consistency **D1** as a single concept vector may not entail the entire label set.
- Concepts only (`CO`) is the converse, i.e., it does not apply CP to the task labels, violating coverage **D2**; moreover, if employed, Bonferroni inflates $\mathbf{\Gamma}^{\times}(\mathbf{x})$, breaking conciseness **D3**.
- Task plus Abduction (`TAb`) computes $(\Upsilon_{\alpha}(\mathbf{x}), g^{-\dagger}(\Upsilon_{\alpha}(\mathbf{x})))$, in which case the abduced concept set contains *all* concept vectors $\mathbf{c}$ entailing some label in $\Upsilon_{\alpha}(\mathbf{x})$, affecting conciseness **D3**.
- Concepts plus Deduction (`CDe`) computes $(\Gamma_{\beta}(\mathbf{x}), g^{\dagger}(\Gamma_{\beta}(\mathbf{x})))$ instead, meaning that it inflates the label side, violating **D3**.

Ramalingam et al. [57] propose a variation of the revision operator, here denoted as `RPB`, that applies R asymmetrically, revising only the label side. It matches `COCOCO` $\Upsilon^{\mathsf{rev}}(\mathbf{x})$ but leaves $\Gamma_{\beta}(\mathbf{x})$ untouched, breaking concept-side consistency **D1**. We defer to Section A for an in-depth discussion.

5 Related Work

Conformal Prediction. CP produces prediction sets containing the ground-truth value with a pre-specified coverage [13]. Core theoretical guarantees are established by Papadopoulos et al. [40]. Extensions cover regression [62, 63]; predictive distributions [64, 65, 66]; conditional guarantees [67, 68]; noisy ground truth [69, 70], covariate shifts [71] and size control [20, 21]. We refer the reader to Angelopoulos and Bates [41] for a broader survey on conformal prediction.

Conformal prediction for NeSy. The two closest works are Ledaguenel et al. [61], who apply CP only at the label level (matching our `TO` baseline and inheriting its lack of concept-side coverage), and Ramalingam et al. [57], who propose three strategies for symbolic programs over bounding-boxes that align with our `CDe`, `TO`, and `RPB` baselines. Both are subsumed by our framework. `COCOCO` goes further by *also* revising the concept set via $g^{-\dagger}(\Upsilon_{\alpha}(\mathbf{x}))$, which is what guarantees concept-side consistency and drives the empirical concept-set reductions reported in Section 6. Other lines of work [72, 73] use CP for LLM-generated programs and are orthogonal to ours. Beyond also revising the concept side, `COCOCO` differs from `RPB` in scope. We derive the revision from explicit desiderata for general NeSy-CBMs, prove it is the minimal-deletion fix (Proposition 4.1), characterize coverage under imperfect knowledge via $\delta_{\mathrm{ab}}, \delta_{\mathrm{de}}$ (Proposition 4.2), and support user-specified size budgets via e-values.

NeSy-CBMs and Reasoning Shortcuts. NeSy-CBMs constitute a broad family of architectures that combine neural concept extractors with symbolic inference layers [1, 5, 23, 74], encompassing models based on probabilistic logic [75, 8, 76, 77, 78, 51, 45, 60, 79, 80], fuzzy logic [81, 30, 28, 82], and algebraic constraints [47, 51, 83]. A known failure mode of these models is that of *reasoning shortcuts* [24], whereby the concept extractor learns incorrect concepts that allow to predict correct labels in training [6, 84]. This phenomenon is mainly tackled by resorting to additional supervision [24, 85, 86] and uncertainty estimation [9, 10].

Uncertainty quantification in NeSy-CBMs. NeSy-CBMs, despite their architectural bias toward reliability, are not immune to overconfident predictions [9]. Existing approaches fall into two categories. The first modifies the concept distribution to be more expressive [33], e.g., through deep

ensembles [9] or diffusion-based parameterizations [10]. The second enforces uncertainty-related properties during training through constraints-based objectives [87]. Our approach operates on top of an already-trained NeSy-CBM and provides *distribution-free* coverage on top of any of these methods. Orthogonal to our work, another line of research seeks to integrate uncertainty quantification into CBMs. Recent approaches explore integration with Gaussian processes, credal sets, and learning to defer frameworks [88, 89, 90].

6 Experiments

We address the following research questions:

RQ1 How does COCOCO fare in terms of **D1**–**D3** under correct and noisy knowledge?

RQ2 How does COCOCO behave when deductive soundness fails?

RQ3 Can we impose *hard size constraints* on the prediction sets while retaining coverage guarantees?

The code is publicly available at https://github.com/samuelebortolotti/conformal-nesy.

6.1 Experimental Settings

We report results in the main text using *concept supervision*, as the unsupervised setting yields substantially larger prediction sets that are harder to read and less operationally meaningful; the unsupervised results follow the same qualitative trends and are reported in full in Section E.

Data sets. We focus on four datasets, grouped by the properties of the prior knowledge K. *Complete-and-correct* K: CHEST-XRAYS (CHX [91]) is a high-stakes medical diagnosis task in which the label is the number of active binary symptoms; CEBAB (CEB [92]) is a restaurant-review dataset with four aspect-level concepts and an overall rating derived by majority vote. *Incomplete* K: RIVAL-10 (RIV10 [93]) contains 10 classes with hand-crafted visual attributes such that at least two class pairs share the same concept signature, so deductive soundness fails (Definition 3.1). *Reasoning-shortcut benchmark*: MNIST-EVENODD (MN-EO [94]) is a biased variant of MN-Add [8] designed to induce reasoning shortcuts. Details for these and the five additional datasets are in Section C.5.

Competitors and metrics. We employ two state-of-the-art NeSy-CBMs, DPL [8] and LTN [28], trained with and without concept supervision, and compare COCOCO against RPB [57], the strongest competitor; full ablations against TO, TAb, CO, and CDe are in Section E.6. Following **D1**–**D3**, we report COVERAGE, SIZE, and CONSISTENCY at nominal coverages $1-\alpha = 1-\beta = 0.9$, averaged over 10 seeds. Details about hyperparameters and metrics can be found in Section C and Section D.

Setup. We train DPL and LTN end-to-end on the training split following [94]. We partition the test set into a calibration set $\mathcal{D}_{\text{cal}}$ (20%) and a held-out evaluation set $\mathcal{D}_{\text{test}}$ (80%), computing nonconformity scores $s(\mathbf{x}, \mathbf{y}) = -\log p_\theta(\mathbf{y} \mid \mathbf{x})$[6] on $\mathcal{D}_{\text{cal}}$ for both concepts and labels, and use them to construct the conformal sets $\Gamma_\beta(\mathbf{x})$ (in practice, $\mathbf{\Gamma}^\times(\mathbf{x})$) and $\Upsilon_\alpha(\mathbf{x})$ at the desired miscoverage levels. COCOCO revision step (Eq. (10)) is then applied at test time on $\mathcal{D}_{\text{test}}$, for which we report all metrics. For COCOCO*, we replace the standard nonconformity scores with soft-rank e-values (Eq. (99)) and adaptively select $\tilde{\alpha}$ and $\tilde{\beta}$ via bootstrap (Section C.3) to satisfy user-specified size budgets. Full training, conformalization, and evaluation details are in Section C.

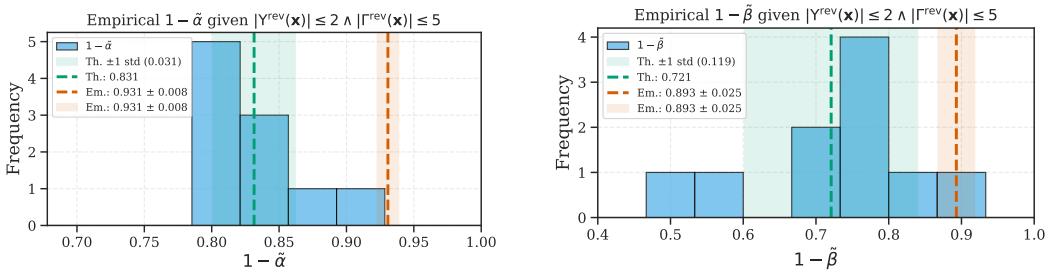
6.2 Experimental Results and Discussion

RQ1: COCOCO preserves coverage and consistency with smaller prediction sets across all regimes. By symmetrically revising both sides, COCOCO matches RPB label sets while *strictly* improving the concept side. COCOCO attains concept consistency 1.00 across all four datasets, whereas RPB ranges from 0.12 (MN-EO) to 0.94 (RIV10). With *complete-and-correct* K: on CHX, COCOCO shrinks the concept set from 5.30 (RPB) to 2.90 (1.8×); on CEB, a consistent reduction, $75.06 \rightarrow 53.49$ (1.4×), with concept consistency rising $0.78 \rightarrow 1.00$. With *incomplete* K (RIV10), COCOCO reduces the concept set $6.33 \rightarrow 5.85$ and increases consistency $0.94 \rightarrow 1.00$. On *RS-benchmark* (MN-EO, unsupervised), RPB returns concept sets of size 100, i.e., all possible digits combinations, at consistency 0.12, reflecting the strong miscalibration induced by RSs [9, 10]; COCOCO shrinks this

[6] For LTN, we define $p_\theta(\mathbf{y} \mid \mathbf{x})$ as the fuzzy satisfaction degrees $\mathcal{T}_{\mathsf{K}}(p_\theta(\mathbf{C} \mid \mathbf{x}), \mathbb{1}\{\mathbf{Y} = \mathbf{y}\})$ over $\mathcal{Y}$.

Table 2: Results for DPL on three representative datasets. We report concept and label consistency (CONST., fraction of consistent pairs, see Section C.2), size, and coverage at nominal coverage $1-\alpha = 1-\beta = 0.9$, averaged over 10 seeds (mean $\pm$ std). Full results, including LTN, are in Section E.

		CONCEPTS			LABELS		
	METHOD	CONST.	SIZE	COV.	CONST.	SIZE	COV.
CHX	DPL	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	0.47 ± 0.11	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	0.72 ± 0.01
	DPL + RPB	0.55 ± 0.05	5.30 ± 1.01	$\mathbf{0.91 \pm 0.03}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.60 \pm 0.18}$	$\mathbf{0.88 \pm 0.03}$
	DPL + COCOCO	$\mathbf{1.00 \pm 0.00}$	$\mathbf{2.90 \pm 0.56}$	0.83 ± 0.04	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.60 \pm 0.18}$	$\mathbf{0.88 \pm 0.03}$
RIV10	DPL	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	0.43 ± 0.02	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	0.80 ± 0.00
	DPL + RPB	0.94 ± 0.01	6.33 ± 0.65	$\mathbf{0.92 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.20 \pm 0.02}$	$\mathbf{0.89 \pm 0.01}$
	DPL + COCOCO	$\mathbf{1.00 \pm 0.00}$	$\mathbf{5.85 \pm 0.62}$	0.84 ± 0.01	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.20 \pm 0.02}$	$\mathbf{0.89 \pm 0.01}$
CEB	DPL	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	0.18 ± 0.02	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	0.58 ± 0.03
	DPL + RPB	0.78 ± 0.06	75.06 ± 38.35	$\mathbf{0.90 \pm 0.04}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{2.31 \pm 0.16}$	$\mathbf{0.87 \pm 0.01}$
	DPL + COCOCO	$\mathbf{1.00 \pm 0.00}$	$\mathbf{53.49 \pm 25.93}$	0.83 ± 0.03	$\mathbf{1.00 \pm 0.00}$	$\mathbf{2.31 \pm 0.16}$	$\mathbf{0.87 \pm 0.01}$



(a) Label coverage

(b) Concept coverage

Figure 2: COCOCO* on CHX with DPL under average size budgets $|\Upsilon^{\text{rev}}(\mathbf{x})| \leq 2$ and $|\Gamma^{\text{rev}}(\mathbf{x})| \leq 5$, bootstrapped over 100 calibration resamples and averaged across 10 seeds. Empirical coverage (**orange**) exceeds the theoretical target (**green**), while sizes remain *strictly* below the imposed budgets.

to 12.13 (8.2×) at consistency 1.00, with label coverage preserved at 0.90. COCOCO and RPB both maintain label-set sizes ≤ 2, except on CEB (2.31); label coverage is ≥ 0.87 for correct and concept coverage is ≥ 0.83. Results for LTN are deferred to Appendix Section E due to space constraints.

Trade-off note. Concept coverage of COCOCO is systematically 7–10 points below RPB (e.g., 0.83 vs 0.91 on CHX, 0.84 vs 0.92 on RIV10). This is unavoidable since $\Gamma^{\text{rev}}(\mathbf{x}) \subseteq \Gamma_\beta(\mathbf{x})$, so any concept removed by intersection with $\Gamma^{\text{ab}}(\mathbf{x})$ previously contributed to coverage. The drop reflects cases in which $\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x})$, potentially causing $\Gamma^{\text{ab}}(\mathbf{x})$ to exclude $\mathbf{c}^*$. COCOCO trades this small coverage drop for guaranteed concept-side consistency (always 1.00) and substantially smaller sets. This trade-off is tunable. By splitting a fixed total miscoverage budget between labels and concepts ($\alpha + \beta = 0.1$), thereby matching COCOCO nominal guarantee to RPB, we obtain concept coverage on par with RPB (0.89–0.91) while using smaller concept sets (4.08–4.57 vs. 5.30 on CHX), maintaining perfect consistency, and achieving higher label coverage (Section E.4).

RQ2: coverage guarantees degrade controllably under incomplete knowledge. The bound in Proposition 4.2 degrades with δ_{ab} and δ_{de}. RIV10 is *designed* so that two pairs of classes (dog/horse, automobile/truck) share the same concepts, yielding $\delta_{\text{de}} = 0.8$ by construction (verified empirically; see Table 50). Abductive soundness holds, $\hat{\delta}_{\text{ab}} = 1.00$. Plugging $\delta_{\text{de}} = 0.80$, $\beta = 0.1$ into Proposition 4.2 yields $\Pr(\mathbf{y}^* \in \Upsilon^{\text{rev}}(\mathbf{x})) \geq 0.62$.[7] This bound is respected: DPL attains label coverage 0.80 on RIV10. Section E reports the joint-failure correction $\Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\text{de}}(\mathbf{x}))$ and $\Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\text{ab}}(\mathbf{x}))$, respectively, for both CIF10 and RIV10, confirming it is small (≤ 0.06) and closes the gap between the theoretical lower bound and observed coverage. Results on CIF10 follow a similar trend (Section E).

RQ3: COCOCO supports user-specified size budgets while retaining coverage. A recurring practical concern with CP is that size and coverage cannot be jointly specified. Conformal e-predictions [18, 20] invert the procedure: given size budgets, the method adaptively chooses $\tilde{\alpha}$ and $\tilde{\beta}$ while retaining the conformal guarantee (Proposition B.10). E-values additionally avoid Bonferroni's correction (Proposition B.9), so guarantees do not degrade with the number of concepts. We

[7] This excludes the joint-failure correction term, which further tightens the bound slightly; see Section E for more details.

evaluate on CHX, fixing label and concept budgets to 2 and 5, operationally meaningful for a clinical decision-support setting. Fig. 2 reports the both-fixed regime, bootstrapped over 100 calibration resamples. Average set sizes are 1.80 (labels) and 3.57 (concepts), strictly below the budgets, with empirical coverages 0.91 and 0.89, both exceeding by far the theoretical targets $1 - \mathbb{E}[\tilde{\alpha}] = 0.82$ and $1 - \mathbb{E}[\tilde{\beta}] = 0.77$. Single-sided regimes and full results are in Table 45 and additional ablation on different e-value formulation in Section E.3.

7 Conclusion

We introduced COCOCO, a post-hoc conformal framework for NeSy-CBMs that jointly conformalizes concept and label predictions and reconciles them via a *single* deduction–abduction revision step. This step provably yields the largest logically consistent revision of the two sets and reaches a fixed point in one pass (Proposition 4.1). The framework retains distribution-free coverage that degrades controllably under imperfect knowledge, and, combined with e-values, supports user-specified size budgets. To our knowledge, COCOCO is the first method that exploits symbolic knowledge to *simultaneously* shrink conformal prediction sets and enforce label–concept consistency while retaining distribution-free coverage. Empirically, COCOCO consistently attains target coverage while reducing concept-set size on eight benchmarks.

Limitations and Future Work. Conformalizing concepts requires concept annotations for the calibration set, but not for training. Although this set can be small, on CHX, just 43 examples suffice (Section E.5), collecting concept annotations may still be costly in some domains. Cheaper annotation methods that rely on foundation models are a promising complement, although they may be unreliable [95, 96]. COCOCO assumes K is given. When it is incomplete, the bounds of Proposition 4.2 degrade linearly in $\delta_{\mathrm{ab}}, \delta_{\mathrm{de}}$. When K is acquired or learned from data [97, 98, 99, 100, 101], soundness cannot be assumed a priori and these quantities must be estimated. Abduction and deduction require enumerating $g^{\dagger}(\Gamma_{\beta}(\mathbf{x}))$ and $g^{-\dagger}(\Upsilon_{\alpha}(\mathbf{x}))$, while $\mathbf{\Gamma}^{\times}(\mathbf{x})$ requires to compute the product over a possibly large number of concepts. For very large concept spaces, this becomes a bottleneck, although exploiting the structure of K can make it tractable, as we show on BDD-OIA (Section E.1). Since the inference layer g is evaluated multiple times, we rely on knowledge-compiled circuits [26, 102, 27, 103, 104] (as in DPL) to keep inference tractable in practice. Finally, the product set $\Gamma^{\times}(\mathbf{x})$ relies on per-concept conformal sets, which is natural for NeSy-CBMs with conditionally independent concepts [33, 10]. As with any CP method, set sizes depend on how well the NeSy-CBM fits the underlying concept and label distributions. Regarding the coverage guarantees, since $\Gamma^{\mathsf{rev}}(\mathbf{x}) \subseteq \Gamma_{\beta}(\mathbf{x})$, concept coverage is necessarily lower for $\Gamma^{\mathsf{rev}}(\mathbf{x})$ than for $\Gamma_{\beta}(\mathbf{x})$; this is the price of enforcing consistency, quantified by Proposition 4.2 and controlled through α and β (Section E.4). Importantly, the COCOCO revision step itself does *not* require independence. It operates on any concept set satisfying Eq. (6). Plugging in expressive concept distributions [33, 10] is a direct extension. As future work, we plan to extend COCOCO to a broader range of NeSy-CBMs architectures, leveraging, once mature, recent initiatives that aim to make NeSy-AI more accessible and easier to adopt for practitioners, such as DeepLog [104] and ULLER [105], and to architectures that *learn* the knowledge from data [106, 52, 53, 107, 54, 108, 55], where soundness can no longer be assumed a priori and must be estimated, affecting the bounds of Proposition 4.2.

Broader Impact

By coupling distribution-free coverage with logical consistency, COCOCO makes NeSy-CBMs suitable for decision-support systems [34, 35, 38, 36], particularly in high-stakes settings such as CHX where practitioners consume both intermediate concepts and final labels. We foresee no negative societal impact specific to COCOCO beyond those shared by any conformal method.

Author Contributions

SB conceived the idea, developed the theory, and ran the experiments. EM and APu contributed to the formalization and verification of the proofs. APu, EM, APa and ST supervised the work.

Acknowledgments

The authors are grateful to Tommaso Carraro for his precious help in the experimental evaluation, and to Cesare Barbera and Carlotta Giacchetta for their useful discussions. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, the MSCA-PF programme, or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them. Grant Agreement no. 101120763 - TANGO. EM is funded by the Horizon Europe/Marie Sklodowska-Curie under Grant Agreement No. 101282115 - AI-CURE.

References

[1] Luc De Raedt, Sebastijan Dumančić, Robin Manhaeve, and Giuseppe Marra. From statistical relational to neural-symbolic artificial intelligence. In *IJCAI*, 2021.

[2] Artur dAvila Garcez, Sebastian Bader, Howard Bowman, Luis C Lamb, Leo de Penning, BV Illuminoo, Hoifung Poon, and COPPE Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342:1, 2022.

[3] Eleonora Giunchiglia, Mihaela Catalina Stoian, and Thomas Lukasiewicz. Deep learning with logical constraints. pages 5478–5485, 2022. doi: 10.24963/IJCAI.2022/767. URL https://doi.org/10.24963/ijcai.2022/767.

[4] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 2022.

[5] Patrick Knab, David Steinmann, Christian Bartelt, Kristian Kersting, Bernt Schiele, Thomas Seidl, Udo Schlegel, and Wolfgang Stammer. What's in the bottle? A survey and roadmap of concept bottleneck models. *Trans. Mach. Learn. Res.*, 2026, 2026.

[6] Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Paolo Morettin, Elena Umili, Antonio Vergari, Efthymia Tsamoura, Andrea Passerini, and Stefano Teso. Symbol grounding in neuro-symbolic ai: A gentle introduction to reasoning shortcuts. *Journal of Artificial Intelligence Research*, 86, 2026. doi: 10.1613/jair.1.22389.

[7] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[8] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural Probabilistic Logic Programming. *NeurIPS*, 2018.

[9] Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Antonio Vergari, Andrea Passerini, and Stefano Teso. BEARS Make Neuro-Symbolic Models Aware of their Reasoning Shortcuts. *Uncertainty in AI*, 2024.

[10] Emile van Krieken, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. Neurosymbolic diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[11] Salvatore Ruggieri and Andrea Pugnana. Things machine learning models know that they don't know. In *AAAI*, pages 28684–28693. AAAI Press, 2025.

[12] Charvi Rastogi, Yunfeng Zhang, Dennis Wei, Kush R Varshney, Amit Dhurandhar, and Richard Tomsett. Deciding fast and slow: The role of cognitive biases in ai-assisted decision-making. *Proceedings of the ACM on Human-computer Interaction*, 6(CSCW1):1–22, 2022.

[13] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of machine learning research*, 9(3), 2008.

[14] Glenn Shafer and Vladimir Vovk. *Game-theoretic foundations for probability and finance*. John Wiley & Sons, 2019.

[15] Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

[16] Peter Grünwald, Rianne de Heide, and Wouter M Koolen. Safe testing. In *2020 Information theory and applications workshop (ITA)*, pages 1–54. IEEE, 2020.

[17] Aaditya Ramdas and Ruodu Wang. Hypothesis testing with e-values. *Foundations and Trends® in Statistics*, 1(1-2):1–390, 2025.

[18] Vladimir Vovk. Conformal e-prediction. *Pattern Recognition*, 166:111674, 2025.

[19] Etienne Gauthier, Francis Bach, and Michael I Jordan. E-values expand the scope of conformal prediction. *arXiv preprint arXiv:2503.13050*, 2025.

[20] Etienne Gauthier, Francis R. Bach, and Michael I. Jordan. Backward conformal prediction. In *NeurIPS*, 2025.

[21] Etienne Gauthier, Francis R. Bach, and Michael I. Jordan. Adaptive coverage policies in conformal prediction. In *AISTATS*, 2026.

[22] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Neural probabilistic logic programming in deepproblog. *Artificial Intelligence*, 298: 103504, 2021.

[23] Samuele Bortolotti, Emanuele Marconato, Paolo Morettin, Andrea Passerini, and Stefano Teso. Shortcuts and identifiability in concept-based models from a neuro-symbolic lens. 2025.

[24] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. In *NeurIPS*, 2023.

[25] Luc De Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100:5–47, 2015.

[26] Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[27] Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. *NeurIPS*, 2021.

[28] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

[29] Davide Rinaldi and Luciano Serafini. sltn: Structural logic tensor networks, 2026. URL https://arxiv.org/abs/2608.11136.

[30] Ivan Donadello, Luciano Serafini, and Artur D'Avila Garcez. Logic tensor networks for semantic image interpretation. In *IJCAI*, 2017.

[31] Emile van Krieken, Erman Acar, and Frank van Harmelen. Analyzing Differentiable Fuzzy Implications. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, pages 893–903, 9 2020. doi: 10.24963/kr.2020/92. URL https://doi.org/10.24963/kr.2020/92.

[32] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretable, leak-proof concept-based models. In *NeurIPS*, 2022.

[33] Emile Van Krieken, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. On the independence assumption in neurosymbolic learning. In *International Conference on Machine Learning*, pages 49078–49097. PMLR, 2024.

[34] Giovanni De Toni, Nastaran Okati, Suhas Thejaswi, Eleni Straitouri, and Manuel Gomez-Rodriguez. Towards human-ai complementarity with prediction sets. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 31380–31409. Curran Associates, Inc., 2024. doi: 10.52202/079017-0986. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/37d4d4413b7c7558cc27a6d3d42ea998-Paper-Conference.pdf.

[35] Eleni Straitouri and Manuel Gomez Rodriguez. Designing decision support systems using counterfactual prediction sets. In *International Conference on Machine Learning*, pages 46722–46744. PMLR, 2024.

[36] Alessandro Bogani, Nicola Debole, Emanuele Marconato, Andrea Pugnana, Katya Tentori, and Andrea Passerini. Are concept bottleneck models effective as decision-support systems?, 2026. URL https://arxiv.org/abs/2608.25581.

[37] Adrián Arnaiz-Rodríguez, Nina Corvelo Benz, Suhas Thejaswi, Nuria Oliver, and Manuel Gomez Rodriguez. Towards human-ai complementarity in matching tasks. In *PKDD/ECML Workshops (2)*, volume 2840 of *Communications in Computer and Information Science*, pages 36–59. Springer, 2025.

[38] Eleni Straitouri, Suhas Thejaswi, and Manuel Gomez Rodriguez. Controlling counterfactual harm in decision support systems based on prediction sets. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 129443–129479. Curran Associates, Inc., 2024. doi: 10.52202/079017-4113. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e9e3e5bdb017bbc887271c6f6de5353f-Paper-Conference.pdf.

[39] Ron S. Boger, Seyone Chithrananda, Anastasios N. Angelopoulos, Peter H. Yoon, Michael I. Jordan, and Jennifer A. Doudna. Functional protein mining with conformal guarantees. *Nature Communications*, 16(1):85, Jan 2025. ISSN 2041-1723. doi: 10.1038/s41467-024-55676-y. URL https://doi.org/10.1038/s41467-024-55676-y.

[40] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alexander Gammerman. Inductive confidence machines for regression. In *ECML*, volume 2430 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2002.

[41] Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.*, 16(4):494–591, 2023.

[42] Eleonora Giunchiglia, Fergus Imrie, Mihaela van der Schaar, and Thomas Lukasiewicz. Machine learning with requirements: A manifesto. *Neurosymbolic Artificial Intelligence*, 1:NAI–240767, 2025. doi: 10.3233/NAI-240767. URL https://doi.org/10.3233/NAI-240767.

[43] Eleonora Giunchiglia, Mihaela Cătălina Stoian, Salman Khan, Fabio Cuzzolin, and Thomas Lukasiewicz. Road-r: the autonomous driving dataset with logical requirements. *Machine Learning*, 112(9):3261–3291, 2023.

[44] Wang-Zhou Dai and Stephen H. Muggleton. Abductive knowledge induction from raw data. In *IJCAI*, pages 1845–1851. ijcai.org, 2021.

[45] Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic Probabilistic Layers for Neuro-Symbolic Learning. In *NeurIPS*, 2022.

[46] Raffaele Pojer, Andrea Passerini, and Manfred Jaeger. Generalized reasoning with graph neural networks by relational bayesian network encodings. In Soledad Villar and Benjamin Chamberlain, editors, *Proceedings of the Second Learning on Graphs Conference*, volume 231 of *Proceedings of Machine Learning Research*, pages 16:1–16:12. PMLR, 27–30 Nov 2024. URL https://proceedings.mlr.press/v231/pojer24a.html.

[47] Leander Kurscheidt, Paolo Morettin, Roberto Sebastiani, Andrea Passerini, and Antonio Vergari. A probabilistic neuro-symbolic layer for algebraic constraint satisfaction. In Silvia Chiappa and Sara Magliacane, editors, *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 2431–2471. PMLR, 21–25 Jul 2025. URL https://proceedings.mlr.press/v286/kurscheidt25a.html.

[48] Leander Kurscheidt, Gabriele Masina, Roberto Sebastiani, and Antonio Vergari. The theory and practice of MAP inference over non-convex constraints. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=jIZqAemuqk.

[49] Weixin Chen, Simon Yu, Huajie Shao, Lui Sha, and Han Zhao. Neural probabilistic circuits: Enabling compositional and interpretable predictions through logical reasoning. *Machine Learning*, 115(9):207, 2026.

[50] Weixin Chen and Han Zhao. Understanding and improving adversarial robustness of neural probabilistic circuits. In *NeurIPS*, 2025.

[51] Eleonora Giunchiglia, Alex Tatomir, Mihaela Cătălina Stoian, and Thomas Lukasiewicz. Ccn+: A neuro-symbolic framework for deep learning with requirements. *International Journal of Approximate Reasoning*, page 109124, 2024.

[52] Alessandro Daniele, Tommaso Campari, Sagar Malhotra, and Luciano Serafini. Deep symbolic learning: discovering symbols and rules from perceptions. In *IJCAI*, 2023.

[53] Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, and Kristian Kersting. (alpha)ilp: thinking visual scenes as differentiable logic programs. *Machine Learning*, 112(5):1465–1497, 2023.

[54] Antonia Wüst, Wolfgang Stammer, Quentin Delfosse, Devendra Singh Dhami, and Kristian Kersting. Pix2code: Learning to compose neural visual concepts as programs. In *UAI*, volume 244 of *Proceedings of Machine Learning Research*, pages 3829–3852. PMLR, 2024.

[55] David Debot, Pietro Barbiero, Francesco Giannini, Gabriele Ciravegna, Michelangelo Diligenti, and Giuseppe Marra. Interpretable concept-based memory reasoning. In *NeurIPS*, 2024.

[56] Ghassan Najjar, Céline Berthou, and Héléna Vorobieva. Scaling multi-label conformal prediction with label interactions for a large number of labels. In Rita P. Ribeiro, Bernhard Pfahringer, Nathalie Japkowicz, Pedro Larrañaga, Alípio M. Jorge, Carlos Soares, Pedro H. Abreu, and João Gama, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 111–128, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-06109-6.

[57] Ramya Ramalingam, Sangdon Park, and Osbert Bastani. Uncertainty quantification for neurosymbolic programs via compositional conformal prediction. *arXiv preprint arXiv:2405.15912*, 2024.

[58] Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Science China. Information Sciences*, 62(7):76101, 2019.

[59] Myeongjun Jang and Thomas Lukasiewicz. Are training resources insufficient? predict first then explain! *arXiv preprint arXiv:2110.02056*, 2021.

[60] Emile van Krieken, Thiviyan Thanapalasingam, Jakub M. Tomczak, Frank van Harmelen, and Annette ten Teije. A-nesi: A scalable approximate method for probabilistic neurosymbolic inference. In *NeurIPS*, 2023.

[61] Arthur Ledaguenel, Céline Hudelot, and Mostepha Khouadjia. Neurosymbolic conformal classification. *arXiv preprint arXiv:2409.13585*, 2024.

[62] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, 2014.

[63] Jing Lei, Max GSell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[64] Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. Nonparametric predictive distributions based on conformal prediction. In *Conformal and probabilistic prediction and applications*, pages 82–102. PMLR, 2017.

[65] Vladimir Vovk, Ilia Nouretdinov, Valery Manokhin, and Alex Gammerman. Cross-conformal predictive distributions. *Proceedings of Machine Learning Research*, 91:1–15, 2018.

[66] Vladimir Vovk, Ivan Petej, Ilia Nouretdinov, Valery Manokhin, and Alexander Gammerman. Computationally efficient versions of conformal predictive distributions. *Neurocomputing*, 397:292–308, 2020.

[67] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

[68] Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(4): 1100–1126, 2025.

[69] David Stutz, Abhijit Guha Roy, Tatiana Matejovicova, Patricia Strachan, Ali Taylan Cemgil, and Arnaud Doucet. Conformal prediction under ambiguous ground truth. *Trans. Mach. Learn. Res.*, 2023, 2023.

[70] Michele Caprio, David Stutz, Shuo Li, and Arnaud Doucet. Conformalized credal regions for classification with ambiguous ground truth. *Trans. Mach. Learn. Res.*, 2025, 2025.

[71] Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *NeurIPS*, pages 2526–2536, 2019.

[72] Stephen Mell, Botong Zhang, David Mell, Shuo Li, Ramya Ramalingam, Nathan Yu, Steve Zdancewic, and Osbert Bastani. A fast, reliable, and secure programming language for llm agents with code actions, 2025. URL https://arxiv.org/abs/2506.12202.

[73] Celeste Barnaby, Qiaochu Chen, Ramya Ramalingam, Osbert Bastani, and Işıl Dillig. Active learning for neurosymbolic program synthesis. *Proc. ACM Program. Lang.*, 9(OOPSLA2), October 2025. doi: 10.1145/3763102. URL https://doi.org/10.1145/3763102.

[74] Weixin Chen, Antonio Vergari, and Han Zhao. Can VLMs reason robustly? a neuro-symbolic investigation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=4y6jiE6Q60.

[75] Marco Lippi and Paolo Frasconi. Prediction of protein β-residue contacts by markov logic networks with grounding-specific weights. *Bioinformatics*, 2009.

[76] Giuseppe Marra and Ondřej Kuželka. Neural markov logic networks. In *Uncertainty in Artificial Intelligence*, 2021.

[77] Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, and Xujie Si. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. *NeurIPS*, 2021.

[78] Connor Pryor, Charles Dickens, Eriq Augustine, Alon Albalak, William Yang Wang, and Lise Getoor. Neupsl: Neural probabilistic soft logic. In *IJCAI*, pages 4145–4153. ijcai.org, 2023.

[79] Jaron Maene and Luc De Raedt. Soft-unification in deep probabilistic logic. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[80] Arseny Skryagin, Wolfgang Stammer, Daniel Ochs, Devendra Singh Dhami, and Kristian Kersting. Neural-probabilistic answer set programming. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 463–473, 2022.

[81] Michelangelo Diligenti, Marco Gori, and Claudio Sacca. Semantic-based regularization for learning and inference. *Artificial Intelligence*, 2017.

[82] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. *NeurIPS*, 2020.

[83] Lennert De Smet, Pedro Zuidberg Dos Martires, Robin Manhaeve, Giuseppe Marra, Angelika Kimmig, and Luc De Raedt. Neural probabilistic logic programming in discrete-continuous domains. In *Uncertainty in Artificial Intelligence*, pages 529–538. PMLR, 2023.

[84] David Steinmann, Felix Divo, Maurice Kraus, Antonia Wüst, Lukas Struppek, Felix Friedrich, and Kristian Kersting. Navigating shortcuts, spurious correlations, and confounders: From origins via detection to mitigation. *arXiv preprint arXiv:2412.05152*, 2024.

[85] Xiao-Wen Yang, Wen-Da Wei, Jie-Jing Shao, Yu-Feng Li, and Zhi-Hua Zhou. Analysis for abductive learning and neural-symbolic reasoning shortcuts. In *ICML*, 2024.

[86] Luca Andolfi and Eleonora Giunchiglia. Right for the right reasons: Avoiding reasoning shortcuts via prototypical neurosymbolic ai. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 157309–157343. Curran Associates, Inc., 2025. doi: 10.52202/085713-5256. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e6bb63d00f597c380c012bd99c68b0c1-Paper-Conference.pdf.

[87] Shahnawaz Alam, Mohammed Mudassir Uddin, and Mohammed Kaif Pasha. Constraint-aware neurosymbolic uncertainty quantification with bayesian deep learning for scientific discovery. *arXiv preprint arXiv:2601.12442*, 2026.

[88] Nicola Debole, Andrea Passerini, Stefano Teso, Andrea Pugnana, and Emanuele Marconato. Concepts worth having: Refining vlm-guided concept bottleneck models with minimal annotations. In *NeurIPS*, 2026.

[89] Tanmoy Mukherjee, Thomas Bailleux, Pierre Marquis, and Zied Bouraoui. Credal concept bottleneck models for epistemic-aleatoric uncertainty decomposition. In *ACL (1)*, pages 42961–42980. Association for Computational Linguistics, 2026.

[90] Andrea Pugnana, Riccardo Massidda, Francesco Giannini, Pietro Barbiero, Mateo Espinosa Zarlenga, Roberto Pellungrini, Gabriele Dominici, Fosca Giannotti, and Davide Bacciu. Deferring concept bottleneck models: Learning to defer interventions to inaccurate experts. In *NeurIPS*, 2025.

[91] Joseph Paul Cohen, Joseph D. Viviano, Paul Bertin, Paul Morrison, Parsa Torabian, Matteo Guarrera, Matthew P Lungren, Akshay Chaudhari, Rupert Brooks, Mohammad Hashir, and Hadrien Bertrand. Torchxrayvision: A library of chest x-ray datasets and models. In Ender Konukoglu, Bjoern Menze, Archana Venkataraman, Christian Baumgartner, Qi Dou, and Shadi Albarqouni, editors, *Proceedings of The 5th International Conference on Medical Imaging with Deep Learning*, volume 172 of *Proceedings of Machine Learning Research*, pages 231–249. PMLR, 06–08 Jul 2022. URL https://proceedings.mlr.press/v172/cohen22a.html.

[92] Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. *Advances in Neural Information Processing Systems*, 35: 17582–17596, 2022.

[93] Mazda Moayeri, Phillip Pope, Yogesh Balaji, and Soheil Feizi. A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022.

[94] Samuele Bortolotti, Emanuele Marconato, Tommaso Carraro, Paolo Morettin, Emile van Krieken, Antonio Vergari, Stefano Teso, and Andrea Passerini. A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. In *NeurIPS*, 2024.

[95] Nicola Debole, Pietro Barbiero, Francesco Giannini, Andrea Passerini, Stefano Teso, and Emanuele Marconato. If concept bottlenecks are the question, are foundation models the answer? *Mach. Learn.*, 115(5):97, 2026.

[96] Antonia Wüst, Tim Tobiasch, Lukas Helff, Inga Ibs, Wolfgang Stammer, Devendra Singh Dhami, Constantin A Rothkopf, and Kristian Kersting. Bongard in wonderland: Visual puzzles that still make ai go mad? In *International Conference on Machine Learning*, pages 68118–68142. PMLR, 2025.

[97] Stephen H. Muggleton. Inductive logic programming. *New Gener. Comput.*, 8(4):295–318, 1991.

[98] Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. In *ALT*, volume 3244 of *Lecture Notes in Computer Science*, pages 19–36. Springer, 2004.

[99] Antonia Wüst, Wolfgang Stammer, Hikaru Shindo, Lukas Helff, Devendra Singh Dhami, and Kristian Kersting. Synthesizing visual concepts as vision-language programs. In *CVPR*, pages 17346–17356. Computer Vision Foundation, 2026.

[100] Lukas Helff, Ahmad Omar, Felix Friedrich, Antonia Wüst, Hikaru Shindo, Rupert Mitchell, Tim Woydt, Patrick Schramowski, Wolfgang Stammer, and Kristian Kersting. SLR: Automated synthesis for scalable logical reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–426, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.16. URL https://aclanthology.org/2026.acl-long.16/.

[101] Samuele Bortolotti, Weixin Chen, Han Zhao, Andrea Passerini, Stefano Teso, and Antonio Vergari. Auto-formalizing neuro-symbolic predictors, 2026. URL https://arxiv.org/abs/2610.01519.

[102] Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*, page 6, 2020.

[103] Jaron Maene, Vincent Derkinderen, and Pedro Zuidberg Dos Martires. Klay: Accelerating arithmetic circuits for neurosymbolic ai. In *ICLR*, 2025.

[104] Robin Manhaeve, Stefano Colamonaco, Vincent Derkinderen, Rik Adriaensen, Lucas Van Praet, Luc De Raedt, and Giuseppe Marra. Deeplog: A software framework for modular neurosymbolic AI. In *IJCAI/ECAI*, pages 8462–8465. ijcai.org, 2026.

[105] Emile Van Krieken, Samy Badreddine, Robin Manhaeve, and Eleonora Giunchiglia. Uller: A unified language for learning and reasoning. In *International Conference on Neural-Symbolic Learning and Reasoning*, pages 219–239. Springer, 2024.

[106] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

[107] Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, and Kristian Kersting. Learning differentiable logic programs for abstract visual reasoning. *Machine Learning*, 113(11): 8533–8584, 2024.

[108] Hao Tang and Kevin Ellis. From perception to programs: regularize, overparameterize, and amortize. In *ICML*, 2023.

[109] Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3):822–852, 2022.

[110] Andrea Pugnana, Giovanni De Toni, Cesare Barbera, Roberto Pellungrini, Bruno Lepri, and Andrea Passerini. To ask or not to ask: Learning to require human feedback. *arXiv preprint arXiv:2510.08314*, 2025.

[111] Nick W Koning. Post-hoc and anytime valid permutation and group invariance testing. *arXiv preprint arXiv:2310.01153*, 2023.

[112] Alexander A. Balinsky and Alexander David Balinsky. Enhancing conformal prediction using e-test statistics. In *COPA*, volume 230 of *Proceedings of Machine Learning Research*, pages 65–72. PMLR, 2024.

[113] Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, 1994. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(94)90072-8. URL https://www.sciencedirect.com/science/article/pii/0004370294900728.

[114] Tommaso Carraro, Luciano Serafini, and Fabio Aiolli. Ltntorch: Pytorch implementation of logic tensor networks. *arXiv preprint arXiv:2409.16045*, 2024.

[115] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[116] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[117] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. In *ICML*, 2023.

[118] Yiran Xu, Xiaoyin Yang, Lihang Gong, Hsuan-Chu Lin, Tz-Ying Wu, Yunsheng Li, and Nuno Vasconcelos. Explainable object-induced action decision for autonomous vehicles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[119] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[120] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.

[121] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[122] Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951. URL https://api.semanticscholar.org/CorpusID:16945044.

[123] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[124] Shuhei Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*, 2023.

[125] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[126] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[127] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

A Extended One-Sided Strategies and Prior Work

COCOCO combines two design choices: (i) constructing both a label set $\Upsilon_\alpha(\mathbf{x})$ and a concept set $\Gamma_\beta(\mathbf{x})$, and (ii) revising each via the logical image of the other. A natural question is whether either choice can be relaxed without sacrificing the desiderata. In this section we systematically ablate COCOCO along these two axes, showing that each one-sided strategy fails at least one desideratum. The ablations also subsume prior work, i.e., Ledaguenel et al. [61] strategy corresponds to applying CP on the label side, while Ramalingam et al. [57] "compositional," "direct," and "full" can be matched in three of our ablations (CDe, TO and RPB, respectively). Table 3 summarizes the comparison.

	Consistency (**D1**)		Coverage (**D2**)		Conciseness (**D3**)	
Method	Task	Concepts	Task $\mathbf{Y}$	Concepts $\mathbf{C}$	Task $\mathbf{Y}$	Concepts $\mathbf{C}$
Task Only (TO) [61]	✗	–	✓ (4)	✗	✗	–
Task plus Abduction (TAb)	✓	✓	✓ (4)	✓ (3.2)	✗	✗
Concepts Only (CO)	–	✗	✗	✓ (3.1)	–	✗
Concepts plus Deduction (CDe)	✓	✓	✓ (3.3)	✓ (3.1)	✗	✗
Joint (RPB) [57]	✓	✗	✓	✓	✓	✗
COCOCO	✓ (B.7)	✓ (B.7)	✓ (4.2)	✓ (4.2)	✓ (B.8, 6.2)	✓ (B.8, 6.2)

Table 3: Summary of guarantees provided by different NeSy conformal strategies.

One-side CP. The simplest strategies apply CP to only one level: TO: Ledaguenel et al. [61] apply CP only to the label distribution to obtain $\Upsilon_\alpha(\mathbf{x})$, and predict the single argmax concept $\hat{\mathbf{c}}$; this also corresponds to the "direct" approach of Ramalingam et al. [57]. Label coverage is satisfied by construction, but the concept side has no conformalization (**D2**), and the predicted concept may not entail *all* labels in $\Upsilon_\alpha(\mathbf{x})$, breaking consistency (**D1**). CO symmetrically applies CP only to the concepts (via $\mathbf{\Gamma}^\times(\mathbf{x})$) and predicts the single argmax label. The label side is unconformalized and inconsistencies arise in the opposite direction. Moreover, Bonferroni's correction (Corollary B.4) inflates $\mathbf{\Gamma}^\times(\mathbf{x})$ as the number of concepts k grows, breaking conciseness (**D3**).

Cross-Level CP without revision. A natural improvement is to use one of $g^{-\dagger}, g^\dagger$ to derive the missing side from the conformalized one, but without revising both jointly. TAb reports the label set $\Upsilon_\alpha(\mathbf{x})$ and the abduced concept set $\Gamma^{\mathsf{ab}}(\mathbf{x}) = g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$. Both sets are consistent by construction (**D1**) and inherit coverage from $\Upsilon_\alpha(\mathbf{x})$ via Proposition 3.2 (**D2**). However, $\Gamma^{\mathsf{ab}}(\mathbf{x})$ contains *every* concept vector entailing some label in $\Upsilon_\alpha(\mathbf{x})$, including low-probability ones under f, so it is potentially much larger than the input-aware $\Gamma_\beta(\mathbf{x})$, breaking conciseness (**D3**). CDe (the "compositional" approach of Ramalingam et al. [57]) is the symmetric strategy: report $\Gamma_\beta(\mathbf{x})$ and the deduced label set $\Upsilon^{\mathsf{de}}(\mathbf{x}) = g^\dagger(\Gamma_\beta(\mathbf{x}))$. Coverage (Proposition 3.3) and consistency hold, but $\Upsilon^{\mathsf{de}}(\mathbf{x})$ inflates when $\Gamma_\beta(\mathbf{x})$ contains low-probability concepts whose images inflate $\mathcal{Y}$, breaking **D3** on the label side.

One-Sided Joint Revision: The closest competitor, RPB [57] (their "full" approach), can be represented in our framework as a revision applied only to the label side:

$$\Upsilon^{\mathsf{rev}}(\mathbf{x})^{\mathsf{RPB}} = \Upsilon_\alpha(\mathbf{x}) \cap g^\dagger(\Gamma_\beta(\mathbf{x})), \qquad \Gamma_\beta(\mathbf{x})^{\mathsf{RPB}} = \Gamma_\beta(\mathbf{x}) \quad \text{(unchanged)}. \tag{13}$$

This delivers label-side consistency and matches COCOCO label set by construction. However, the concept set is left untouched as it still contains concept vectors that entail labels *outside* $\Upsilon^{\mathsf{rev}}(\mathbf{x})^{\mathsf{RPB}}$, breaking concept-side consistency (**D1**). Empirically (Section 6), this manifests as concept-side consistency dropping to ≈ 0.08–0.94 across our datasets, while COCOCO attains 1.00 throughout.

Discussion: On the one hand, $\Upsilon_\alpha(\mathbf{x})$ and $\Gamma_\beta(\mathbf{x})$, used in isolation, leave the other side unconformalized, so stakeholders cannot tell which concepts (resp. labels) can be trusted, violating coverage (**D2**), and the predicted side may not entail or abduce *all* elements of the conformalized one, violating consistency (**D1**). On the other hand, $\Gamma^{\mathsf{ab}}(\mathbf{x})$ and $\Upsilon^{\mathsf{de}}(\mathbf{x})$ ensure concept-side (resp. label-side) consistency by design, but potentially inflate the resulting set, $\Gamma^{\mathsf{ab}}(\mathbf{x})$ contains every concept vector $\mathbf{c}$ entailing some label in $\Upsilon_\alpha(\mathbf{x})$, and $\Upsilon^{\mathsf{de}}(\mathbf{x})$ every label deducible from some concept in $\Gamma_\beta(\mathbf{x})$, violating conciseness (**D3**). Section E.6 analyzes these failure modes in detail, exploring all combinations of these four objects, including those of Ledaguenel et al. [61] and Ramalingam et al. [57] as special cases, and shows that none satisfies the three desiderata simultaneously.

B Proofs

Notation and Standing Assumptions

We recall the definition of $g^\dagger$ instead is

$$g^\dagger := \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, g(\mathbf{c}) \subseteq \mathcal{Y}\,, \tag{14}$$

and its inverse $g^{-\dagger}$ when it is applied to a subset $\mathcal{Y}_s \subseteq \mathcal{Y}$

$$g^{-\dagger}(\mathbf{y}) = \{\mathbf{c}\in\mathcal{C} : \mathbf{y} \in \operatorname{argmax} g(\mathbf{c})\}, \quad g^{-\dagger}(\mathcal{Y}_s) = \bigcup_{\mathbf{y}\in\mathcal{Y}_s} \{\mathbf{c}\in\mathcal{C} : \mathbf{y} \in \operatorname{argmax} g(\mathbf{c})\}. \tag{15}$$

We also recall that for a given $0 < \alpha < 1$ and every pair $(\mathbf{x}, \mathbf{y}^*) \in \mathcal{D}_{\mathbf{Y}}$, we define the ***label set*** $\Upsilon_\alpha(\mathbf{x}) \subseteq \mathcal{Y}$ that satisfies

$$\Pr(\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})) \geq 1-\alpha. \tag{16}$$

Similarly, for $0 < \beta < 1$ and $(\mathbf{x}, \mathbf{c}^*) \in \mathcal{D}_{\mathbf{C}}$, we define the ***concept set*** $\Gamma_\beta(\mathbf{x}) \subset \mathcal{C}$ that satisfies

$$\Pr(\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})) \geq 1-\beta. \tag{17}$$

To make self-contained this appendix without loss of generality, we prove the main statements in the text, providing equivalent statements with the material here recollected.

B.1 Proof of abductive and deductive assumptions (Definition 3.1)

Proposition B.1 (Deductive soundness implies abductive soundness)**.** *For any NeSy-CBM (f, g), deductive soundness implies abductive soundness (Definition 3.1). Consequently, a model that satisfies deductive soundness but not abductive soundness cannot exist.*

Proof. Suppose the NeSy-CBM satisfies deductive soundness, i.e., $g^\dagger(\mathbf{c}^*) = \mathbf{y}^*$ for every $(\mathbf{c}^*, \mathbf{y}^*)$ in $\operatorname{supp}(p^*)$. Then $g^\dagger(\mathbf{c}^*) = \mathbf{y}^*$ means $\mathbf{c}^*$ belongs to $g^{-\dagger}(\mathbf{y}^*) = \{\mathbf{c}\in\mathcal{C} : \operatorname{argmax} g(\mathbf{c}) = \mathbf{y}^*\}$, which is exactly abductive soundness (Definition 3.1). □

Proposition B.2 (Abductive soundness implies $\delta_{\text{ab}} = 1$)**.** *Let $\Gamma^{\mathsf{ab}}(\mathbf{x}) := g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$ be the concept set obtained by abduction according to Eq. (15). If the NeSy-CBM satisfies abductive soundness (Definition 3.1), then*

$$\delta_{\text{ab}} := \Pr\big(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) \mid \mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})\big) = 1. \tag{18}$$

Proof. From Definition 3.1, abductive soundness ensures that $\mathbf{c}^* \in g^{-\dagger}(\mathbf{y}^*)$ for every $(\mathbf{c}^*, \mathbf{y}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$. Suppose $\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})$. Because $\Gamma^{\mathsf{ab}}(\mathbf{x}) = g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})) = \bigcup_{\mathbf{y}\in\Upsilon_\alpha(\mathbf{x})} g^{-\dagger}(\mathbf{y})$ and $\mathbf{c}^* \in g^{-\dagger}(\mathbf{y}^*)$ with $\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})$, we have $\mathbf{c}^* \in g^{-\dagger}(\mathbf{y}^*) \subseteq \Gamma^{\mathsf{ab}}(\mathbf{x})$. Hence, the event $\{\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})\}$ holds surely whenever $\{\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})\}$ holds, giving

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) \mid \mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})) = 1. \tag{19}$$

□

Proposition B.3 (Deductive soundness implies $\delta_{\text{de}} = 1$)**.** *Let $\Upsilon^{\mathsf{de}}(\mathbf{x}) := g^\dagger(\Gamma_\beta(\mathbf{x}))$ be the label set obtained by the image of g on $\Gamma_\beta(\mathbf{x})$. If the NeSy-CBM satisfies deductive soundness (Definition 3.1), then*

$$\delta_{\text{de}} := \Pr\big(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x}) \mid \mathbf{c}^* \in \Gamma_\beta(\mathbf{x})\big) = 1. \tag{20}$$

Proof. Deductive soundness states that $g^\dagger(\mathbf{c}^*) = \mathbf{y}^*$ for every $(\mathbf{c}^*, \mathbf{y}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$. Suppose $\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})$. Then $g^\dagger(\mathbf{c}^*) \in g^\dagger(\Gamma_\beta(\mathbf{x})) = \Upsilon^{\mathsf{de}}(\mathbf{x})$. Since $g^\dagger(\mathbf{c}^*) = \mathbf{y}^*$, we have $\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})$. Hence the event $\{\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})\}$ holds surely whenever $\{\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})\}$ holds, giving

$$\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x}) \mid \mathbf{c}^* \in \Gamma_\beta(\mathbf{x})) = 1. \tag{21}$$

□

Remark 1. Propositions B.2 and B.3 are central to give tighter bounds stated in Propositions 3.2, 3.3 and 4.2 when $\delta_{\text{ab}} = 1$ (resp. $\delta_{\text{de}} = 1$) in the general bounds, this yields the $1-\alpha$ (resp. $1-\beta$) guarantees.

B.2 Coverage proof of TAb (Proposition 3.2)

Proposition (3.2, Marginal coverage of $\Gamma^{\mathsf{ab}}(\mathbf{x})$)**.** *For $0 < \alpha < 1$ and for any $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$, suppose $\Upsilon_\alpha(\mathbf{x})$ satisfies Eq. (16). Let $\Gamma^{\mathsf{ab}}(\mathbf{x}) := g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$. Then*

$$(1-\alpha)\delta_{\mathrm{ab}} \leq \Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})) \leq \alpha + \delta_{\mathrm{ab}} \tag{22}$$

where $\delta_{\mathrm{ab}} := \Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) \mid \mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x}))$ is the probability that abduction recovers the ground-truth concepts $\mathbf{c}^*$ given that the label set $\Upsilon_\alpha(\mathbf{x})$ contains the correct label $\mathbf{y}^*$. Moreover, if the NeSy-CBM satisfies abductive soundness (Definition 3.1), the bound tightens to

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})) \geq 1 - \alpha. \tag{23}$$

Proof. Let the two binary events $A = (\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x}))$ and $B = (\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}))$. We first derive the lower bound. By the law of total probability,

$$\begin{aligned} \Pr(B) &= \Pr(B \mid A)\Pr(A) + \Pr(B \mid \neg A)\Pr(\neg A) && (24)\\ &\geq \Pr(B \mid A)\Pr(A) && (25) \end{aligned}$$

Since $\Upsilon_\alpha(\mathbf{x})$ is a conformal label set with miscoverage α, we have that, by Eq. (4):

$$\Pr(A) \geq 1 - \alpha \tag{26}$$

We can plug this result into the previous one, i.e., $\Pr(B) \geq \Pr(B \mid A)\Pr(A)$ (Eq. (25)), to obtain

$$\begin{aligned} &\Pr(B) \geq \Pr(B|A)(1-\alpha) && (27)\\ & && (28) \end{aligned}$$

Which proofs the lower bound:

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})) \geq \Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) \mid \mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x}))(1-\alpha). \tag{29}$$

Next, we prove the upper bound. Consider again the right-hand side of Eq. (24). Since $\Pr(B \mid \neg A) \leq 1$ and $\Pr(\neg A) \leq \alpha$, we have

$$\Pr(B \mid \neg A)\Pr(\neg A) \leq \alpha. \tag{30}$$

Moreover, $\Pr(A) \leq 1$. Hence,

$$\Pr(B \mid A)\Pr(A) \leq \Pr(B \mid A). \tag{31}$$

Combining these observations yields:

$$\begin{aligned} \Pr(B) &= \Pr(B \mid A)\Pr(A) + \Pr(B \mid \neg A)\Pr(\neg A) && \text{(Law of total probability - Eq. (24))}\\ &\leq \Pr(B \mid A) + \Pr(B \mid \neg A)\Pr(\neg A) && \text{(Substitute the left term - Eq. (31))}\\ &\leq \Pr(B \mid A) + \alpha && \text{(Substitute the right term - Eq. (30))} \end{aligned}$$

Substituting the expression for A and B, we have the upper bound

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x})) \leq \alpha + \delta_{\mathrm{ab}}. \tag{32}$$

Under *abductive soundness* (Definition 3.1), by Proposition B.2 we have $\delta_{\mathrm{ab}} = 1$. Hence, we get

$$\Pr(B) \geq \delta_{\mathrm{ab}}(1-\alpha) \geq 1 - \alpha\,. \tag{33}$$

While for the upper bound Eq. (30)

$$\Pr(B) \leq \delta_{\mathrm{ab}} + \alpha \leq \min(1, \delta_{\mathrm{ab}} + \alpha)\,. \tag{34}$$

This concludes the proof. □

B.3 Coverage proof of Product Set (Proposition 3.1)

For $0 < \beta_j < 1$, with $j \in \{1, \dots k\}$, and for any $(\mathbf{x}, \mathbf{c}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$, we recall the definition of individual ***concept sets*** $\Gamma^j_{\beta_j}(\mathbf{x})$ that satisfy:

$$\Pr(\mathbf{c}^*_j \in \Gamma^j_{\beta_j}(\mathbf{x})) \geq 1 - \beta_j \,. \tag{35}$$

The cartesian product set is given by

$$\Gamma_\beta(\mathbf{x}) = \bigtimes_{j=1}^{k} \Gamma^j_{\beta_j}(\mathbf{x}). \tag{36}$$

Proposition (Marginal coverage of Product Set 3.1)**.** *, For any NeSy CBM (f, g), let $0 < \beta_j < 1$ and $0 < \beta < 1$ such that $\beta = \sum_{j=1}^{k} \beta_j$. If $\Gamma^j_{\beta_j}(\mathbf{x})$, for $j \in [1, \dots, k]$, are concept conformal sets with miscoverage level β_j, respectively (Eq. (6)), then let $\mathbf{\Gamma}^\times(\mathbf{x})$ be the set constructed as in Eq. (5) based on the k sets $\Gamma^j_{\beta_j}(\mathbf{x})$. Then, for every $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*) \in \operatorname{supp}(p^*(\mathbf{x}, \mathbf{c}, \mathbf{y}))$:*

$$\Pr(\mathbf{c}^* \in \mathbf{\Gamma}^\times(\mathbf{x})) \geq 1 - \beta. \tag{37}$$

Proof. By construction, we have $\mathbf{c}^* \in \mathbf{\Gamma}^\times(\mathbf{x})$ if and only if $\mathbf{c}^*_j \in \Gamma^j_{\beta_j}(\mathbf{x})$ for all $j \in [k]$. Therefore, the probability that the joint vector $\mathbf{c}^*$ does not belong to $\mathbf{\Gamma}^\times(\mathbf{x})$ can be written as

$$\Pr(\mathbf{c}^* \notin \mathbf{\Gamma}^\times(\mathbf{x})) = \Pr\left(\bigcup_{j=1}^{k} \{\mathbf{c}^*_j \notin \Gamma^j_{\beta_j}(\mathbf{x})\} \right). \tag{38}$$

Applying the union bound (Boole's inequality) yields

$$\Pr\left(\bigcup_{j=1}^{k} \{\mathbf{c}^*_j \notin \Gamma^j_{\beta_j}(\mathbf{x})\} \right) \leq \sum_{j=1}^{k} \Pr(\mathbf{c}^*_j \notin \Gamma^j_{\beta_j}(\mathbf{x})) \leq \sum_{j=1}^{k} \beta_j, \tag{39}$$

where the last inequality uses the individual coverage guarantees. Subtracting from 1, we obtain

$$\Pr(\mathbf{c}^* \in \mathbf{\Gamma}^\times(\mathbf{x})) = 1 - \Pr(\mathbf{c}^* \notin \mathbf{\Gamma}^\times(\mathbf{x})) \geq 1 - \sum_{i=j}^{k} \beta_j. \tag{40}$$

□

Corollary B.4 (Bonferroni's correction)**.** *If we desire joint coverage at level $1-\beta$, using Bonferroni's correction, i.e., setting $\beta_j = \beta/k$ for all j guarantees*

$$\Pr(\mathbf{c}^* \in \mathbf{\Gamma}^\times(\mathbf{x})) \geq 1 - \sum_{i=1}^{k} \frac{\beta}{k} = 1 - \beta. \tag{41}$$

This follows directly from the proposition above. □

B.4 Coverage proof of CDe (Proposition 3.3)

Proposition (Marginal coverage of CDe Proposition 3.3)**.** *For $0 < \beta < 1$ and for any $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$, suppose $\Gamma_\beta(\mathbf{x})$ satisfies Eq. (17). Let $\Upsilon^{\mathsf{de}}(\mathbf{x}) := g^\dagger(\Gamma_\beta(\mathbf{x}))$. Then*

$$(1 - \beta)\delta_{\mathrm{de}} \leq \Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})) \leq \beta + \delta_{\mathrm{de}} \tag{42}$$

where $\delta_{\mathrm{de}} := \Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x}) \mid \mathbf{c}^* \in \Gamma_\beta(\mathbf{x}))$ is the probability that deduction recovers the ground-truth labels $\mathbf{y}^*$ given that the concept set $\Gamma_\beta(\mathbf{x})$ contains the correct concepts $\mathbf{c}^*$. Moreover, if the NeSy-CBM satisfies deductive soundness (Definition 3.1), the bound tightens to:

$$\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})) \geq 1 - \beta. \tag{43}$$

Proof. Let the two binary events $A = (\mathbf{c}^* \in \Gamma_\beta(\mathbf{x}))$ and $B = (\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x}))$. We first derive the lower bound. By the law of total probability Eq. (24)

$$\Pr(B) = \Pr(B \mid A)\Pr(A) + \Pr(B \mid \neg A)\Pr(\neg A) \tag{44}$$
$$\geq \Pr(B \mid A)\Pr(A) \tag{45}$$

Since $\Gamma_\beta(\mathbf{x})$ is a conformal concept set with miscoverage β by Eq. (17), we have that

$$\Pr(A) = \Pr(\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})) \geq 1 - \beta. \tag{46}$$

We can plug this result into the previous one, i.e., $\Pr(B) \geq \Pr(B \mid A)\Pr(A)$ (Eq. (45)), to obtain

$$\Pr(B) \geq \Pr(B \mid A)(1 - \beta) \tag{47}$$

Which proofs the lower-bound:

$$\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})) \geq \Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x}) \mid \mathbf{c}^* \in \Gamma_\beta(\mathbf{x}))(1 - \beta). \tag{48}$$

Next, we prove the upper bound. Consider again the right-hand side of Eq. (44). Since $\Pr(B \mid \neg A) \leq 1$ and $\Pr(\neg A) \leq \beta$, we have:

$$\Pr(B \mid \neg A)\Pr(\neg A) \leq \beta. \tag{49}$$

Moreover, $\Pr(A) \leq 1$. Hence,

$$\Pr(B \mid A)\Pr(A) \leq \Pr(B \mid A). \tag{50}$$

Combining these observations yields:

$$\begin{aligned}
\Pr(B) &= \Pr(B \mid A)\Pr(A) + \Pr(B \mid \neg A)\Pr(\neg A) && \text{(Law of total probability - Eq. (24))} \\
&\leq \Pr(B \mid A) + \Pr(B \mid \neg A)\Pr(\neg A) && \text{(Substitute the left term - Eq. (50))} \\
&\leq \Pr(B \mid A) + \beta && \text{(Substitute the right term - Eq. (49))}
\end{aligned}$$

Substituting the expression for A and B, we have the upper bound

$$\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{de}}(\mathbf{x})) \leq \beta + \delta_{\mathrm{de}} \tag{51}$$

Under *deductive soundness* (Definition 3.1), by Proposition 3.3, we have $\delta_{\mathrm{de}} = 1$. Hence, we get

$$\Pr(B) \geq \delta_{\mathrm{de}}(1 - \beta) \geq 1 - \beta. \tag{52}$$

While for the upper bound Eq. (49):

$$\Pr(B) \leq \delta_{\mathrm{de}} + \beta \leq \min(1, \delta_{\mathrm{de}} + \beta). \tag{53}$$

This concludes the proof.

□

B.5 Coverage proofs of COCOCO concept set (Proposition 4.2)

Proposition (Marginal coverage of $\Gamma^{\mathsf{rev}}(\mathbf{x})$ (Proposition 4.2))**.** *For $0 < \alpha < 1$ and $0 < \beta < 1$, and for any $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$, suppose $\Upsilon_\alpha(\mathbf{x})$ satisfies Eq. (16) and $\Gamma_\beta(\mathbf{x})$ satisfies Eq. (17). Let $\Upsilon^{\mathsf{de}}(\mathbf{x}) := g^\dagger(\Gamma_\beta(\mathbf{x}))$ and $\Gamma^{\mathsf{ab}}(\mathbf{x}) := g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$. Define the* **refined concept set** *as $\Gamma^{\mathsf{rev}}(\mathbf{x}) := \Gamma_\beta(\mathbf{x}) \cap \Gamma^{\mathsf{ab}}(\mathbf{x})$. Then*

$$\begin{aligned}
\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) \geq (1 - \alpha)\ &\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) \mid \mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})) - \beta \\
&+ \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x}))
\end{aligned} \tag{54}$$

Proof. Let $A := (\mathbf{c}^* \in \Gamma_\beta(\mathbf{x}))$ and $B := (\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}))$, so that

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) = \Pr(A \cap B). \tag{55}$$

By the inclusion-exclusion identity, we have

$$\Pr(A \cap B) = \Pr(A) + \Pr(B) - \Pr(A \cup B). \tag{56}$$

Since $\Pr(A \cup B) = 1 - \Pr(\neg A \cap \neg B)$, this is equivalent to

$$\Pr(A \cap B) = \Pr(A) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B). \tag{57}$$

We know that $\Pr(A) \geq 1 - \beta$ follows from Eq. (17), and that $\Pr(B) \geq (1 - \alpha)\delta_{\text{ab}}$ follows from Proposition 3.2. Therefore, we can proceed to obtain the lower bound as follows:

$$\begin{aligned}
\Pr(\mathbf{c}^* \in \Gamma^{\text{rev}}(\mathbf{x})) &= \Pr(A \cap B) && \text{(def. of } \Gamma^{\text{rev}}(\mathbf{x})\text{)} \\
&= \Pr(A) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B) && \text{(inclusion–exclusion Eq. (57))} \\
&\geq (1 - \beta) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B) && \text{(Eq. (17))} \\
&\geq (1 - \beta) + (1 - \alpha)\delta_{\text{ab}} - 1 + \Pr(\neg A \cap \neg B) && \text{(Proposition 3.2)} \\
&= (1 - \alpha)\delta_{\text{ab}} - \beta + \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\text{ab}}(\mathbf{x})). && \text{(subst. } \Pr(\neg A \cap \neg B)\text{)}
\end{aligned}$$

Under *abductive soundness* (Definition 3.1), we have $\delta_{\text{ab}} = 1$. Hence, we get

$$\Pr(\mathbf{c}^* \in \Gamma^{\text{rev}}(\mathbf{x})) \geq 1 - \alpha - \beta + \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\text{ab}}(\mathbf{x})). \tag{58}$$

This concludes the proof.

□

Remark 2. The term $\Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\text{ab}}(\mathbf{x}))$ is the probability that the ground-truth concept $\mathbf{c}^*$ is not captured by *neither* the concept conformal set ($\Gamma_\beta(\mathbf{x})$) *nor* the abduced concept set ($\Gamma^{\text{ab}}(\mathbf{x})$).

B.6 Coverage proofs of COCOCO label set (Proposition 4.2)

Proposition (Marginal coverage of $\Upsilon^{\text{rev}}(\mathbf{x})$ (Proposition 4.2))**.** *For $0 < \alpha < 1$ and $0 < \beta < 1$, and for any $(\mathbf{x}, \mathbf{c}^*, \mathbf{y}^*)$ in the support of $p^*(\mathbf{x}, \mathbf{c}, \mathbf{y})$, suppose $\Upsilon_\alpha(\mathbf{x})$ satisfies Eq. (16) and $\Gamma_\beta(\mathbf{x})$ satisfies Eq. (17). Let $\Upsilon^{\text{de}}(\mathbf{x}) := g^\dagger(\Gamma_\beta(\mathbf{x}))$ and $\Gamma^{\text{ab}}(\mathbf{x}) := g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$. Define the* **refined label set** *as $\Upsilon^{\text{rev}}(\mathbf{x}) = \Upsilon_\alpha(\mathbf{x}) \cap \Upsilon^{\text{de}}(\mathbf{x})$. Then:*

$$\begin{aligned}
\Pr(\mathbf{y}^* \in \Upsilon^{\text{rev}}(\mathbf{x})) \geq (1 - \beta)\ &\Pr(\mathbf{y}^* \in \Upsilon^{\text{de}}(\mathbf{x}) \mid \mathbf{c}^* \in \Gamma_\beta(\mathbf{x})) - \alpha \\
&+ \Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\text{de}}(\mathbf{x})).
\end{aligned} \tag{59}$$

Proof. Let $A := (\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x}))$ and $B := (\mathbf{y}^* \in \Upsilon^{\text{de}}(\mathbf{x}))$, so that

$$\Pr(\mathbf{y}^* \in \Upsilon^{\text{rev}}(\mathbf{x})) = \Pr(A \cap B). \tag{60}$$

By the inclusion–exclusion identity,

$$\Pr(A \cap B) = \Pr(A) + \Pr(B) - \Pr(A \cup B). \tag{61}$$

Since $\Pr(A \cup B) = 1 - \Pr(\neg A \cap \neg B)$, this is equivalent to

$$\Pr(A \cap B) = \Pr(A) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B). \tag{62}$$

We know that $\Pr(A) \geq 1 - \alpha$ follows from Eq. (16), and that $\Pr(B) \geq (1 - \beta)\delta_{\text{de}}$ follows from Proposition 3.3. Therefore:

$$\begin{aligned}
\Pr(\mathbf{y}^* \in \Upsilon^{\text{rev}}(\mathbf{x})) &= \Pr(A \cap B) && \text{(def. of } \Upsilon^{\text{rev}}(\mathbf{x})\text{)} \\
&= \Pr(A) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B) && \text{(inclusion–exclusion Eq. (62))} \\
&\geq (1 - \alpha) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B) && \text{(Eq. (16))} \\
&\geq (1 - \alpha) + (1 - \beta)\delta_{\text{de}} - 1 + \Pr(\neg A \cap \neg B) && \text{(Proposition 3.3)} \\
&= (1 - \beta)\delta_{\text{de}} - \alpha + \Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\text{de}}(\mathbf{x})). && \text{(subst. } \Pr(\neg A \cap \neg B)\text{)}
\end{aligned}$$

Under *deductive soundness* (Definition 3.1), we have $\delta_{\text{de}} = 1$. Hence, we get

$$\Pr(\mathbf{y}^* \in \Upsilon^{\text{rev}}(\mathbf{x})) \geq 1 - \alpha - \beta + \Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\text{de}}(\mathbf{x})). \tag{63}$$

This concludes the proof. □

Remark 3. The term $\Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\text{de}}(\mathbf{x}))$ is the probability that the ground-truth label $\mathbf{y}^*$ is captured by *neither* the label conformal set ($\Upsilon_\alpha(\mathbf{x})$) *nor* the deduced label set ($\Upsilon^{\text{de}}(\mathbf{x})$).

B.7 Fixed point proof of COCOCO (Proposition B.7)

To prove that COCOCO does not require further iterations, let R be the operator that *revises* the label and concept sets. Formally:

$$R : (\Gamma, \Upsilon) \mapsto (R(\Gamma), R(\Upsilon)) \subseteq \mathcal{C} \times \mathcal{Y}, \qquad R(\Gamma, \Upsilon) := \big(\Gamma \cap g^{-\dagger}(\Upsilon), \Upsilon \cap g^{\dagger}(\Gamma)\big). \tag{64}$$

By construction, COCOCO applies R **only once** to the pair $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$, producing

$$(\Gamma^{\text{rev}}(\mathbf{x}), \Upsilon^{\text{rev}}(\mathbf{x})) = R(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x})) \tag{65}$$

Before introducing the main proposition, we show two useful lemmas.

Lemma B.5 (Set behavior of $g^{-\dagger}$ on intersection)**.** *For any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{Y}$,*

$$g^{-\dagger}(\mathcal{A} \cap \mathcal{B}) \;\subseteq\; g^{-\dagger}(\mathcal{A}) \cap g^{-\dagger}(\mathcal{B}). \tag{66}$$

The reverse inclusion does not *hold in general for $g^{\dagger}$.*

Proof. Let $\mathbf{c} \in g^{-\dagger}(\mathcal{A} \cap \mathcal{B})$, there exists $\mathbf{y} \in g^{\dagger}(\mathbf{c}) \cap (\mathcal{A} \cap \mathcal{B})$. Then $\mathbf{y} \in g^{\dagger}(\mathbf{c}) \cap \mathcal{A}$, so $\mathbf{c} \in g^{-\dagger}(\mathcal{A})$, and likewise $\mathbf{c} \in g^{-\dagger}(\mathcal{B})$. Hence $\mathbf{c} \in g^{-\dagger}(\mathcal{A}) \cap g^{-\dagger}(\mathcal{B})$.

The reverse inclusion fails when ties are present: if $g^{\dagger}(\mathbf{c}) = \{y_1, y_2\}$ with $y_1 \in \mathcal{A} \setminus \mathcal{B}$ and $y_2 \in \mathcal{B} \setminus \mathcal{A}$, then $\mathbf{c} \in g^{-\dagger}(\mathcal{A}) \cap g^{-\dagger}(\mathcal{B})$ but $\mathbf{c} \notin g^{-\dagger}(\mathcal{A} \cap \mathcal{B})$. □

Lemma B.6. *For any $\Gamma' \subseteq \mathcal{C}$,*

$$\Gamma' \subseteq g^{-\dagger}\big(g^{\dagger}(\Gamma')\big). \tag{67}$$

Proof. Let $\mathbf{c} \in \Gamma'$. By definition of the image, $g^{\dagger}(\mathbf{c}) \in g^{\dagger}(\Gamma')$. By Eq. (15),

$$g^{-\dagger}\big(g^{\dagger}(\Gamma')\big) = \bigcup_{\mathbf{y} \in g^{\dagger}(\Gamma')} \{\mathbf{c}' \in \mathcal{C} : \mathbf{y} \in \operatorname{argmax} g(\mathbf{c}')\}, \tag{68}$$

so for any $\mathbf{c}' \in \mathcal{C}$,

$$g^{\dagger}(\mathbf{c}') \in g^{\dagger}(\Gamma') \implies \mathbf{c}' \in g^{-\dagger}\big(g^{\dagger}(\Gamma')\big). \tag{69}$$

In particular, $\Gamma' \subseteq \mathbf{g}^{-\dagger}(\mathbf{g}^{\dagger}(\Gamma'))$. □

Now we can proceed with the main proposition.

Proposition B.7 (COCOCO reaches a fixed point)**.** *For a NeSy-CBM (f, g), let R be defined as Eq. (64). Then, for any $\Gamma \subseteq \mathcal{C}$ and $\Upsilon \subseteq \mathcal{Y}$,*

$$R\big(R(\Gamma, \Upsilon)\big) = R(\Gamma, \Upsilon)\,. \tag{70}$$

In particular, $(\Gamma^{\text{rev}}(\mathbf{x}), \Upsilon^{\text{rev}}(\mathbf{x}))$ is a **fixed point** *of R, and no iterative refinement is needed.*

Proof. Let $(\Gamma_1, \Upsilon_1) = R(\Gamma, \Upsilon)$. By definition of R, these are

$$\Gamma_1 = \Gamma \cap g^{-\dagger}(\Upsilon), \qquad \Upsilon_1 = \Upsilon \cap g^{\dagger}(\Gamma)\,. \tag{71}$$

Now, denote with (Γ_2, Υ_2) the sets obtained by applying R twice on Γ, Υ, namely

$$(\Gamma_2, \Upsilon_2) = R(\Gamma_1, \Upsilon_1) = R(R(\Gamma, \Upsilon))\,, \tag{72}$$

which are given by

$$\Gamma_2 = \Gamma_1 \cap g^{-\dagger}(\Upsilon_1), \qquad \Upsilon_2 = \Upsilon_1 \cap g^{\dagger}(\Gamma_1). \tag{73}$$

Now, to proceed with the proof, we use the standard set identity. For any two sets S, T,

$$S \subseteq T \iff S \cap T = S. \tag{74}$$

This means that, following Eq. (74):

$$(I) \qquad \Gamma_2 = \Gamma_1 \iff \Gamma_1 \subseteq g^{-\dagger}(\Upsilon_1) = g^{-\dagger}(\Upsilon \cap g^{\dagger}(\Gamma)) \tag{75}$$

$$(II) \qquad \Upsilon_2 = \Upsilon_1 \iff \Upsilon_1 \subseteq g^{\dagger}(\Gamma_1) = g^{\dagger}(\Gamma \cap g^{-\dagger}(\Upsilon)). \tag{76}$$

To prove the fixed point, we prove these two inclusions in turn.

Proof of (I): $\Gamma_1 \subseteq g^{-\dagger}\big(\Upsilon \cap g^{\dagger}(\Gamma)\big)$.

Let $\mathbf{c} \in \Gamma_1$, this means that, since $\Gamma_1 = \Gamma \cap g^{-\dagger}(\Upsilon)$, $\mathbf{c} \in \Gamma$ and $\mathbf{c} \in g^{-\dagger}(\Upsilon)$. We prove $\mathbf{c} \in g^{-\dagger}(\Upsilon_1)$, i.e., that $g^{\dagger}(\mathbf{c}) \cap \Upsilon_1 \neq \varnothing$.

By definition of $\Gamma_1 = \Gamma \cap g^{-\dagger}(\Upsilon)$, the concept $\mathbf{c}$ satisfies

(a) $\mathbf{c} \in \Gamma$, hence $g^{\dagger}(\mathbf{c}) \subseteq g^{\dagger}(\Gamma)$;

(b) $\mathbf{c} \in g^{-\dagger}(\Upsilon)$, hence $g^{\dagger}(\mathbf{c}) \cap \Upsilon \neq \varnothing$.

Now, let $\mathbf{y}$ be any $\mathbf{y} \in (g^{\dagger}(\mathbf{c}) \cap \Upsilon) \neq \varnothing$ from (b). By (a), $\mathbf{y} \in g^{\dagger}(\Gamma)$ as well. Therefore

$$\mathbf{y} \in g^{\dagger}(\mathbf{c}) \cap \Upsilon \cap g^{\dagger}(\Gamma) \tag{77}$$
$$\mathbf{y} \in g^{\dagger}(\mathbf{c}) \cap \Upsilon_1 \qquad (\text{from } \Upsilon_1 = \Upsilon \cap g^{\dagger}(\Gamma))$$

since $g^{\dagger}(\mathbf{c}) \cap \Upsilon_1 \neq \varnothing$, this concludes the proof of the first inclusion (I) Eq. (75).

Proof of (II): $\Upsilon_1 \subseteq g^{\dagger}(\Gamma_1) \subseteq g^{\dagger}\big(\Gamma \cap g^{-\dagger}(\Upsilon)\big)$. This proof does not follow the steps of (I): unlike $g^{-\dagger}$, the image $g^{\dagger}$ does not distribute over intersection in general, so we cannot mirror the argument above. We proceed elementwise.

Let $\mathbf{y} \in \Upsilon_1 = \Upsilon \cap g^{\dagger}(\Gamma)$. Then:

- $\mathbf{y} \in \Upsilon$, and
- $\mathbf{y} \in g^{\dagger}(\Gamma)$, so by definition of the image there exists some $\mathbf{c} \in \Gamma$ such that $\mathbf{y} \in g^{\dagger}(\mathbf{c})$.

Let us then fix $\mathbf{c}$ such that $\mathbf{y} \in g^{\dagger}(\mathbf{c})$.Now, we verify that $\mathbf{c} \in \Gamma_1 = \Gamma \cap g^{-\dagger}(\Upsilon)$:

- $\mathbf{c} \in \Gamma$ holds by the choice of $\mathbf{c}$;
- $g^{\dagger}(\mathbf{c}) = \mathbf{y} \in \Upsilon$, so by the definition of Eq. (15):

$$g^{-\dagger}(\Upsilon) = \bigcup_{\mathbf{y} \in \mathcal{C}} \{\mathbf{c} \in \Gamma : \mathbf{y} \in \operatorname{argmax} g(\mathbf{c})\} \tag{78}$$

  We can conclude that $\mathbf{c} \in g^{-\dagger}(\Upsilon)$.

Hence $\mathbf{c} \in \Gamma_1$ and $g^{\dagger}(\mathbf{c}) = \mathbf{y}$, which means $\mathbf{y} \in g^{\dagger}(\Gamma_1)$, proving the second inclusion (II) Eq. (76).

Conclusion. By (I), $\Gamma_2 = \Gamma_1 \cap g^{-\dagger}(\Upsilon_1) = \Gamma_1$. By (II), $\Upsilon_2 = \Upsilon_1 \cap g^{\dagger}(\Gamma_1) = \Upsilon_1$. Therefore $R\big(R(\Gamma, \Upsilon)\big) = (\Gamma_1, \Upsilon_1) = R(\Gamma, \Upsilon)$, i.e. (Γ_1, Υ_1) is a fixed point of R.

This concludes the proof. □

B.8 Proof of the Optimality of COCOCO

Proposition B.8 (Optimality of COCOCO)**.** *Let $\Gamma_\beta(\mathbf{x}) \subseteq \mathcal{C}$ and $\Upsilon_\alpha(\mathbf{x}) \subseteq \mathcal{Y}$ be arbitrary prediction sets, and let $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x})) = R(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$ be their joint revision.*

Then $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x}))$ is the largest pair $(\Gamma', \Upsilon') \subseteq \mathcal{C} \times \mathcal{Y}$ satisfying:

1. Containment: $\Gamma' \subseteq \Gamma_\beta(\mathbf{x})$ *and* $\Upsilon' \subseteq \Upsilon_\alpha(\mathbf{x})$.
2. Mutual coverage:
   (a) *every concept entails at least one retained label:* $\forall \mathbf{c} \in \Gamma',\ g^{\dagger}(\mathbf{c}) \cap \Upsilon' \neq \varnothing$ *(equivalently,* $\Gamma' \subseteq g^{-\dagger}(\Upsilon')$*);*
   (b) *every label is entailed by at least one retained concept:* $\forall \mathbf{y} \in \Upsilon',\ \exists \mathbf{c} \in \Gamma'$ *with* $\mathbf{y} \in g^{\dagger}(\mathbf{c})$ *(equivalently,* $\Upsilon' \subseteq g^{\dagger}(\Gamma')$*).*

Consequently,

$$|\Gamma^{\mathsf{rev}}(\mathbf{x})| \leq |\Gamma_\beta(\mathbf{x})|, \qquad |\Upsilon^{\mathsf{rev}}(\mathbf{x})| \leq |\Upsilon_\alpha(\mathbf{x})|. \tag{79}$$

Moreover, $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x}))$ *is a minimal revision of* $(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$ *that enforces logical consistency with* g*.*

Proof. **Step 1:** $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x}))$ **itself satisfies (1) and (2)**.

By definition,

$$\Gamma^{\mathsf{rev}}(\mathbf{x}) = \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})) \subseteq \Gamma_\beta(\mathbf{x}) \tag{80}$$

and

$$\Upsilon^{\mathsf{rev}}(\mathbf{x}) = \Upsilon_\alpha(\mathbf{x}) \cap g^{\dagger}(\Gamma_\beta(\mathbf{x})) \subseteq \Upsilon_\alpha(\mathbf{x}), \tag{81}$$

therefore (1) holds.

Mutual coverage (2a): $\Gamma^{\mathsf{rev}}(\mathbf{x}) \subseteq g^{-\dagger}(\Upsilon^{\mathsf{rev}}(\mathbf{x}))$.

By definition, $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x})) = R(\Gamma_\beta(\mathbf{x}), \Upsilon_\alpha(\mathbf{x}))$, so $\Gamma^{\mathsf{rev}}(\mathbf{x}) = \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$ plays the role of Γ_1 and $\Upsilon^{\mathsf{rev}}(\mathbf{x}) = \Upsilon_\alpha(\mathbf{x}) \cap g^{\dagger}(\Gamma_\beta(\mathbf{x}))$ plays the role of Υ_1 in the proof of Proposition B.7. Inclusion (I) of that proof established $\Gamma_1 \subseteq g^{-\dagger}(\Upsilon_1)$; specialising to the present sets gives $\Gamma^{\mathsf{rev}}(\mathbf{x}) \subseteq g^{-\dagger}(\Upsilon^{\mathsf{rev}}(\mathbf{x}))$.

Mutual coverage (2b): $\Upsilon^{\mathsf{rev}}(\mathbf{x}) \subseteq g^{\dagger}(\Gamma^{\mathsf{rev}}(\mathbf{x}))$.

Recall that $\Upsilon^{\mathsf{rev}}(\mathbf{x}) = \Upsilon_\alpha(\mathbf{x}) \cap g^{\dagger}(\Gamma_\beta(\mathbf{x}))$ and $\Gamma^{\mathsf{rev}}(\mathbf{x}) = \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$.

Let $\mathbf{y} \in \Upsilon^{\mathsf{rev}}(\mathbf{x})$. Since $\mathbf{y} \in g^{\dagger}(\Gamma_\beta(\mathbf{x}))$, by definition of the image there is some $\mathbf{c} \in \Gamma_\beta(\mathbf{x})$ with $\mathbf{y} \in g^{\dagger}(\mathbf{c})$. This $\mathbf{c}$ lies in $\Gamma^{\mathsf{rev}}(\mathbf{x})$ by definition of $\Gamma^{\mathsf{rev}}(\mathbf{x})$. Using $\mathbf{y} \in \Upsilon^{\mathsf{rev}}(\mathbf{x}) \subseteq \Upsilon_\alpha(\mathbf{x})$ from (1),

$$\mathbf{y} \in g^{\dagger}(\mathbf{c}) \cap \Upsilon_\alpha(\mathbf{x}) \neq \varnothing \quad \Longrightarrow \quad \mathbf{c} \in g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})), \tag{82}$$

hence $\mathbf{c} \in \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})) = \Gamma^{\mathsf{rev}}(\mathbf{x})$.

Step 2: Any pair satisfying (1) and (2) is contained in $(\Gamma^{\mathsf{rev}}(\mathbf{x}), \Upsilon^{\mathsf{rev}}(\mathbf{x}))$.

Let (Γ', Υ') satisfy (1) and (2). We must show that $\Gamma' \subseteq \Gamma^{\mathsf{rev}}(\mathbf{x})$ and $\Upsilon' \subseteq \Upsilon^{\mathsf{rev}}(\mathbf{x})$.

Show $\Gamma' \subseteq \Gamma^{\mathsf{rev}}(\mathbf{x})$. Let $\mathbf{c} \in \Gamma'$. We prove $\mathbf{c} \in \Gamma^{\mathsf{rev}}(\mathbf{x})$ by checking the two conditions defining $\Gamma^{\mathsf{rev}}(\mathbf{x}) = \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$.

(i) $\mathbf{c} \in \Gamma_\beta(\mathbf{x})$: this follows directly from $\mathbf{c} \in \Gamma' \subseteq \Gamma_\beta(\mathbf{x})$ by (1).

(ii) $\mathbf{c} \in g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$: by definition, this requires $g^{\dagger}(\mathbf{c}) \cap \Upsilon_\alpha(\mathbf{x}) \neq \varnothing$. From (2a), $g^{\dagger}(\mathbf{c}) \cap \Upsilon' \neq \varnothing$, and from (1), $\Upsilon' \subseteq \Upsilon_\alpha(\mathbf{x})$. Hence

$$\varnothing \neq g^{\dagger}(\mathbf{c}) \cap \Upsilon' \subseteq g^{\dagger}(\mathbf{c}) \cap \Upsilon_\alpha(\mathbf{x}). \tag{83}$$

Combining (i) and (ii) gives $\mathbf{c} \in \Gamma_\beta(\mathbf{x}) \cap g^{-\dagger}(\Upsilon_\alpha(\mathbf{x})) = \Gamma^{\mathsf{rev}}(\mathbf{x})$.

Show $\Upsilon' \subseteq \Upsilon^{\mathsf{rev}}(\mathbf{x})$. Let $\mathbf{y} \in \Upsilon'$. We prove $\mathbf{y} \in \Upsilon^{\mathsf{rev}}(\mathbf{x})$ by checking the two conditions defining $\Upsilon^{\mathsf{rev}}(\mathbf{x}) = \Upsilon_\alpha(\mathbf{x}) \cap g^{\dagger}(\Gamma_\beta(\mathbf{x}))$.

(i) $\mathbf{y} \in \Upsilon_\alpha(\mathbf{x})$: this follows directly from $\mathbf{y} \in \Upsilon' \subseteq \Upsilon_\alpha(\mathbf{x})$ by (1).

(ii) $\mathbf{y} \in g^{\dagger}(\Gamma_\beta(\mathbf{x}))$: by definition of the image, this requires some $\mathbf{c} \in \Gamma_\beta(\mathbf{x})$ with $\mathbf{y} \in g^{\dagger}(\mathbf{c})$. From (2b), there exists $\mathbf{c} \in \Gamma'$ with $\mathbf{y} \in g^{\dagger}(\mathbf{c})$, and from (1), $\mathbf{c} \in \Gamma' \subseteq \Gamma_\beta(\mathbf{x})$.

Combining (i) and (ii) gives $\mathbf{y} \in \Upsilon_\alpha(\mathbf{x}) \cap g^{\dagger}(\Gamma_\beta(\mathbf{x})) = \Upsilon^{\mathsf{rev}}(\mathbf{x})$.

This concludes the proof. □

B.9 Coverage proof of Product Set leveraging e-values

Proposition B.9 (Concept coverage via aggregated e-values)**.** *For $0 < \beta < 1$ and any $(\mathbf{x}, \mathbf{c}^*) \in \mathcal{D}_{\mathbf{C}}$, let $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_K) \in \mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_K$ be a concept vector, where each $\mathbf{c}_j \in \{0, \ldots, V-1\}$. For each concept j, let $s_j : \mathcal{X} \times \mathcal{C}_j \to \mathbb{R}_{\geq 0}$ be a non-negative conformity score and let $\mathcal{D}_j = \{(\mathbf{x}_i, \mathbf{c}^*_{ij})\}_{i=1}^n$ be a calibration set. Define the* soft-rank e-value *[109] for concept j and candidate v as*

$$E_j(\mathbf{x}, \mathbf{v}) = \frac{(n+1)\, s_j(\mathbf{x}, v)}{\sum_{i=1}^{n} s_j(\mathbf{x}_i, c^*_{ij}) + s_j(\mathbf{x}, \mathbf{v})}. \tag{84}$$

the joint e-value *for a candidate tuple $\mathbf{v} = (v_1, \ldots, v_K)$ as*

$$E(\mathbf{x}, \mathbf{v}) = \frac{1}{K} \sum_{j=1}^{K} E_j(\mathbf{x}, v_j). \tag{85}$$

and the concept prediction set *for threshold $\beta > 0$ as*

$$\Gamma_\beta(\mathbf{x}) = \left\{ \mathbf{v} \in \bigtimes_{j=1}^{K} \{0, \ldots, V-1\} : E(\mathbf{x}, \mathbf{v}) < \frac{1}{\beta} \right\}. \tag{86}$$

Then:

$$\Pr\left(\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})\right) \geq 1 - \beta. \tag{87}$$

In particular, this guarantee holds without Bonferroni correction (Corollary B.4) and does not degrade with the number of concepts K.

Proof. **Step 1: each $E_j(\mathbf{x}, \mathbf{c}^*_j)$ is a valid e-variable**. By exchangeability of the calibration data and the test point, each soft-rank e-value satisfies $\mathbb{E}[E_j(\mathbf{x}, \mathbf{c}^*_j)] \leq 1$ Vovk [18], Gauthier et al. [19].

Step 2: the average $E(\mathbf{x}, \mathbf{c}) = \frac{1}{K} \sum_{j=1}^{K} E_j(\mathbf{x}, \mathbf{c}_j)$ is a valid e-variable. By linearity of expectation:

$$\mathbb{E}\left[E(\mathbf{x}, \mathbf{c}^*)\right] = \mathbb{E}\left[\frac{1}{K} \sum_{j=1}^{K} E_j(\mathbf{x}, \mathbf{c}^*_j)\right] = \frac{1}{K} \sum_{j=1}^{K} \mathbb{E}\left[E_j(\mathbf{x}, \mathbf{c}^*_j)\right] \leq \frac{1}{K} \cdot K = 1, \tag{88}$$

where the inequality uses Step 1 termwise. Hence $E(\mathbf{x}, \mathbf{c}^*)$ is a valid e-variable.

Step 3: Coverage follows by Markov's inequality [17]. Since $E(\mathbf{x}, \mathbf{c}) \geq 0$ and $\mathbb{E}[E(\mathbf{x}, \mathbf{c})] \leq 1$, Markov's inequality gives:

$$\Pr\left(E(\mathbf{x}, \mathbf{c}) \geq \frac{1}{\beta}\right) \leq \beta, \tag{89}$$

or equivalently,

$$\Pr\left(E(\mathbf{x}, \mathbf{c}) < \frac{1}{\beta}\right) \geq 1 - \beta. \tag{90}$$

Since $\mathbf{c} \in \Gamma_\beta(\mathbf{x})$ if and only if $E(\mathbf{x}, \mathbf{c}) < 1/\beta$, the coverage guarantee follows. □

Remark 4 (Comparison with Bonferroni correction (Corollary B.4)). Proposition 3.1 applies Bonferroni's correction, allocating miscoverage β/K to each individual concept and requiring K separate thresholds. The e-value approach in Proposition B.9 avoids this entirely. Independence is exploited at the level of expectations via the product factorization in Step 2, so a single threshold $1/\beta$ suffices and the guarantee does not degrade with K.

Remark 5 (Average vs. product aggregation). An alternative aggregation is the *product* $E(\mathbf{x}, \mathbf{c}) = \prod_{j=1}^{K} E_j(\mathbf{x}, \mathbf{c}_j)$. To make it a valid e-variable, it requires:

$$\mathbb{E}\left[\prod_{j=1}^{K} E_j(\mathbf{x}, \mathbf{c}^*_j)\right] = \prod_{j=1}^{K} \mathbb{E}\left[E_j(\mathbf{x}, \mathbf{c}^*_j)\right], \tag{91}$$

which is *not* implied by the conditional concept independence $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j \mid \mathbf{X}$ standard in NeSy-CBMs [10, 33], because the per-concept e-values share the same test input $\mathbf{x}$ and the same calibration set $\mathcal{D}_{\text{cal}}$. The product is a valid e-variable if one additionally assumes *conditional* independence of the per-concept e-values given $(\mathbf{x}, \mathcal{D}_{\text{cal}})$, which might not hold in pratice. Despite this, COCOCO* also behaves well when using the product, as shown in our experiments for CHX (see Section E.3).

B.10 Coverage proof of COCOCO leveraging e-values

Proposition B.10 (Joint coverage of COCOCO using e-values)**.** *For $0 < \tilde{\alpha}, \tilde{\beta}$ possibly data-dependent, let $\Gamma^{\mathsf{rev}}(\mathbf{x})$ and $\Upsilon^{\mathsf{rev}}(\mathbf{x})$ be the jointly revised sets of COCOCO (Eq. (10)). Suppose the concept set $\Gamma_\beta(\mathbf{x})$ and the label set $\Upsilon_\alpha(\mathbf{x})$ are constructed using e-values so that*

$$\Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x})) \leq \mathbb{E}[\tilde{\beta}], \qquad \Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x})) \leq \mathbb{E}[\tilde{\alpha}]. \tag{92}$$

If the NeSy-CBM satisfies both deductive and abductive soundness (Definition 3.1), then

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) \geq 1 - \mathbb{E}[\tilde{\alpha}] - \mathbb{E}[\tilde{\beta}] + \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x})), \tag{93}$$

$$\Pr(\mathbf{y}^* \in \Upsilon^{\mathsf{rev}}(\mathbf{x})) \geq 1 - \mathbb{E}[\tilde{\alpha}] - \mathbb{E}[\tilde{\beta}] + \Pr(\mathbf{y}^* \notin \Upsilon_\alpha(\mathbf{x}) \wedge \mathbf{y}^* \notin \Upsilon^{\mathsf{de}}(\mathbf{x})). \tag{94}$$

Proof. We prove the bound for $\Gamma^{\mathsf{rev}}(\mathbf{x})$; the argument for $\Upsilon^{\mathsf{rev}}(\mathbf{x})$ is identical by symmetry.

Let $A := (\mathbf{c}^* \in \Gamma_\beta(\mathbf{x}))$ and $B := (\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}))$, so $\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) = \Pr(A \cap B)$. By inclusion–exclusion,

$$\Pr(A \cap B) = \Pr(A) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B). \tag{95}$$

Under abductive soundness, Proposition B.2 gives $\delta_{\mathrm{ab}} = 1$, so $\mathbf{c}^* \in \Gamma^{\mathsf{ab}}(\mathbf{x}) = g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$ whenever $\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})$. Hence $\Pr(B) \geq 1 - \mathbb{E}[\tilde{\alpha}]$.

Using also $\Pr(A) \geq 1 - \mathbb{E}[\tilde{\beta}]$ and substituting into Eq. (95),

$$\begin{aligned}
\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) &= \Pr(A) + \Pr(B) - 1 + \Pr(\neg A \cap \neg B) && \text{(Eq. (95))} \\
&\geq (1 - \mathbb{E}[\tilde{\beta}]) + (1 - \mathbb{E}[\tilde{\alpha}]) - 1 + \Pr(\neg A \cap \neg B) && \text{(e-value guarantees)} \\
&= 1 - \mathbb{E}[\tilde{\alpha}] - \mathbb{E}[\tilde{\beta}] + \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x})). && \text{(simplify, } \delta_{\mathrm{de}}\text{)}
\end{aligned}$$

This concludes the proof. □

Remark 6 (Relation to fixed miscoverage levels). When $\tilde{\alpha}$ and $\tilde{\beta}$ are fixed constants independent of the data, $\mathbb{E}[\tilde{\alpha}] = \alpha$ and $\mathbb{E}[\tilde{\beta}] = \beta$, and the bound recovers the guarantee of Proposition 4.2 with the additional term $\Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x}))$ [17]. If instead, *abductive* and *deductive* soundness do not hold, the bound can be recovered the same way as Proposition 4.2.

C Implementation Details

Here, we provide additional details about all metrics, datasets, and models useful for reproducibility.

C.1 Implementation

All experiments were conducted using Python 3.9 and PyTorch 2.5.1, and were run on a single NVIDIA A100 GPU. The implementations of DPL and LTN for MN-Add, MN-H, and MN-EO were taken from [94], while they were developed from scratch for all other datasets.

The MN-Add variants were constructed starting from the original MNIST images [7]. The CHX dataset was taken verbatim from [110]. For RIV10, we used the images contained in the ordinary folder, preserving the original train/test split.

For CEB, we used train_observational as the training set, dev as the validation set, and test as the test set. For CIF10, we retained the original train/test split.

C.2 Metrics details

For all datasets, we evaluate predictions at both the label and concept levels using three metrics, *coverage*, *size*, and *consistency*, defined below.

Coverage. Coverage is defined as:

$$\textsc{Coverage} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{test}}} \mathbb{1}\{\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})\}, \tag{96}$$

where $\mathcal{D}_{\text{test}}$ denotes the test set and $|\mathcal{D}_{\text{test}}|$ its cardinality. This metric quantifies the proportion of test samples for which the ground-truth concept $\mathbf{c}^*$ is included in the predicted concept set $\Gamma_\beta(\mathbf{x})$. An analogous definition applies at the label level by replacing $\mathbf{c}^*$ and $\Gamma_\beta(\mathbf{x})$ with $\mathbf{y}^*$ and $\Upsilon_\alpha(\mathbf{x})$, respectively.

Size. We measure the average size of the prediction sets as

$$\textsc{Size} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{test}}} |\Gamma_\beta(\mathbf{x})|, \tag{97}$$

which corresponds to the expected cardinality of the predicted concept sets. The same definition applies to label sets $\Upsilon_\alpha(\mathbf{x})$.

Consistency. Concept-level consistency is defined as

$$\textsc{Consistency} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{test}}} \sum_{\mathbf{c} \in \Gamma_\beta(\mathbf{x})} \frac{\mathbb{1}\{g^\dagger(\mathbf{c}) \cap \Upsilon_\alpha(\mathbf{x}) \neq \varnothing\}}{\max(1, |\Gamma_\beta(\mathbf{x})|)}, \tag{98}$$

this metric measures the extent to which predicted concepts agree with predicted labels. Specifically, it captures the average fraction of concepts in $\Gamma_\beta(\mathbf{x})$ whose induced label $g^\dagger(\mathbf{c})$ is contained in $\Upsilon_\alpha(\mathbf{x})$. A fully consistent model ensures that every predicted concept is supported by at least one compatible label. An analogous definition applies at the label level.

C.3 E-values details

A powerful and recent tool in the conformal prediction literature is conformal e-prediction, which uses *e-values* [14, 15, 16, 17] to build prediction sets [18, 19, 20, 21]. An *e-variable* E is a nonnegative random variable satisfying $\mathbb{E}[E] \leq 1$ [17]. Thresholding E at level $1/\alpha$ yields a valid prediction set with marginal coverage at least $1 - \alpha$. By Markov's inequality, $\Pr(E \geq 1/\alpha) \leq \alpha$, or equivalently, $\Pr(E < 1/\alpha) \geq 1 - \alpha$.

A standard e-value for conformal prediction is the *soft-rank e-variable* [109, 111, 112]. Given a nonnegative nonconformity score $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and a calibration set $\mathcal{D}_{\text{cal}} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^n$, it is defined as:

$$E(\mathbf{x}, \mathbf{y}) = \frac{(n+1)\, s(\mathbf{x}, \mathbf{y})}{\sum_{i=1}^{n} s(\mathbf{x}_i, \mathbf{y}_i^*) + s(\mathbf{x}, \mathbf{y})}. \tag{99}$$

This is a valid e-variable whenever the calibration scores $\{s(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^n$ and the test score $s(\mathbf{x}, \mathbf{y})$ are exchangeable [18].

Proposition C.1 (Gauthier et al. [19])**.** *Consider a calibration set $\mathcal{D}_{\text{cal}} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^n$ and a test point $(\mathbf{x}, \mathbf{y}^*)$ such that all $n+1$ points are exchangeable. Let $\tilde{\alpha}$ be any (possibly data-dependent) miscoverage level. Then:*

$$\mathbb{E}\left[\frac{\Pr(\mathbf{y}^* \notin \Upsilon_{\tilde{\alpha}}(\mathbf{x}) \mid \tilde{\alpha})}{\tilde{\alpha}}\right] \leq 1, \tag{100}$$

where

$$\Upsilon_{\tilde{\alpha}}(\mathbf{x}) = \left\{ y \in \mathcal{Y} : \frac{(n+1)\, s(\mathbf{x}, y)}{\sum_{i=1}^{n} s(\mathbf{x}_i, \mathbf{y}_i^*) + s(\mathbf{x}, y)} < \frac{1}{\tilde{\alpha}} \right\}. \tag{101}$$

Remark 7. When $\tilde{\alpha}$ is a fixed constant independent of the data, Eq. (100) reduces to the standard conformal guarantee $\Pr(\mathbf{y}^* \in \Upsilon_\alpha(\mathbf{x})) \geq 1 - \alpha$ [19]

E-value prediction sets. COCOCO* (Section 4) instantiates Eq. (99) with the negative log-probability nonconformity score $s(\mathbf{x}, y) = -\log p_\theta(y \mid \mathbf{x})$, which is non-negative by design.

Label sets. At calibration time, COCOCO* computes the label score sum $S_Y = \sum_{i=1}^n s(\mathbf{x}_i, \mathbf{y}_i^*)$ and stores the calibration count n. At test time, given a miscoverage level $\tilde{\alpha}$, the label prediction set, following Eq. (101), is

$$\Upsilon_\alpha(\mathbf{x}) = \left\{ y \in \mathcal{Y} : \frac{(n+1)\, s(\mathbf{x}, y)}{S_Y + s(\mathbf{x}, y)} < \frac{1}{\tilde{\alpha}} \right\}. \tag{102}$$

Concept sets. COCOCO* aggregates per-concept e-values via averaging:

$$E(\mathbf{x}, \mathbf{c}) = \frac{1}{K} \sum_{j=1}^{K} E_j(\mathbf{x}, v_j), \tag{103}$$

which is a valid e-variable by linearity of expectation (Proposition B.9). The concept prediction set

$$\Gamma_\beta(\mathbf{x}) = \left\{ \mathbf{c} \in \mathcal{C} : E(\mathbf{x}, \mathbf{c}) < \tfrac{1}{\tilde{\beta}} \right\} \tag{104}$$

satisfies $\Pr(\mathbf{c}^* \in \Gamma_\beta(\mathbf{x})) \geq 1 - \mathbb{E}[\tilde{\beta}]$ (Proposition B.9), *without* Bonferroni correction (Corollary B.4) and *without* degradation in K. The product alternative $\prod_j E_j$ is also valid but requires conditional independence of the E_j given $(\mathbf{x}, \mathcal{D}_{\text{cal}})$, which does not hold in practice (see Remark 5). Nevertheless, COCOCO* remains robust to this violation (see Section E.3).

Size-budget calibration: computing $\tilde{\alpha}$ and $\tilde{\beta}$.

Rather than fixing α and β a priori, COCOCO* selects the *smallest miscoverage* levels that keep the average prediction-set sizes within user-specified budgets C_Y (for labels) and C_C (for concepts).

Formally, following Gauthier et al. [20], given a test set $\mathcal{D}$ and calibration statistics $(S_Y, \{S_j\}_{j=1}^K)$:

$$\tilde{\alpha} = \inf \left\{ \alpha \in (0,1) : \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} |\Upsilon_\alpha(\mathbf{x})| \leq C_Y \right\}, \tag{105}$$

$$\tilde{\beta} = \inf \left\{ \beta \in (0,1) : \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} |\Gamma_\beta(\mathbf{x})| \leq C_C \right\}. \tag{106}$$

Both $\tilde{\alpha}$ and $\tilde{\beta}$ are data-dependent. They depend on the calibration score sums S_Y and $\{S_j\}$. According to Proposition B.10, under deductive and abductive soundness (Definition 3.1), the jointly revised sets $\Gamma^{\mathsf{rev}}(\mathbf{x})$ and $\Upsilon^{\mathsf{rev}}(\mathbf{x})$ produced by COCOCO* satisfy:

$$\Pr(\mathbf{c}^* \in \Gamma^{\mathsf{rev}}(\mathbf{x})) \geq 1 - \mathbb{E}[\tilde{\alpha}] - \mathbb{E}[\tilde{\beta}] + \Pr(\mathbf{c}^* \notin \Gamma_\beta(\mathbf{x}) \wedge \mathbf{c}^* \notin \Gamma^{\mathsf{ab}}(\mathbf{x})). \tag{107}$$

Bootstrap estimation of $\mathbb{E}[\tilde{\alpha}]$ and $\mathbb{E}[\tilde{\beta}]$. We estimate these quantities over $T = 100$ bootstrap iterations. In each iteration t, a calibration sample $\mathcal{D}_t$ of size $|\mathcal{D}_{\text{cal}}|$ is drawn *with replacement* from $\mathcal{D}_{\text{cal}}$, score sums are recomputed on $\mathcal{D}_t$, and prediction sets are built on the fixed test set for all (α, β) in $(\{0.10, 0.11, \ldots, 0.30\}, \{0.10, 0.30, 0.31, 0.60\})$. The minimal miscoverage levels $\tilde{\alpha}_t$ and $\tilde{\beta}_t$ are then selected as the smallest joint values achieving average set sizes at most $C_Y = 2$ and $C_C = 5$, respectively (Eq. (106)). The empirical coverage lower bounds are reported as

$$1 - \frac{1}{T} \sum_{t=1}^{T} \tilde{\alpha}_t \qquad \text{and} \qquad 1 - \frac{1}{T} \sum_{t=1}^{T} \tilde{\beta}_t. \tag{108}$$

Budgets hold on average. Equations (105) and (106) constrain the *average* set size, not the size of each individual set, and the resulting coverage guarantee also holds in expectation (Proposition B.10). As in standard CP, atypical inputs with unusually high nonconformity scores can therefore still produce sets exceeding the budget [41].

C.4 Neuro-Symbolic Models

In this subsection, we report relevant details about the implementation and use of the neuro-symbolic models used in the experiments, i.e., DPL and LTN.

C.4.1 Inference

We begin by describing how inference is carried out at test time in DPL and LTN, that is, how the concept distribution $p_\theta(\mathbf{C} \mid \mathbf{x})$ produced by the neural backbone is combined with prior knowledge K to yield a label prediction $\hat{\mathbf{y}}$.

DeepProbLog. We use the DPL implementation of Bortolotti et al. [94]. At inference time, given an input $\mathbf{x}$, the model computes the joint distribution over concepts and labels under prior knowledge K

$$p_\theta(\mathbf{Y}, \mathbf{C} \mid \mathbf{x}, \mathsf{K}) = p_\theta(\mathbf{C} \mid \mathbf{X}) \mathbb{1}\{(\mathbf{C}, \mathbf{Y}) \models \mathsf{K}\}. \tag{109}$$

Hard predictions are obtained by solving the MAP problem, which is NP-hard [113], that is:

$$(\hat{\mathbf{c}}, \hat{\mathbf{y}}) = \underset{\mathbf{c} \in \mathcal{C}, \mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \, p_\theta(\mathbf{C} \mid \mathbf{x}) \mathbb{1}\{(\mathbf{C}, \mathbf{Y}) \models \mathsf{K}\} \tag{110}$$

where

$$p_\theta(\mathbf{C} \mid \mathbf{x}) = \prod_j^K p_\theta(\mathbf{C}_j \mid \mathbf{x}) \tag{111}$$

Namely, $p_\theta(\mathbf{C} \mid \mathbf{x})$ factorizes over all the K concepts due to conditional independence [8].

Logic Tensor Networks. To implement LTN, we used the `ltn-torch` library [114], following the approach of Bortolotti et al. [94]. Given a formula K and concept distribution $p_\theta(\mathbf{C} \mid \mathbf{x})$, each candidate label $\mathbf{y}$ is scored according to the degree to which the pair $(\mathbf{c}, \mathbf{y})$ satisfies K under the chosen fuzzy relaxation.

We experiment with three t-(co)norms:

- Gödel semantics

$$\mathcal{T}_G(a, b) = \min(a, b), \quad \mathcal{S}_G(a, b) = \max(a, b), \quad \mathcal{N}_G(a) = \mathbb{1}\{a = 0\} \tag{112}$$

- Product semantics

$$\mathcal{T}_P(a, b) = a \cdot b, \quad \mathcal{S}_P(a, b) = a + b - a \cdot b, \quad \mathcal{N}_P(a) = 1 - a \tag{113}$$

- Łukasiewicz semantics

$$\mathcal{T}_L(a, b) = \max(a + b - 1, 0), \quad \mathcal{S}_L(a, b) = \min(a + b, 1), \quad \mathcal{N}_L(a) = 1 - a \tag{114}$$

Where $\mathcal{T}$ denotes a *t-norm*, $\mathcal{S}$ denotes a *t-conorm*, $\mathcal{N}$ denotes the *negation operation*, and a and b are fuzzy truth values in $[0, 1]$.

The inference layer selects the label:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \, \mathcal{T}_\mathsf{K}\left(p_\theta(\mathbf{C} \mid \mathbf{x}), \mathbb{1}\{\mathbf{Y} = \mathbf{y}\}\right) \tag{115}$$

where $\mathcal{T}_\mathcal{K} \in [0, 1]$ denotes the fuzzy satisfaction degree.

In practice, inference is performed by first computing:

$$\hat{\mathbf{c}} = \underset{\mathbf{c} \in \mathcal{C}}{\operatorname{argmax}} \, p_\theta(\mathbf{C} \mid \mathbf{x}) \tag{116}$$

and then predicting the final label leveraging prior knowledge K:

$$\hat{\mathbf{y}} = \mathsf{K}(\hat{\mathbf{c}}) \tag{117}$$

C.4.2 Grounding

For each dataset, we detail how the prior knowledge K is grounded in DPL and LTN. In DPL, K is compiled into a *concept-to-label* matrix $W \in [0, 1]^{|\mathcal{C}| \times |\mathcal{Y}|}$, where each entry $W_{\mathbf{c},\mathbf{y}}$ represents the probability of $(\mathbf{c}, \mathbf{y}) \models \mathsf{K}$. The label distribution is then computed by $p_\theta(\mathbf{Y} \mid \mathbf{x}) = W^\top p_\theta(\mathbf{C} \mid \mathbf{x})$. In LTN, K is expressed as a first-order logic formula whose predicates are grounded on the neural outputs $p_\theta(\mathbf{c} \mid \mathbf{x}) \in [0, 1]$, while logical connectives are interpreted using a chosen t-norm (and corresponding co-tnorm), as detailed in Section D.2.

MN-Add (and MN-EO, MN-H). Each input consists of two images $(\mathbf{x}_1, \mathbf{x}_2)$ with associated digit variables $\mathbf{c}_1, \mathbf{c}_2 \in \{0, \ldots, 9\}$ and output $\mathbf{y} \in \{0, \ldots, 18\}$. DPL: The circuit enumerates all 10^2 digit combinations and sets $W_{(\mathbf{c}_1, \mathbf{c}_2), \mathbf{y}} = \mathbb{1}\{\mathbf{c}_1 + \mathbf{c}_2 = \mathbf{y}\}$, so that each concept vector $\mathbf{c}$ maps deterministically to exactly one sum. LTN: The knowledge base is

$$\mathsf{K}_{\texttt{MN-Add}} := \forall \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}. \exists d_1, d_2. \big(\text{digit}(\mathbf{x}_1, d_1) \wedge \text{digit}(\mathbf{x}_2, d_2)\big) \text{ s.t. } d_1 + d_2 = \mathbf{y}, \tag{118}$$

where $\mathrm{digit}(\mathbf{x}_i, d_i) := p_\theta(\mathbf{c}_i = d \mid \mathbf{x}_i)$ is the digit probability.

`MN-SP`. The setup is as in `MN-Add`, but with $\mathbf{Y} \in \{0, 1\}$ encoding the parity of the sum. DPL: $W_{(\mathbf{c}_1,\mathbf{c}_2),\mathbf{y}} = \mathbb{1}\{(\mathbf{c}_1 + \mathbf{c}_2) \bmod 2 = \mathbf{y}\}$. LTN: Same FOL template as `MN-Add`, with arithmetic condition $d_1 + d_2 = \mathbf{y} \pmod 2$.

`CHX`. The concepts are the four binary symptoms $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4) \in 0, 1^4$ (`Fracture`, `Pneumothorax`, `Airspace opacity`, `Nodule/mass`), and the label $\mathbf{y}$ denotes the severity level. DPL: The model enumerates all 2^4 possible concept combinations and defines $W_{\mathbf{c},\mathbf{y}} = \mathbb{1}\{\sum_{i=1}^4 \mathbf{c}_i = \mathbf{y}\}$, corresponding to the five severity levels `healthy`, `green`, `yellow`, `orange`, `red`. LTN: We instead formulate the knowledge base in first-order logic. Let $\mathrm{present}(\mathbf{x}, i) := p_\theta(\mathbf{c}_i = 1 \mid \mathbf{x})$. The knowledge base is then defined as follows:

$$\mathsf{K}^{\mathrm{mc}}_{\texttt{CHX}} := \forall \mathbf{x}, \mathbf{y}.\, \exists s_1, s_2, s_3, s_4 \in \{0, 1\}.\, \bigwedge_{i=1}^{4} \big(s_i \iff \mathrm{present}(\mathbf{x}, i)\big) \text{ s.t. } \textstyle\sum_{i=1}^4 s_i = \mathbf{y}. \tag{119}$$

`CEB`. Recall that $\mathbf{c} \in$ `Positive`, `Negative`, `Neutral`, `unknown`^{5} and the label $\mathbf{y} \in$ `Positive`, `Negative`, `Neutral`, `unknown`, `conflict` is defined as the majority vote with the tie-breaking rule described in Section C.5. DPL: The circuit enumerates all 4^5 possible concept combinations. For each combination, let $N_v := \sum_{i=1}^5 \mathbb{1}\{\mathbf{c}_i = v\}$ denote the count of each value $v \in$ `Positive`, `Negative`, `Neutral`, `unknown`. Then

$$W_{\mathbf{c},\,y} = \begin{cases} \mathbb{1}\{N_{\texttt{Pos}} = N_{\texttt{Neg}} \wedge N_{\texttt{Pos}} \geq \max(N_{\texttt{Neu}}, N_{\texttt{Unk}})\} & \mathbf{y} = \texttt{Conflict}, \\ \mathbb{1}\{N_v \geq \max_{v' \neq v} N_{v'}\} \cdot \big(1 - W_{\mathbf{c},\texttt{Conflict}}\big) & \mathbf{y} \in \{\texttt{Pos}, \texttt{Neg}, \texttt{Neu}, \texttt{Unk}\}, \end{cases} \tag{120}$$

where the tie-breaking priority `Pos` $\succ$ `Neg` $\succ$ `Neu` $\succ$ `Unk` is enforced by evaluating the cases in order and assigning exactly one label per concept combination. LTN: The discrete argmax rule is not directly differentiable. We therefore adopt a relaxed formulation based on *soft counts* $\tilde{N}v(\mathbf{x}) := \sum i = 1^5 p_\theta(\mathbf{C}_i = v \mid \mathbf{x})$, together with smooth comparison operators:

$$\mathrm{ge}(a, b) := \sigma(a - b), \qquad \mathrm{eq}(a, b) := \exp\big(-|a - b|\big), \tag{121}$$

where σ denotes the sigmoid function, i.e., $\sigma(x) = \frac{1}{1+\exp(-x)}$. Strict inequality is expressed compositionally as $\mathrm{gt}(a, b) := \mathrm{ge}(a, b) \wedge \neg, \mathrm{eq}(a, b)$. The knowledge base is defined as the conjunction, over all five labels, of biconditionals between each label indicator and the corresponding count-based predicate:

$$\begin{aligned} \mathsf{K}_{\texttt{CEB}} := \forall \mathbf{x}, \mathbf{y}.\big(&\mathbb{1}\{\mathbf{y} = \texttt{Pos}\} \iff \mathrm{gt}(\tilde{N}_{\texttt{Pos}}, \tilde{N}_{\texttt{Neg}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Pos}}, \tilde{N}_{\texttt{Neu}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Pos}}, \tilde{N}_{\texttt{Unk}})\big) \\ \wedge\big(&\mathbb{1}\{\mathbf{y} = \texttt{Neg}\} \iff \mathrm{gt}(\tilde{N}_{\texttt{Neg}}, \tilde{N}_{\texttt{Pos}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Neg}}, \tilde{N}_{\texttt{Neu}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Neg}}, \tilde{N}_{\texttt{Unk}})\big) \\ \wedge\big(&\mathbb{1}\{\mathbf{y} = \texttt{Neu}\} \iff \mathrm{gt}(\tilde{N}_{\texttt{Neu}}, \tilde{N}_{\texttt{Pos}}) \wedge \mathrm{gt}(\tilde{N}_{\texttt{Neu}}, \tilde{N}_{\texttt{Neg}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Neu}}, \tilde{N}_{\texttt{Unk}})\big) \\ \wedge\big(&\mathbb{1}\{\mathbf{y} = \texttt{Unk}\} \iff \mathrm{gt}(\tilde{N}_{\texttt{Unk}}, \tilde{N}_{\texttt{Pos}}) \wedge \mathrm{gt}(\tilde{N}_{\texttt{Unk}}, \tilde{N}_{\texttt{Neg}}) \wedge \mathrm{gt}(\tilde{N}_{\texttt{Unk}}, \tilde{N}_{\texttt{Neu}})\big) \\ \wedge\big(&\mathbb{1}\{\mathbf{y} = \texttt{Conflict}\} \iff \mathrm{eq}(\tilde{N}_{\texttt{Pos}}, \tilde{N}_{\texttt{Neg}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Pos}}, \tilde{N}_{\texttt{Neu}}) \wedge \mathrm{ge}(\tilde{N}_{\texttt{Pos}}, \tilde{N}_{\texttt{Unk}})\big). \end{aligned}$$

The tie-breaking priority `Pos` $\succ$ `Neg` $\succ$ `Neu` $\succ$ `Unk` is encoded through the asymmetric use of ge and gt: higher-priority categories are allowed to win ties via $\geq$, whereas lower-priority categories must satisfy the strict inequality $>$.

`CIF10` **and** `RIV10`. The concepts are the 7 binary visual attributes $\mathbf{c} = (\mathbf{c}_{\texttt{whl}}, \mathbf{c}_{\texttt{met}}, \mathbf{c}_{\texttt{wng}}, \mathbf{c}_{\texttt{ani}}, \mathbf{c}_{\texttt{hai}}, \mathbf{c}_{\texttt{hrn}}, \mathbf{c}_{\texttt{snt}}) \in 0, 1^7$, and the labels are the 10 `CIF10` classes. DPL: The circuit enumerates all 2^7 concept combinations. For each combination, the rule base in Section C.5 is evaluated; when the active concepts uniquely entail a single class, we set $W_{\mathbf{c},\mathbf{y}} = 1$ for that class. When two classes share the same concepts (i.e., `automobile/truck` and `dog/horse`), the mass is split equally, yielding $W_{\mathbf{c},\mathbf{y}} = 1/2$ for each. Concept combinations that do not uniquely identify any class are handled by distributing probability uniformly over the consistent classes (e.g., $W_{\mathbf{c},\texttt{plane}} = W_{\mathbf{c},\texttt{car}} = W_{\mathbf{c},\texttt{truck}} = W_{\mathbf{c},\texttt{ship}} = 1/4$ when only `met` is active). This explicitly encodes the *incompleteness* of K in the truth table. LTN: For `LTN`, let $\mathrm{present}(\mathbf{x}, a) = p_\theta(\mathbf{c}_a = 1 \mid \mathbf{x})$ for $a \in$ `whl`, `met`, `wng`, `ani`, `hai`, `hrn`, `snt`. The knowledge base is expressed as a conjunction of

biconditionals, one per class:

$$\begin{aligned}\mathsf{K}_{\texttt{CIF10}} := \forall \mathbf{x}, \mathbf{y}. \big(&\mathbb{1}\{\mathbf{y} = \texttt{plane}\} \iff \text{met} \wedge \neg\,\text{ani} \wedge \text{wng}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{car}\} \vee \mathbb{1}\{\mathbf{y} = \texttt{truck}\} \iff \text{whl} \wedge \neg\,\text{wng} \wedge \text{met} \wedge \neg\,\text{ani}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{bird}\} \iff \text{ani} \wedge \neg\,\text{met} \wedge \text{wng}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{frog}\} \iff \text{ani} \wedge \neg\,\text{met} \wedge \neg\,\text{hai}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{deer}\} \iff \text{ani} \wedge \neg\,\text{met} \wedge \neg\,\text{wng} \wedge \text{hai} \wedge \text{hrn}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{cat}\} \iff \text{ani} \wedge \neg\,\text{met} \wedge \neg\,\text{wng} \wedge \text{hai} \wedge \neg\,\text{hrn} \wedge \neg\,\text{snt}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{dog}\} \vee \mathbb{1}\{\mathbf{y} = \texttt{horse}\} \iff \text{ani} \wedge \neg\,\text{met} \wedge \neg\,\text{wng} \wedge \text{hai} \wedge \neg\,\text{hrn} \wedge \text{snt}\big) \\ \wedge \big(&\mathbb{1}\{\mathbf{y} = \texttt{ship}\} \iff \text{met} \wedge \neg\,\text{ani} \wedge \neg\,\text{whl} \wedge \neg\,\text{wng}\big),\end{aligned}$$

RIV10 uses the same grounding since the two datasets share the same labels and concepts.

C.5 Dataset details

Here we report additional information about the evaluated datasets.

C.5.1 MN-Add

MN-Add, originally introduced in Manhaeve et al. [8], is the canonical benchmark in Neuro-Symbolic AI. Each input consists of two MNIST [7] digit images, and the target label is their arithmetic sum. Formally, each sample is a pair of images $(\mathbf{x}_1, \mathbf{x}_2)$ representing MNIST digits $(\mathbf{c}_1, \mathbf{c}_2)$, each taking values in $[0, 9]$, and the task is to predict:

$$\mathbf{y} = \mathbf{c}_1 + \mathbf{c}_2, \tag{122}$$

In contrast to MN-EO and MN-H, MN-Add contains all possible digit combinations ($54,000$ train examples, $6,000$ validation examples and $10,000$ test examples) and does not induce structural ambiguity in the digit-to-value mapping. The underlying combinatorial structure uniquely identifies each digit, and no deterministic reasoning shortcuts arise. Consequently, MN-Add serves as a shortcut-free reference benchmark for reasoning [23].

Reasoning Shortcuts. MN-Add is provably reasoning shortcuts free. For further information refer to Bortolotti et al. [94].

C.5.2 CHX

CHX [91] is medical dataset characterized by four binary symptoms: Fracture, Pneumothorax, Airspace opacity, and Nodule/mass. Each symptom is represented as a binary variable indicating its presence or absence. For preprocessing, we follow Pugnana et al. [110]. However, in its original form, CHX is not directly suitable for learning and reasoning tasks. For this reason, we construct our setting as follows: let the concept vector be defined as:

$$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4) \in \{0, 1\}^4, \tag{123}$$

where:

$$\mathbf{c} = \begin{pmatrix} \texttt{Fracture} \\ \texttt{Pneumothorax} \\ \texttt{Airspace opacity} \\ \texttt{Nodule/mass} \end{pmatrix} \tag{124}$$

We designed our NeSy task, such that the final label is determined by the total number of present symptoms. Formally, let

$$\mathbf{s} = \sum_{i=1}^{4} \mathbf{c}_i. \tag{125}$$

The diagnostic label $\mathbf{y}$ is defined as:

$$\mathbf{y} = \begin{cases} \texttt{healthy} & \text{if } \mathbf{s} = 0, \\ \texttt{green_code} & \text{if } \mathbf{s} = 1, \\ \texttt{yellow_code} & \text{if } \mathbf{s} = 2, \\ \texttt{orange_code} & \text{if } \mathbf{s} = 3, \\ \texttt{red_code} & \text{if } \mathbf{s} = 4. \end{cases} \tag{126}$$

Thus, the task consists of predicting a severity level derived deterministically from the number of active symptoms. CHX consists of $3,150$ train samples, 350 validation samples, and 876 test samples.

Reasoning Shortcuts. CHX contains reasoning shortcuts. For example, `yellow_code` ($\mathbf{y} = 2$) can be obtained not only by predicting $\mathbf{c} = (1, 0, 1, 0)$, but also via other permutations such as $\mathbf{c}' = (0, 1, 1, 0)$. For the exact (or approximate) number of reasoning shortcuts, we refer the reader to `rs-count` [94] specifying such settings.

C.5.3 CEB

CEB [92] is a dataset designed to evaluate concept-based and causal reasoning in natural language models. It consists of restaurant reviews with structured, aspect-level annotations and counterfactual edits. Each instance consists of:

- A review text called `description`, i.e., the input,
- A set of high-level semantic concepts $\mathbf{c} \in [0, 4]$,
- An overall rating `review_majority` $\in [0, 5]$.

The core concepts correspond to distinct aspects of the restaurant experience:

$$\mathbf{c} = \begin{pmatrix} \texttt{food_aspect_majority} \\ \texttt{service_aspect_majority} \\ \texttt{noise_aspect_majority} \\ \texttt{ambiance_aspect_majority} \end{pmatrix} \tag{127}$$

Each aspect captures the sentiment expressed toward that dimension (i.e., `Positive`, `Negative`, `Neutral`, `unknown`). From this concepts, the goal is to infer the causally influenced label that is `review_majority`. The dataset is distributed in JSON format. Below is an example instance:


```
{
  "id": "000000_000000",
  "original_id": "000000",
  "edit_id": "000000",
  "is_original": true,
  "edit_goal": "Negative",
  "edit_type": "noise",
  "edit_worker": "w82",
  "description": "Overbooked and did not honor reservation time,
  put on wait list with walk-ins",
  "review_majority": "1",
  "review_label_distribution": {
    "1": 4,
    "2": 1
  },
  "review_workers": {
    "w244": "1",
    "w120": "2",
    "w197": "1",
    "w7": "1",
    "w132": "1"
  },
  "food_aspect_majority": "",
  "ambiance_aspect_majority": "",
  "service_aspect_majority": "Negative",
  "noise_aspect_majority": "unknown",
  "service_aspect_label_distribution": {
    "Negative": 5
  },
  "noise_aspect_label_distribution": {
    "unknown": 4,
    "Negative": 1
```

```
  },
  "opentable_metadata": {
    "restaurant_id": 6513,
    "restaurant_name": "Molino's Ristorante",
    "cuisine": "italian",
    "price_tier": "low",
    "dining_style": "Casual Elegant",
    "dress_code": "Smart Casual",
    "parking": "Private Lot",
    "region": "south",
    "rating_ambiance": 1,
    "rating_food": 3,
    "rating_noise": 2,
    "rating_service": 2,
    "rating_overall": 2
  }
}
```


Preprocessing. As originally defined, `CEB` does not provide a deterministic rule mapping aspect-level concepts to the overall review label, making it unsuitable for a NeSy setting in which symbolic inference must be explicit. To address this, we introduce `review_majority` as an additional concept, so that the concept vector $\mathbf{c}$ becomes

$$\mathbf{c} = \begin{pmatrix} \texttt{food_aspect_majority} \\ \texttt{service_aspect_majority} \\ \texttt{noise_aspect_majority} \\ \texttt{ambiance_aspect_majority} \\ \texttt{review_majority} \end{pmatrix} \in \{0, 1, 2, 3\}^5 \tag{128}$$

with each component $\mathbf{c}_i \in \{\texttt{Positive}, \texttt{Neutral}, \texttt{Negative}, \texttt{unknown}\}$.

To make it compliant with the other concepts, the original `review_majority` field is quantized as:

$$\texttt{review_majority} = \begin{cases} \texttt{unknown} & \text{if } \in \{\texttt{no majority}, \texttt{unknown}\}, \\ \texttt{Negative} & \text{if } \leq 2, \\ \texttt{Neutral} & \text{if } = 3, \\ \texttt{Positive} & \text{if } \geq 4, \end{cases} \tag{129}$$

We then construct our reasoning task as selecting the category with the maximum count, representing an aggregate of the restaurant quality:

$$\mathbf{y} = \underset{\mathbf{c}_i}{\operatorname{argmax}} \sum_{\mathbf{c}_j \in C} \mathbb{1}\{\mathbf{c}_i = \mathbf{c}_j\}, \quad \mathbf{y} \in \{\texttt{Positive}, \texttt{Neutral}, \texttt{Negative}, \texttt{unknown}, \texttt{conflict}\}, \tag{130}$$

with the following tie-breaking rule: if both `Positive` and `Negative` attain the maximum count, then $y = \texttt{conflict}$; otherwise, ties follow the priority order $\texttt{Positive} \succ \texttt{Negative} \succ \texttt{Neutral} \succ \texttt{unknown}$. The dataset provides three different training sets: `observational`, `inclusive`, and `exclusive`. Since we do not analyze counterfactuals or interpretability, we use the `observational` training split. The dataset contains $1,755$ training samples, $1,673$ validation samples, and $1,689$ test samples.

Reasoning Shortcuts. `CEB`, as designed, is not free of reasoning shortcuts. For instance, for a given label such as `Negative`, multiple concept combinations are possible, such as $\mathbf{c} = (\texttt{Negative}, \texttt{Negative}, \texttt{Negative}, \texttt{Negative})$ or $(\texttt{Positive}, \texttt{Negative}, \texttt{Neutral}, \texttt{Negative})$. If interested, exact (or approximate) number of reasoning shortcuts can be found by using `rs-count` [94] instatiating the formalized problem.

C.5.4 `CIF10`

The `CIF10` dataset [115] is a standard benchmark for image classification. It contains $60,000$ color images of size 32×32, divided into 10 classes, with $50,000$ training images (which we further split

into $35,000$ for training and $15,000$ for validation) and $10,000$ test images. The original class labels are:

$$\texttt{\{airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck\}} \tag{131}$$

Preprocessing. As with CEB, the raw CIF10 dataset is not directly suitable for learning and reasoning tasks. For this reason, we introduce a set of intermediate concepts that are *intended* to be present in all images and useful for distinguishing the 10 classes. However, to avoid trivial label leakage and to ensure that some concepts remain insufficient, we *deliberately* design the concept-to-label mapping to be incomplete. In particular, we construct the labeling such that, given only the provided concepts, it is not possible to distinguish between automobile and truck, nor between dog and horse. In practice, we build a binary concept vector $\mathbf{c} \in \{0,1\}^7$ for each CIF10 image, encoding high-level semantic attributes such as ani (animal), wng (wing), met (metallic), whl (wheels), hai (hairy), hrn (horn) and snt (snout). These concepts are deterministically derived from the original class label. Then, given the concepts, we build the following rules for the CIF10 labels:

$$\begin{aligned}
\texttt{airplane} &\iff \texttt{met} \wedge \neg\texttt{ani} \wedge \texttt{wng},\\
\texttt{automobile, truck} &\iff \texttt{whl} \wedge \texttt{met} \wedge \neg\texttt{wng} \wedge \neg\texttt{ani},\\
\texttt{bird} &\iff \texttt{ani} \wedge \neg\texttt{met} \wedge \texttt{wng},\\
\texttt{frog} &\iff \texttt{ani} \wedge \neg\texttt{met} \wedge \neg\texttt{hai},\\
\texttt{deer} &\iff \texttt{ani} \wedge \neg\texttt{met} \wedge \neg\texttt{wng} \wedge \texttt{hai} \wedge \texttt{hrn},\\
\texttt{dog, horse} &\iff \texttt{ani} \wedge \neg\texttt{met} \wedge \neg\texttt{wng} \wedge \texttt{hai} \wedge \neg\texttt{hrn} \wedge \texttt{snt},\\
\texttt{ship} &\iff \texttt{met} \wedge \neg\texttt{ani} \wedge \neg\texttt{whl} \wedge \neg\texttt{wng}.
\end{aligned} \tag{132}$$

Reasoning Shortcuts. CIF10, as constructed, is not free of reasoning shortcuts. For instance, airplane is defined as $\texttt{met} \wedge \neg\texttt{ani} \wedge \texttt{wng}$ but nothing is said about other concepts. For example, concept assignments such as $\texttt{met} = \top, \texttt{ani} = \bot, \texttt{wng} = \top, \texttt{hai} = \bot$ and $\texttt{met} = \top, \texttt{ani} = \bot, \texttt{wng} = \top, \texttt{hai} = \top$ both entail the airplane class. For the exact (or approximate) number of reasoning shortcuts, you can instantiate this problem in rs-count [94].

C.5.5 RIV10

RIV10 [93] is a real-world image dataset built from ImageNet [116] that contains natural images spanning the same 10 semantic categories as CIF10: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. We kept the original train/test split as in [93], but creating a validation set from the train data counting $14,768$ train samples, $6,330$ validation samples and $5,286$ test samples.

Preprocessing. Since RIV10 shares the same labels as CIF10, we adopt the exact same preprocessing pipeline.

Reasoning Shortcuts. Since RIV10 shares the same concepts, knowledge and concept combinations of CIF10 (despite having different distributions). Refer to CIF10 for the reasoning shortcuts discussion.

C.5.6 MN-EO

Following [94], we consider the MN-EO dataset, originally introduced in [117]. This dataset is a structured variant of MN-Add in which only a restricted subset of digit combinations is observed during training. Specifically, the dataset contains exclusively pairs composed either of even digits or of odd digits, defined by the following constraints:

$$\begin{cases} 0+6 = 6 \\ 2+8 = 10 \\ 4+6 = 10 \\ 4+8 = 12 \end{cases} \wedge \begin{cases} 1+5 = 6 \\ 3+7 = 10 \\ 1+9 = 10 \\ 3+9 = 12 \end{cases} \tag{133}$$

The dataset contains $6,720$ fully annotated training samples, $1,920$ validation samples, and 960 in-distribution test samples.

Reasoning Shortcuts. MN-EO is one of the first datasets introduced to study reasoning shortcuts; for a detailed analysis, see Marconato et al. [24].

C.5.7 MN-H

MN-H, originally introduced in [9], is a biased variant of MN-Add in which only digits from 0 to 4 are considered. Moreover, only a subset of digit combinations is observed during training, given by:

$$\begin{cases} \boxed{0}+\boxed{0} & =0 \\ \boxed{0}+\boxed{1} & =1 \\ \boxed{2}+\boxed{3} & =5 \\ \boxed{2}+\boxed{4} & =6 \end{cases} \tag{134}$$

This restricted supervision induces ambiguity in the digit-to-value mapping and therefore introduces reasoning shortcuts. In contrast to MN-EO, digits 0 and 1 remain uniquely identifiable from the observed equations, while digits 2, 3, and 4 admit multiple consistent assignments.

The dataset comprises $2,940$ fully annotated training samples, 840 validation samples, and 420 in-distribution test samples.

Reasoning Shortcuts. Similarly to MN-EO, MN-H, was introduced to study reasoning shortcuts; for a detailed analysis, see Marconato et al. [9].

C.5.8 MN-SP

MN-SP, originally introduced in [23], is a variant of MN-Add that has a different logical rule. As in MN-Add, the input consists of two MNIST digit images corresponding to digits $(\mathbf{x}_1, \mathbf{x}_2)$. However, instead of predicting their arithmetic sum, the target label is defined as the *parity* of the sum. For this reason, it has the same amount of samples as MN-Add per split.

Formally, the task is:

$$\mathbf{y} = (\mathbf{x}_1 + \mathbf{x}_2) \bmod 2, \tag{135}$$

where $\mathbf{y} \in \{0, 1\}$, with 0 denoting an even sum and 1 an odd sum. For example:

$$(\boxed{2}+\boxed{4}) \mod 2 = 0 \tag{136}$$
$$(\boxed{1}+\boxed{2}) \mod 2 = 1 \tag{137}$$

Reasoning Shortcuts. Similarly to MN-EO and MN-H, MN-SP was introduced to study shortcuts; for a detailed analysis, see Bortolotti et al. [23].

C.5.9 BDD-OIA

BDD-OIA [118] is a real-world autonomous-driving dataset consisting of frames extracted from driving-scene videos. Each frame is labeled with four binary actions, $\mathbf{y} = (\texttt{move_forward}, \texttt{stop}, \texttt{turn_left}, \texttt{turn_right}) \in \{0,1\}^4$, and annotated with 21 binary concepts $\mathbf{c} \in \{0,1\}^{21}$ that explain the chosen actions, such as the presence of green or red lights, obstacles (cars, pedestrians, riders), and lane markings. The training set contains about 16k fully annotated frames, while the validation and test sets contain about 2k and 4.5k frames, respectively [94].

Preprocessing. Following Marconato et al. [24, 9], we do not process raw images. Each frame is first processed by a Faster R-CNN [119] pre-trained on MS-COCO and fine-tuned on BDD-100k, and then passed through the pre-trained convolutional layer of Sawada and Nakamura [120], yielding a 2048-dimensional feature vector that serves as input to the concept extractor (Section D.2).

Knowledge. We use the same prior knowledge K as Marconato et al. [24, 9]. For the `move_forward` and `stop` actions, the rules are:

$$\begin{aligned}
&\texttt{red_light} \Rightarrow \neg\texttt{green_light},\\
&\texttt{obstacle} = \texttt{car} \vee \texttt{person} \vee \texttt{rider} \vee \texttt{other_obstacle},\\
&\texttt{road_clear} \iff \neg\texttt{obstacle},\\
&\texttt{green_light} \vee \texttt{follow} \vee \texttt{road_clear} \Rightarrow \texttt{move_forward},\\
&\texttt{red_light} \vee \texttt{traffic_sign} \vee \texttt{obstacle} \Rightarrow \texttt{stop},\\
&\texttt{stop} \Rightarrow \neg\texttt{move_forward}.
\end{aligned} \tag{138}$$

For the `turn_left` action, the rules are:

$$\begin{aligned}
&\texttt{can_turn} = \texttt{left_lane} \vee \texttt{left_green_light} \vee \texttt{left_follow},\\
&\texttt{cannot_turn} = \texttt{no_left_lane} \vee \texttt{left_obstacle} \vee \texttt{left_solid_line},\\
&\texttt{can_turn} \wedge \neg\texttt{cannot_turn} \Rightarrow \texttt{turn_left},
\end{aligned} \tag{139}$$

and analogously for `turn_right`, using the corresponding right-side concepts.

Decomposition into sub-tasks. As shown by the rules above, each action depends only on a subset of the concepts. Following Marconato et al. [9], Bortolotti et al. [94], we therefore split the task into three independent sub-tasks, summarized in Table 4: forward/stop (F-S), which depends on 9 concepts, and left (L) and right (R), which depend on 6 concepts each. This reduces the number of worlds to enumerate from 2^{21} to $2^9 = 512$ for F-S and $2^6 = 64$ for L and R, which makes the construction of $\mathbf{\Gamma}^{\times}(\mathbf{x})$ and the computation of $g^{\dagger}$ and $g^{-\dagger}$ tractable (Section E.1). Conformal sets, coverage, size, and consistency are computed separately for each sub-task.

Table 4: Concept groups for the three `BDD-OIA` sub-tasks.

SUB-TASK	CONCEPTS
F-S (9)	`green_light`, `follow`, `road_clear`, `red_light`, `traffic_sign`, `car`, `person`, `rider`, `other_obstacle`
L (6)	`left_lane`, `left_green_light`, `left_follow`, `no_left_lane`, `left_obstacle`, `left_solid_line`
R (6)	`right_lane`, `right_green_light`, `right_follow`, `no_right_lane`, `right_obstacle`, `right_solid_line`

Reasoning Shortcuts. `BDD-OIA` is affected by a very large number of reasoning shortcuts [24, 9]. For instance, since both `red_light` and `person` entail `stop`, a model can confuse pedestrians with red lights while still predicting the correct action. Moreover, the knowledge is not abductively sound for the L and R sub-tasks, i.e., $\delta_{\text{ab}} < 1$ (Section E.1).

D Architecture Details & Model Selection

D.1 Model Selection

All experiments use either the `Adam` [121] or `SGD` [122] optimizer, selected as part of the hyperparameter search. Hyperparameters were selected through Bayesian optimization via `Optuna` [123], using a `BoTorch` sampler [123] with a `TPE` sampler [124] for the initial startup trials, maximizing the macro F1 score of the final label on a validation set. Each study ran for `<M>` (at least 20) trials, with 10 warm-up trials before the Bayesian model is fitted. All experiments were run for `<N>` epochs employing early stopping, saving the checkpoint which achieves the highest F1 score on the validation set. The learning rate γ was selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, and the momentum ι from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5, 0.9, 0.99\}$. The batch size ν was chosen from $\{32, 64, 128, 256\}$.

For `LTN`, a consistent logic family was selected jointly for all operators, choosing among Gödel, Product, and Łukasiewicz. Under Product logic, the implication operator defaults to Goguen implication for consistency. The quantifier parameter p was searched over integer values in the range $[1, 9]$. For `LTN`, since the training objective can be hard to optimize since the p parameter governs the sharpness of the existential and universal quantifiers. We therefore linearly anneal p from 1 to the target value p^* across all training epochs:

$$p_t = \text{round}\left(1 + (p^* - 1)\,\frac{t}{T-1}\right), \quad t = 0, \ldots, T-1, \tag{140}$$

where T is the total number of epochs. This keeps the logic soft early in training and progressively makes it crisper as concept representations stabilize. Similarly, when concept supervision is available, we suppress concept supervision for the first 25% of training and then linearly ramp the concept-loss weight from 0 to 1.

All models were trained for 200 epochs with early stopping, except for `CEB`, where models were trained for 500 epochs. Early stopping selects the checkpoint achieving the highest F1 score on the validation set with respect to the final label. All experiments have been tested with the same 10 seeds for consistency, specifically $\{1011, 1213, 1415, 1617, 1819, 2021, 2223, 2425, 2627, 2829\}$.

Below, you find all the hyperparameters which performed the best on our datasets.

Hyperparameters for `MN-Add`.

- `DPL`, $\gamma = 10^{-1}$, $\nu = 32$, $\iota = 0.1$, optimizer = `SGD`;
- `LTN`, $\gamma = 10^{-1}$, $\nu = 64$, $\iota = 10^{-5}$, optimizer = `SGD`, $p = 8$, AND/OR/IMPLICATION set to Product.

Hyperparameters for `CHX`.

- `DPL` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 128$, $\iota = 10^{-2}$, optimizer = `Adam`;
- `LTN` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 256$, $\iota = 0.9$, optimizer = `Adam`, $p = 4$, AND/OR/IMPLICATION set to Product.

Hyperparameters for `RIV10`:

- `DPL` (with concept supervision), $\gamma = 10^{-3}$, $\nu = 128$, $\iota = 0.99$, optimizer = `SGD`;
- `DPL` (without concept supervision), $\gamma = 10^{-4}$, $\nu = 128$, $\iota = 0.99$, optimizer = `SGD`;
- `LTN` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 256$, $\iota = 0.9$, optimizer = `Adam`, $p = 4$, AND/OR/IMPLICATION set to Product.

Hyperparameters for `CIF10`:

- `DPL` (with concept supervision), $\gamma = 10^{-3}$, $\nu = 128$, $\iota = 0.99$, optimizer = `SGD`;
- `DPL` (without concept supervision), $\gamma = 10^{-4}$, $\nu = 128$, $\iota = 10^{-2}$, optimizer = `Adam`;
- `LTN` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 256$, $\iota = 0.9$, optimizer = `Adam`, $p = 4$, AND/OR/IMPLICATION set to Product.

Hyperparameters for `CEB`:

- `DPL` (with concept supervision), $\gamma = 10^{-2}$, $\nu = 64$, $\iota = 1e-05$, optimizer = `Adam`;
- `DPL` (without concept supervision), $\gamma = 10^{-4}$, $\nu = 128$, $\iota = 10^{-4}$, optimizer = `Adam`;
- `LTN` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 256$, $\iota = 0.9$, optimizer = `Adam`, $p = 4$, AND/OR/IMPLICATION set to Product.

Hyperparameters for other `MNIST`**-based datasets**:

- `DPL` (`MN-H`, `MN-SP`, `MN-EO`), $\gamma = 10^{-1}$, $\nu = 32$, $\iota = 10^{-1}$, optimizer = `SGD`;
- `LTN` (`MN-H`, `MN-SP`), $\gamma = 10^{-4}$, $\nu = 256$, $\iota = 0.9$, optimizer = `Adam`, $p = 4$, AND/OR/IMPLICATION set to Product.
- `LTN` (`MN-EO`), $\gamma = 10^{-2}$, $\nu = 64$, $\iota = 0.99$, optimizer = `Adam`, $p = 3$, AND/OR/IMPLICATION set to Gödel.

Hyperparameters for `BDD-OIA`:

- `DPL` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 128$, $\iota = 10^{-2}$, optimizer = `Adam`;
- `LTN` (with and without concept supervision), $\gamma = 10^{-4}$, $\nu = 128$, $\iota = 10^{-1}$, optimizer = `SGD`, $p = 9$, AND/OR/IMPLICATION set to Product.

D.2 Model Architectures

We describe below the three neural architectures used as concept extractors f across our experiments.

LeNet5. LeNet5 [125] is used as backbone for `MNIST` [7] images (i.e.,`MN-Add` and its variants). The architecture takes as input a single-channel 28×28 image and produces a `num_concepts`-dimensional output. Table 5 details the layer configuration.

Table 5: LeNet5 architecture used for `MN-Add` and its variants.

INPUT	LAYER TYPE	PARAMETER	ACTIVATION
$(1, 28, 28)$	Conv2d	depth= 6, kernel= 5, stride= 1, padding= 2	Tanh
$(6, 28, 28)$	AvgPool2d	kernel= 2, stride= 2	
$(6, 14, 14)$	Conv2d	depth= 16, kernel= 5, stride= 1	Tanh
$(16, 10, 10)$	AvgPool2d	kernel= 2, stride= 2	
$(16, 5, 5)$	Flatten	dim= 400	
(400)	Linear	$400 \to 120$	Tanh
(120)	Linear	$120 \to 84$	Tanh
(84)	Linear	$84 \to$ `num_concepts`	

ResNet18. ResNet18 [126] is used as backbone for higher-resolution image datasets (i.e.,`CHX`, `CIF10`, and `RIV10`). We use weights pretrained on ImageNet [116] for both `CIF10` and `CHX`, but not for `RIV10`, as its images are sourced from the same `ImageNet` [116] dataset on which ResNet-18 was pretrained. For all datasets, we resize the images to 224×224 to make them compatible with the ResNet18 architecture. The final fully connected layer is replaced with a linear layer mapping to `num_concepts` outputs. Table 6 details the layer configuration.

Table 6: ResNet18 architecture used for `CHX`, `CIF10`, and `RIV10`.

INPUT	LAYER TYPE	PARAMETER	ACTIVATION
$(3, 224, 224)$	Conv2d	depth= 64, kernel= 7, stride= 2, padding= 3	
$(64, 112, 112)$	BatchNorm2d	dim= 64	ReLU
$(64, 112, 112)$	MaxPool2d	kernel= 3, stride= 2, padding= 1	
$(64, 56, 56)$	2×BasicBlock	depth= 64, kernel= $(1, 1)$, stride= $(1, 1)$, padding= $(1, 1)$	
$(128, 28, 28)$	2×BasicBlock	depth= 128, kernel= $(1, 1)$, stride= $(2, 2)$, padding= $(1, 1)$	
$(256, 14, 14)$	2×BasicBlock	depth= 256, kernel= $(1, 1)$, stride= $(2, 2)$, padding= $(1, 1)$	
$(512, 7, 7)$	2×BasicBlock	depth= 512, kernel= $(1, 1)$, stride= $(2, 2)$, padding= $(1, 1)$	
$(512, 7, 7)$	AvgPool2d	dim= $(1, 1)$	
$(512, 1, 1)$	Flatten	dim= 512	
(512)	Linear	$512 \to$ `num_concepts`	

BERT. `bert-base-uncased` [127] is used as backbone for the text dataset `CEB`. We adopt the standard pretrained `bert-base-uncased` encoder, freezing its weights, and append a trainable MLP classification head producing `num_concepts` × `num_concept_dim` logits, which are reshaped into per-concept predictions of dimension `num_concept_dim`. Table 7 details the layer configuration. Table 7 details the layer configuration.

Table 7: BERT architecture used for `CEB`.

INPUT	LAYER TYPE	PARAMETER	ACTIVATION
Token sequence	`bert-base-uncased` (frozen)	hidden dim= 768, layers= 12, heads= 12	
`[CLS]` token (768)	Linear	$768 \to 256$	ReLU
(256)	Dropout	$p = 0.1$	
(256)	Linear	$256 \to 128$	ReLU
(128)	Dropout	$p = 0.1$	
(128)	Linear	$128 \to$ `num_concepts` × `num_concept_dim`	
(`num_concepts` × `num_concept_dim`)	Reshape	$\to$ (`num_concepts`, `num_concept_dim`)	

Linear. For `BDD-OIA`, raw images are not processed directly. Instead, we use 2048-dimensional visual features pre-extracted by the ResNet-50 backbone provided by rsbench [94]. The concept extractor f is a single fully-connected layer that maps these features to `num_concepts` $= 21$ concept logits, one per road-scene binary concept.

Table 8: Linear architecture for BDD-OIA

INPUT	LAYER TYPE	PARAMETER	ACTIVATION
(2048)	Linear	out=21	

E Additional Experiments

In this section, we report experiments on additional datasets and under different settings (with or without concept supervision), as well as configurations involving e-values. All tables, by default, use concept supervision unless otherwise specified. We also provide two ablation studies on the proposed method for encoding conformal predictions in NeSy-CBMs, along with a runtime analysis.

Table 9: Results for DPL (without concept supervision) on MN-Add.

	CONCEPTS			LABELS		
METHOD	CONST.	SIZE	COV.	CONST.	SIZE	COV.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00
DPL + TO	0.90 ± 0.01	1.00 ± 0.00	0.98 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	6.01 ± 0.05	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
DPL + CO	1.00 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	0.91 ± 0.01	1.00 ± 0.00	0.98 ± 0.00
DPL + CDe	1.00 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	1.00 ± 0.00	0.91 ± 0.01	0.90 ± 0.01
DPL + RPB	1.00 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
DPL + COCOCO	1.00 ± 0.00	0.99 ± 0.00	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01

Table 10: Results for LTN (without concept supervision) on MN-Add.

	CONCEPTS			LABELS		
METHOD	CONST.	SIZE	COV.	CONST.	SIZE	COV.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.00
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.00
LTN + TAb	1.00 ± 0.00	6.66 ± 0.02	0.97 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.00
LTN + CO	1.00 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01	1.00 ± 0.00	0.97 ± 0.00
LTN + CDe	1.00 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
LTN + RPB	1.00 ± 0.00	1.00 ± 0.00	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
LTN + COCOCO	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01

Table 11: Results for DPL (without concept supervision) on CEB.

	CONCEPTS			LABELS		
METHOD	CONST.	SIZE	COV.	CONST.	SIZE	COV.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.61 ± 0.01
DPL + TO	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.33 ± 0.14	3.59 ± 1.49	0.94 ± 0.06
DPL + TAb	1.00 ± 0.00	742.55 ± 296.73	0.94 ± 0.06	1.00 ± 0.00	3.59 ± 1.49	0.94 ± 0.06
DPL + CO	0.23 ± 0.01	907.88 ± 76.19	0.97 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.61 ± 0.01
DPL + CDe	1.00 ± 0.00	907.88 ± 76.19	0.97 ± 0.02	1.00 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
DPL + RPB	0.74 ± 0.27	907.88 ± 76.19	0.97 ± 0.02	1.00 ± 0.00	3.59 ± 1.49	0.94 ± 0.06
DPL + COCOCO	1.00 ± 0.00	670.15 ± 252.46	0.92 ± 0.06	1.00 ± 0.00	3.59 ± 1.49	0.94 ± 0.06

Table 12: Results for DPL (with concept supervision) on CEB.

	CONCEPTS			LABELS		
METHOD	CONST.	SIZE	COV.	CONST.	SIZE	COV.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.18 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.58 ± 0.03
DPL + TO	1.00 ± 0.00	1.00 ± 0.00	0.18 ± 0.02	0.42 ± 0.03	2.41 ± 0.18	0.90 ± 0.02
DPL + TAb	1.00 ± 0.00	504.38 ± 34.71	0.90 ± 0.02	1.00 ± 0.00	2.41 ± 0.18	0.90 ± 0.02
DPL + CO	0.39 ± 0.05	75.06 ± 38.35	0.90 ± 0.04	0.97 ± 0.01	1.00 ± 0.00	0.58 ± 0.03
DPL + CDe	1.00 ± 0.00	75.06 ± 38.35	0.90 ± 0.04	1.00 ± 0.00	3.71 ± 0.65	0.96 ± 0.02
DPL + RPB	0.78 ± 0.06	75.06 ± 38.35	0.90 ± 0.04	1.00 ± 0.00	2.31 ± 0.16	0.87 ± 0.01
DPL + COCOCO	1.00 ± 0.00	53.49 ± 25.93	0.83 ± 0.03	1.00 ± 0.00	2.31 ± 0.16	0.87 ± 0.01

Table 13: Results for LTN (without concept supervision) on CEB.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.62 ± 0.01
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.20 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	1024.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.30 ± 0.01	349.70 ± 34.12	0.91 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.62 ± 0.01
LTN + CDe	1.00 ± 0.00	349.70 ± 34.12	0.91 ± 0.02	1.00 ± 0.00	4.31 ± 0.18	0.99 ± 0.01
LTN + RPB	1.00 ± 0.00	349.70 ± 34.12	0.91 ± 0.02	1.00 ± 0.00	4.31 ± 0.18	0.99 ± 0.01
LTN + COCOCO	1.00 ± 0.00	349.70 ± 34.12	0.91 ± 0.02	1.00 ± 0.00	4.31 ± 0.18	0.99 ± 0.01

Table 14: Results for LTN (with concept supervision) on CEB.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.24 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.63 ± 0.01
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.24 ± 0.01	0.20 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	1024.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.41 ± 0.02	36.19 ± 4.07	0.90 ± 0.02	0.98 ± 0.01	1.00 ± 0.00	0.63 ± 0.01
LTN + CDe	1.00 ± 0.00	36.19 ± 4.07	0.90 ± 0.02	1.00 ± 0.00	3.13 ± 0.13	0.96 ± 0.01
LTN + RPB	1.00 ± 0.00	36.19 ± 4.07	0.90 ± 0.02	1.00 ± 0.00	3.13 ± 0.13	0.96 ± 0.01
LTN + COCOCO	1.00 ± 0.00	36.05 ± 4.08	0.90 ± 0.02	1.00 ± 0.00	3.13 ± 0.13	0.96 ± 0.01

Table 15: Results for DPL (without concept supervision) on CHX.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.45 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	0.73 ± 0.01
DPL + TO	1.00 ± 0.00	1.00 ± 0.00	0.45 ± 0.09	0.61 ± 0.08	1.67 ± 0.21	0.90 ± 0.03
DPL + TAb	1.00 ± 0.00	5.05 ± 0.92	0.90 ± 0.03	1.00 ± 0.00	1.67 ± 0.21	0.90 ± 0.03
DPL + CO	0.28 ± 0.06	6.85 ± 2.53	0.91 ± 0.02	1.00 ± 0.01	1.00 ± 0.00	0.73 ± 0.01
DPL + CDe	1.00 ± 0.00	6.85 ± 2.53	0.91 ± 0.02	1.00 ± 0.00	3.50 ± 0.49	0.99 ± 0.01
DPL + RPB	0.52 ± 0.12	6.85 ± 2.53	0.91 ± 0.02	1.00 ± 0.00	1.66 ± 0.22	0.90 ± 0.04
DPL + COCOCO	1.00 ± 0.00	3.36 ± 0.80	0.84 ± 0.04	1.00 ± 0.00	1.66 ± 0.22	0.90 ± 0.04

Table 16: Results for DPL (with concept supervision) on CHX.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.47 ± 0.11	1.00 ± 0.00	1.00 ± 0.00	0.72 ± 0.01
DPL + TO	1.00 ± 0.00	1.00 ± 0.00	0.47 ± 0.11	0.62 ± 0.07	1.63 ± 0.19	0.89 ± 0.03
DPL + TAb	1.00 ± 0.00	4.96 ± 0.86	0.89 ± 0.03	1.00 ± 0.00	1.63 ± 0.19	0.89 ± 0.03
DPL + CO	0.31 ± 0.04	5.30 ± 1.01	0.91 ± 0.03	0.99 ± 0.01	1.00 ± 0.00	0.72 ± 0.01
DPL + CDe	1.00 ± 0.00	5.30 ± 1.01	0.91 ± 0.03	1.00 ± 0.00	3.19 ± 0.30	0.98 ± 0.01
DPL + RPB	0.55 ± 0.05	5.30 ± 1.01	0.91 ± 0.03	1.00 ± 0.00	1.60 ± 0.18	0.88 ± 0.03
DPL + COCOCO	1.00 ± 0.00	2.90 ± 0.56	0.83 ± 0.04	1.00 ± 0.00	1.60 ± 0.18	0.88 ± 0.03

Table 17: Results for LTN (without concept supervision) on CHX.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.45 ± 0.10	1.00 ± 0.00	1.00 ± 0.00	0.73 ± 0.02
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.45 ± 0.10	0.20 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	16.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.24 ± 0.05	7.58 ± 2.49	0.91 ± 0.02	0.99 ± 0.01	1.00 ± 0.00	0.73 ± 0.02
LTN + CDe	1.00 ± 0.00	7.58 ± 2.49	0.91 ± 0.02	1.00 ± 0.00	3.68 ± 0.44	0.98 ± 0.01
LTN + RPB	1.00 ± 0.00	7.58 ± 2.49	0.91 ± 0.02	1.00 ± 0.00	3.68 ± 0.44	0.98 ± 0.01
LTN + COCOCO	1.00 ± 0.00	7.54 ± 2.51	0.91 ± 0.02	1.00 ± 0.00	3.68 ± 0.44	0.98 ± 0.01

Table 18: Results for LTN (with concept supervision) on CHX.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.46 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	0.74 ± 0.02
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.46 ± 0.09	0.20 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	16.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	5.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.26 ± 0.04	6.58 ± 1.64	0.92 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.74 ± 0.02
LTN + CDe	1.00 ± 0.00	6.58 ± 1.64	0.92 ± 0.01	1.00 ± 0.00	3.53 ± 0.35	0.99 ± 0.01
LTN + RPB	1.00 ± 0.00	6.58 ± 1.64	0.92 ± 0.01	1.00 ± 0.00	3.53 ± 0.35	0.99 ± 0.01
LTN + COCOCO	1.00 ± 0.00	6.56 ± 1.66	0.92 ± 0.01	1.00 ± 0.00	3.53 ± 0.35	0.99 ± 0.01

Table 19: Results for DPL (without concept supervision) on CIF10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	0.97 ± 0.05	1.00 ± 0.00	0.34 ± 0.02	0.97 ± 0.05	1.00 ± 0.00	0.66 ± 0.00
DPL + TO	0.99 ± 0.01	1.00 ± 0.00	0.34 ± 0.02	0.84 ± 0.05	1.83 ± 0.08	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	171.66 ± 7.97	0.90 ± 0.01	1.00 ± 0.00	1.83 ± 0.08	0.90 ± 0.01
DPL + CO	0.77 ± 0.05	57.60 ± 24.96	0.97 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.66 ± 0.00
DPL + CDe	1.00 ± 0.00	57.60 ± 24.96	0.97 ± 0.01	1.00 ± 0.00	5.54 ± 0.98	0.78 ± 0.01
DPL + RPB	0.79 ± 0.05	57.60 ± 24.96	0.97 ± 0.01	1.00 ± 0.00	1.83 ± 0.08	0.90 ± 0.01
DPL + COCOCO	1.00 ± 0.00	45.43 ± 19.68	0.90 ± 0.01	1.00 ± 0.00	1.83 ± 0.08	0.90 ± 0.01

Table 20: Results for DPL (with concept supervision) on CIF10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.34 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.66 ± 0.00
DPL + TO	1.00 ± 0.00	1.00 ± 0.00	0.34 ± 0.01	0.88 ± 0.02	1.76 ± 0.06	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	164.21 ± 5.73	0.90 ± 0.01	1.00 ± 0.00	1.76 ± 0.06	0.90 ± 0.01
DPL + CO	0.88 ± 0.02	20.75 ± 2.33	0.93 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.66 ± 0.00
DPL + CDe	1.00 ± 0.00	20.75 ± 2.33	0.93 ± 0.01	1.00 ± 0.00	2.89 ± 0.25	0.75 ± 0.00
DPL + RPB	0.91 ± 0.02	20.75 ± 2.33	0.93 ± 0.01	1.00 ± 0.00	1.76 ± 0.06	0.90 ± 0.01
DPL + COCOCO	1.00 ± 0.00	18.87 ± 2.12	0.88 ± 0.01	1.00 ± 0.00	1.76 ± 0.06	0.90 ± 0.01

Table 21: Results for LTN (without concept supervision) on CIF10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	0.98 ± 0.00	1.00 ± 0.00	0.13 ± 0.04	0.98 ± 0.00	1.00 ± 0.00	0.50 ± 0.03
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.13 ± 0.04	0.73 ± 0.05	3.09 ± 0.16	0.90 ± 0.01
LTN + TAb	1.00 ± 0.00	288.11 ± 15.71	0.90 ± 0.00	1.00 ± 0.00	3.09 ± 0.16	0.90 ± 0.01
LTN + CO	0.84 ± 0.04	28.83 ± 3.66	0.93 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.03
LTN + CDe	1.00 ± 0.00	28.83 ± 3.66	0.93 ± 0.01	1.00 ± 0.00	3.15 ± 0.11	0.75 ± 0.01
LTN + RPB	0.94 ± 0.03	28.83 ± 3.66	0.93 ± 0.01	1.00 ± 0.00	3.08 ± 0.17	0.90 ± 0.01
LTN + COCOCO	1.00 ± 0.00	27.09 ± 2.77	0.88 ± 0.01	1.00 ± 0.00	3.08 ± 0.17	0.90 ± 0.01

Table 22: Results for LTN (with concept supervision) on CIF10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	0.98 ± 0.00	1.00 ± 0.00	0.31 ± 0.01	0.98 ± 0.00	1.00 ± 0.00	0.58 ± 0.01
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.31 ± 0.01	0.77 ± 0.03	2.06 ± 0.14	0.89 ± 0.01
LTN + TAb	1.00 ± 0.00	196.29 ± 13.29	0.90 ± 0.01	1.00 ± 0.00	2.06 ± 0.14	0.89 ± 0.01
LTN + CO	0.85 ± 0.03	19.65 ± 2.28	0.93 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.58 ± 0.01
LTN + CDe	1.00 ± 0.00	19.65 ± 2.28	0.93 ± 0.01	1.00 ± 0.00	3.12 ± 0.14	0.75 ± 0.01
LTN + RPB	0.93 ± 0.02	19.65 ± 2.28	0.93 ± 0.01	1.00 ± 0.00	2.06 ± 0.14	0.89 ± 0.01
LTN + COCOCO	1.00 ± 0.00	18.22 ± 1.79	0.88 ± 0.01	1.00 ± 0.00	2.06 ± 0.14	0.89 ± 0.01

Table 23: Results for DPL (without concept supervision) on RIV10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.41 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	0.78 ± 0.00
DPL + TO	0.92 ± 0.01	1.00 ± 0.00	0.41 ± 0.03	1.00 ± 0.00	1.24 ± 0.01	0.91 ± 0.01
DPL + TAb	1.00 ± 0.00	116.77 ± 0.97	0.91 ± 0.01	1.00 ± 0.00	1.24 ± 0.01	0.91 ± 0.01
DPL + CO	0.97 ± 0.03	8.11 ± 2.22	0.93 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.78 ± 0.00
DPL + CDe	1.00 ± 0.00	8.11 ± 2.22	0.93 ± 0.01	1.00 ± 0.00	2.06 ± 0.26	0.76 ± 0.00
DPL + RPB	0.92 ± 0.03	8.11 ± 2.22	0.93 ± 0.01	1.00 ± 0.00	1.23 ± 0.02	0.90 ± 0.01
DPL + COCOCO	1.00 ± 0.00	7.31 ± 1.77	0.86 ± 0.01	1.00 ± 0.00	1.23 ± 0.02	0.90 ± 0.01

Table 24: Results for DPL (with concept supervision) on RIV10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.43 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.80 ± 0.00
DPL + TO	0.91 ± 0.01	1.00 ± 0.00	0.43 ± 0.02	1.00 ± 0.00	1.21 ± 0.02	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	114.20 ± 1.38	0.90 ± 0.01	1.00 ± 0.00	1.21 ± 0.02	0.90 ± 0.01
DPL + CO	0.99 ± 0.00	6.33 ± 0.65	0.92 ± 0.01	0.97 ± 0.01	1.00 ± 0.00	0.80 ± 0.00
DPL + CDe	1.00 ± 0.00	6.33 ± 0.65	0.92 ± 0.01	1.00 ± 0.00	1.71 ± 0.10	0.74 ± 0.01
DPL + RPB	0.94 ± 0.01	6.33 ± 0.65	0.92 ± 0.01	1.00 ± 0.00	1.20 ± 0.02	0.89 ± 0.01
DPL + COCOCO	1.00 ± 0.00	5.85 ± 0.62	0.84 ± 0.01	1.00 ± 0.00	1.20 ± 0.02	0.89 ± 0.01

Table 25: Results for LTN (without concept supervision) on RIV10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	0.99 ± 0.02	1.00 ± 0.00	0.14 ± 0.11	0.99 ± 0.02	1.00 ± 0.00	0.56 ± 0.12
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.14 ± 0.11	0.90 ± 0.04	2.15 ± 0.38	0.90 ± 0.02
LTN + TAb	1.00 ± 0.00	201.12 ± 33.62	0.90 ± 0.02	1.00 ± 0.00	2.15 ± 0.38	0.90 ± 0.02
LTN + CO	0.93 ± 0.03	12.00 ± 1.78	0.92 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.56 ± 0.12
LTN + CDe	1.00 ± 0.00	12.00 ± 1.78	0.92 ± 0.01	1.00 ± 0.00	1.97 ± 0.08	0.74 ± 0.01
LTN + RPB	0.99 ± 0.00	12.00 ± 1.78	0.92 ± 0.01	1.00 ± 0.00	2.05 ± 0.30	0.89 ± 0.02
LTN + COCOCO	1.00 ± 0.00	11.65 ± 1.74	0.85 ± 0.01	1.00 ± 0.00	2.05 ± 0.30	0.89 ± 0.02

Table 26: Results for LTN (with concept supervision) on RIV10.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	0.99 ± 0.00	1.00 ± 0.00	0.41 ± 0.02	0.99 ± 0.00	1.00 ± 0.00	0.72 ± 0.01
LTN + TO	0.97 ± 0.01	1.00 ± 0.00	0.41 ± 0.02	0.99 ± 0.01	1.35 ± 0.03	0.90 ± 0.01
LTN + TAb	1.00 ± 0.00	128.86 ± 2.72	0.90 ± 0.01	1.00 ± 0.00	1.35 ± 0.03	0.90 ± 0.01
LTN + CO	0.97 ± 0.01	9.41 ± 0.72	0.92 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.72 ± 0.01
LTN + CDe	1.00 ± 0.00	9.41 ± 0.72	0.92 ± 0.01	1.00 ± 0.00	1.84 ± 0.12	0.75 ± 0.01
LTN + RPB	0.96 ± 0.01	9.41 ± 0.72	0.92 ± 0.01	1.00 ± 0.00	1.33 ± 0.02	0.89 ± 0.01
LTN + COCOCO	1.00 ± 0.00	8.85 ± 0.68	0.85 ± 0.01	1.00 ± 0.00	1.33 ± 0.02	0.89 ± 0.01

Table 27: Results for DPL (without concept supervision) on MN-EO.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.13 ± 0.08	1.00 ± 0.00	1.00 ± 0.00	0.72 ± 0.01
DPL + TO	1.00 ± 0.00	1.00 ± 0.00	0.13 ± 0.08	0.65 ± 0.07	1.55 ± 0.18	0.90 ± 0.02
DPL + TAb	1.00 ± 0.00	12.13 ± 1.37	0.90 ± 0.02	1.00 ± 0.00	1.55 ± 0.18	0.90 ± 0.02
DPL + CO	0.08 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.72 ± 0.01
DPL + CDe	1.00 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	19.00 ± 0.00	1.00 ± 0.00
DPL + RPB	0.12 ± 0.01	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.55 ± 0.18	0.90 ± 0.02
DPL + COCOCO	1.00 ± 0.00	12.13 ± 1.37	0.90 ± 0.02	1.00 ± 0.00	1.55 ± 0.18	0.90 ± 0.02

Table 28: Results for LTN (without concept supervision) on MN-EO.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.01
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.01
LTN + TAb	1.00 ± 0.00	8.05 ± 0.05	0.94 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.01
LTN + CO	0.08 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.01
LTN + CDe	1.00 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	19.00 ± 0.00	1.00 ± 0.00
LTN + RPB	0.08 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.01
LTN + COCOCO	1.00 ± 0.00	8.05 ± 0.05	0.94 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.01

Table 29: Results for DPL (without concept supervision) on MN-H.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.01
DPL + TO	0.89 ± 0.05	1.00 ± 0.00	0.50 ± 0.03	1.00 ± 0.00	0.89 ± 0.05	0.89 ± 0.05
DPL + TAb	1.00 ± 0.00	2.20 ± 0.14	0.89 ± 0.05	1.00 ± 0.00	0.89 ± 0.05	0.89 ± 0.05
DPL + CO	0.10 ± 0.00	25.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.01
DPL + CDe	1.00 ± 0.00	25.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	9.00 ± 0.00	1.00 ± 0.00
DPL + RPB	0.09 ± 0.01	25.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.89 ± 0.05	0.89 ± 0.05
DPL + COCOCO	1.00 ± 0.00	2.20 ± 0.14	0.89 ± 0.05	1.00 ± 0.00	0.89 ± 0.05	0.89 ± 0.05

Table 30: Results for LTN (without concept supervision) on MN-H.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	0.83 ± 0.12
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.03	0.47 ± 0.46	5.80 ± 4.13	0.99 ± 0.02
LTN + TAb	1.00 ± 0.00	16.00 ± 11.62	0.99 ± 0.02	1.00 ± 0.00	5.80 ± 4.13	0.99 ± 0.02
LTN + CO	0.10 ± 0.01	19.48 ± 1.70	0.87 ± 0.04	1.00 ± 0.00	1.00 ± 0.00	0.83 ± 0.12
LTN + CDe	1.00 ± 0.00	19.48 ± 1.70	0.87 ± 0.04	1.00 ± 0.00	8.62 ± 0.24	1.00 ± 0.00
LTN + RPB	0.64 ± 0.46	19.48 ± 1.70	0.87 ± 0.04	1.00 ± 0.00	5.60 ± 3.97	0.99 ± 0.02
LTN + COCOCO	1.00 ± 0.00	12.77 ± 9.35	0.86 ± 0.04	1.00 ± 0.00	5.60 ± 3.97	0.99 ± 0.02

Table 31: Results for DPL (without concept supervision) on MN-SP.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
DPL + TO	0.90 ± 0.01	1.00 ± 0.00	0.03 ± 0.02	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	45.19 ± 0.35	0.90 ± 0.01	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
DPL + CO	0.81 ± 0.22	46.81 ± 36.72	0.93 ± 0.05	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
DPL + CDe	1.00 ± 0.00	46.81 ± 36.72	0.93 ± 0.05	1.00 ± 0.00	1.34 ± 0.46	1.00 ± 0.00
DPL + RPB	0.73 ± 0.19	46.81 ± 36.72	0.93 ± 0.05	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01
DPL + COCOCO	1.00 ± 0.00	27.92 ± 12.05	0.84 ± 0.05	1.00 ± 0.00	0.90 ± 0.01	0.90 ± 0.01

Table 32: Results for LTN (without concept supervision) on MN-SP.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00
LTN + TAb	1.00 ± 0.00	50.00 ± 0.00	0.98 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00
LTN + CO	0.50 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00
LTN + CDe	1.00 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + RPB	0.50 ± 0.00	100.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00
LTN + COCOCO	1.00 ± 0.00	50.00 ± 0.00	0.98 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.00

Table 33: Results for DPL (without concept supervision) on BDD-OIA on Forward-Stop Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL †	0.61 ± 0.04	1.00 ± 0.00	0.04 ± 0.00	0.61 ± 0.04	1.00 ± 0.00	0.74 ± 0.00
DPL + TO †	0.85 ± 0.02	1.00 ± 0.00	0.04 ± 0.00	0.32 ± 0.01	2.67 ± 0.07	0.94 ± 0.01
DPL + TAb †	0.71 ± 0.02	396.58 ± 5.54	0.61 ± 0.02	0.41 ± 0.01	2.67 ± 0.07	0.94 ± 0.01
DPL + CO †	0.39 ± 0.04	192.39 ± 49.91	0.92 ± 0.01	0.99 ± 0.00	1.00 ± 0.00	0.74 ± 0.00
DPL + CDe †	0.99 ± 0.00	192.39 ± 49.91	0.92 ± 0.01	0.98 ± 0.00	1.94 ± 0.02	0.91 ± 0.01
DPL + RPB †	0.80 ± 0.04	192.39 ± 49.91	0.92 ± 0.01	1.00 ± 0.00	1.49 ± 0.03	0.86 ± 0.01
DPL + COCOCO †	1.00 ± 0.00	149.99 ± 35.63	0.87 ± 0.01	1.00 ± 0.00	1.49 ± 0.03	0.86 ± 0.01

Table 34: Results for DPL (without concept supervision) on BDD-OIA on Left Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL †	0.67 ± 0.01	1.00 ± 0.00	0.21 ± 0.02	0.67 ± 0.01	1.00 ± 0.00	0.74 ± 0.00
DPL + TO †	0.85 ± 0.01	1.00 ± 0.00	0.21 ± 0.02	0.60 ± 0.01	1.41 ± 0.03	0.90 ± 0.01
DPL + TAb †	1.00 ± 0.00	52.61 ± 0.76	0.70 ± 0.02	1.00 ± 0.00	1.41 ± 0.03	0.90 ± 0.01
DPL + CO †	0.63 ± 0.04	25.98 ± 7.55	0.91 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.74 ± 0.00
DPL + CDe †	1.00 ± 0.00	25.98 ± 7.55	0.91 ± 0.01	1.00 ± 0.00	1.93 ± 0.02	0.97 ± 0.01
DPL + RPB †	0.82 ± 0.03	25.98 ± 7.55	0.91 ± 0.01	1.00 ± 0.00	1.35 ± 0.03	0.86 ± 0.01
DPL + COCOCO †	1.00 ± 0.00	21.41 ± 7.02	0.64 ± 0.02	1.00 ± 0.00	1.35 ± 0.03	0.86 ± 0.01

Table 35: Results for DPL (without concept supervision) on BDD-OIA on Right Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL †	0.32 ± 0.03	1.00 ± 0.00	0.19 ± 0.02	0.32 ± 0.03	1.00 ± 0.00	0.73 ± 0.00
DPL + TO †	0.59 ± 0.03	1.00 ± 0.00	0.19 ± 0.02	0.41 ± 0.02	1.44 ± 0.03	0.90 ± 0.01
DPL + TAb †	1.00 ± 0.00	52.92 ± 0.92	0.85 ± 0.02	1.00 ± 0.00	1.44 ± 0.03	0.90 ± 0.01
DPL + CO †	0.61 ± 0.02	27.69 ± 8.18	0.91 ± 0.01	0.97 ± 0.01	1.00 ± 0.00	0.73 ± 0.00
DPL + CDe †	1.00 ± 0.00	27.69 ± 8.18	0.91 ± 0.01	1.00 ± 0.00	1.94 ± 0.01	0.97 ± 0.01
DPL + RPB †	0.84 ± 0.02	27.69 ± 8.18	0.91 ± 0.01	1.00 ± 0.00	1.38 ± 0.03	0.86 ± 0.01
DPL + COCOCO †	1.00 ± 0.00	23.20 ± 7.25	0.77 ± 0.01	1.00 ± 0.00	1.38 ± 0.03	0.86 ± 0.01

Table 36: Results for DPL (with concept supervision) on BDD-OIA on Forward-Stop Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	0.61 ± 0.04	1.00 ± 0.00	0.03 ± 0.00	0.61 ± 0.04	1.00 ± 0.00	0.74 ± 0.00
DPL + TO	0.85 ± 0.02	1.00 ± 0.00	0.03 ± 0.00	0.32 ± 0.01	2.67 ± 0.07	0.94 ± 0.00
DPL + TAb	0.71 ± 0.02	397.39 ± 5.61	0.61 ± 0.02	0.41 ± 0.01	2.67 ± 0.07	0.94 ± 0.00
DPL + CO	0.36 ± 0.03	187.49 ± 48.31	0.92 ± 0.01	0.99 ± 0.00	1.00 ± 0.00	0.74 ± 0.00
DPL + CDe	0.99 ± 0.00	187.49 ± 48.31	0.92 ± 0.01	0.98 ± 0.00	1.94 ± 0.02	0.91 ± 0.01
DPL + RPB	0.79 ± 0.04	187.49 ± 48.31	0.92 ± 0.01	1.00 ± 0.00	1.50 ± 0.03	0.86 ± 0.01
DPL + COCOCO	1.00 ± 0.00	144.35 ± 34.56	0.87 ± 0.01	1.00 ± 0.00	1.50 ± 0.03	0.86 ± 0.01

Table 37: Results for DPL (with concept supervision) on BDD-OIA on Left Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	0.67 ± 0.01	1.00 ± 0.00	0.21 ± 0.02	0.67 ± 0.01	1.00 ± 0.00	0.74 ± 0.00
DPL + TO	0.85 ± 0.01	1.00 ± 0.00	0.21 ± 0.02	0.60 ± 0.01	1.41 ± 0.03	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	52.61 ± 0.76	0.70 ± 0.02	1.00 ± 0.00	1.41 ± 0.03	0.90 ± 0.01
DPL + CO	0.64 ± 0.04	26.55 ± 7.91	0.92 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.74 ± 0.00
DPL + CDe	1.00 ± 0.00	26.55 ± 7.91	0.92 ± 0.01	1.00 ± 0.00	1.93 ± 0.02	0.98 ± 0.01
DPL + RPB	0.82 ± 0.03	26.55 ± 7.91	0.92 ± 0.01	1.00 ± 0.00	1.36 ± 0.03	0.86 ± 0.01
DPL + COCOCO	1.00 ± 0.00	21.95 ± 7.37	0.64 ± 0.02	1.00 ± 0.00	1.36 ± 0.03	0.86 ± 0.01

Table 38: Results for DPL (with concept supervision) on BDD-OIA on Right Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
DPL	0.32 ± 0.03	1.00 ± 0.00	0.19 ± 0.02	0.32 ± 0.03	1.00 ± 0.00	0.73 ± 0.00
DPL + TO	0.58 ± 0.03	1.00 ± 0.00	0.19 ± 0.02	0.41 ± 0.02	1.44 ± 0.03	0.90 ± 0.01
DPL + TAb	1.00 ± 0.00	52.92 ± 0.92	0.85 ± 0.02	1.00 ± 0.00	1.44 ± 0.03	0.90 ± 0.01
DPL + CO	0.61 ± 0.02	28.23 ± 8.76	0.91 ± 0.01	0.97 ± 0.01	1.00 ± 0.00	0.73 ± 0.00
DPL + CDe	1.00 ± 0.00	28.23 ± 8.76	0.91 ± 0.01	1.00 ± 0.00	1.95 ± 0.01	0.97 ± 0.01
DPL + RPB	0.84 ± 0.02	28.23 ± 8.76	0.91 ± 0.01	1.00 ± 0.00	1.38 ± 0.03	0.86 ± 0.01
DPL + COCOCO	1.00 ± 0.00	23.64 ± 7.69	0.77 ± 0.01	1.00 ± 0.00	1.38 ± 0.03	0.86 ± 0.01

Table 39: Results for LTN (without concept supervision) on BDD-OIA on Forward-Stop Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN †	0.00 ± 0.00	1.00 ± 0.00	0.02 ± 0.01	0.00 ± 0.00	1.00 ± 0.00	0.46 ± 0.04
LTN + TO †	1.00 ± 0.00	1.00 ± 0.00	0.02 ± 0.01	0.25 ± 0.00	4.00 ± 0.00	1.00 ± 0.00
LTN + TAb †	1.00 ± 0.00	512.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	4.00 ± 0.00	1.00 ± 0.00
LTN + CO †	0.47 ± 0.22	73.92 ± 8.94	0.90 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.46 ± 0.04
LTN + CDe †	1.00 ± 0.00	73.92 ± 8.94	0.90 ± 0.01	0.99 ± 0.00	1.95 ± 0.01	0.92 ± 0.01
LTN + RPB †	1.00 ± 0.00	73.92 ± 8.94	0.90 ± 0.01	1.00 ± 0.00	1.95 ± 0.01	0.92 ± 0.01
LTN + COCOCO †	1.00 ± 0.00	73.92 ± 8.94	0.90 ± 0.01	1.00 ± 0.00	1.95 ± 0.01	0.92 ± 0.01

Table 40: Results for LTN (without concept supervision) on BDD-OIA on Left Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN †	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.28 ± 0.01
LTN + TO †	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.01	0.50 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + TAb †	1.00 ± 0.00	64.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + CO †	0.57 ± 0.10	12.01 ± 2.42	0.91 ± 0.01	0.99 ± 0.00	1.00 ± 0.00	0.28 ± 0.01
LTN + CDe †	1.00 ± 0.00	12.01 ± 2.42	0.91 ± 0.01	1.00 ± 0.00	1.92 ± 0.02	0.95 ± 0.01
LTN + RPB †	1.00 ± 0.00	12.01 ± 2.42	0.91 ± 0.01	1.00 ± 0.00	1.92 ± 0.02	0.95 ± 0.01
LTN + COCOCO †	1.00 ± 0.00	12.01 ± 2.42	0.91 ± 0.01	1.00 ± 0.00	1.92 ± 0.02	0.95 ± 0.01

Table 41: Results for LTN (without concept supervision) on BDD-OIA on Right Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN †	1.00 ± 0.00	1.00 ± 0.00	0.01 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.30 ± 0.01
LTN + TO †	1.00 ± 0.00	1.00 ± 0.00	0.01 ± 0.01	0.50 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + TAb †	1.00 ± 0.00	64.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + CO †	0.14 ± 0.01	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.30 ± 0.01
LTN + CDe †	1.00 ± 0.00	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.96 ± 0.01	0.98 ± 0.01
LTN + RPB †	1.00 ± 0.00	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.96 ± 0.01	0.98 ± 0.01
LTN + COCOCO †	1.00 ± 0.00	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.96 ± 0.01	0.98 ± 0.01

Table 42: Results for LTN (with concept supervision) on BDD-OIA on Forward-Stop Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	0.00 ± 0.00	1.00 ± 0.00	0.04 ± 0.02	0.00 ± 0.00	1.00 ± 0.00	0.46 ± 0.04
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.04 ± 0.02	0.25 ± 0.00	4.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	512.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	4.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.48 ± 0.22	71.15 ± 8.81	0.91 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.46 ± 0.04
LTN + CDe	1.00 ± 0.00	71.15 ± 8.81	0.91 ± 0.01	0.99 ± 0.00	1.95 ± 0.01	0.92 ± 0.01
LTN + RPB	1.00 ± 0.00	71.15 ± 8.81	0.91 ± 0.01	1.00 ± 0.00	1.95 ± 0.01	0.92 ± 0.01
LTN + COCOCO	1.00 ± 0.00	71.15 ± 8.81	0.91 ± 0.01	1.00 ± 0.00	1.95 ± 0.01	0.92 ± 0.01

Table 43: Results for LTN (with concept supervision) on BDD-OIA on Left Action.

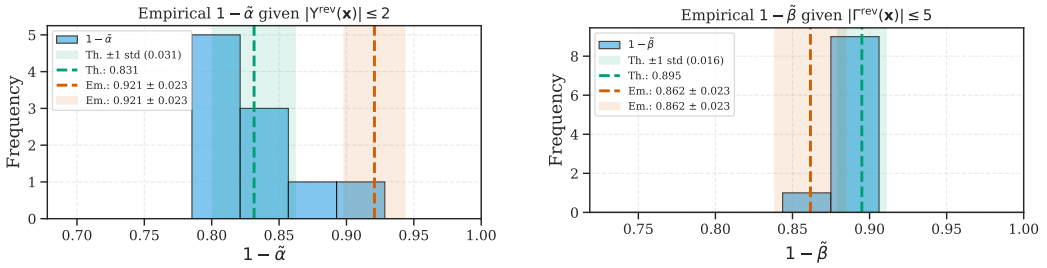
	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.28 ± 0.01
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.03 ± 0.01	0.50 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	64.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.57 ± 0.10	12.01 ± 2.42	0.91 ± 0.01	0.99 ± 0.00	1.00 ± 0.00	0.28 ± 0.01
LTN + CDe	1.00 ± 0.00	12.01 ± 2.42	0.91 ± 0.01	1.00 ± 0.00	1.92 ± 0.02	0.95 ± 0.01
LTN + RPB	1.00 ± 0.00	12.01 ± 2.42	0.91 ± 0.01	1.00 ± 0.00	1.92 ± 0.02	0.95 ± 0.01
LTN + COCOCO	1.00 ± 0.00	12.01 ± 2.42	0.91 ± 0.01	1.00 ± 0.00	1.92 ± 0.02	0.95 ± 0.01

Table 44: Results for LTN (with concept supervision) on BDD-OIA on Right Action.

	Concepts			Labels		
Method	Const.	Size	Cov.	Const.	Size	Cov.
LTN	1.00 ± 0.00	1.00 ± 0.00	0.01 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.30 ± 0.01
LTN + TO	1.00 ± 0.00	1.00 ± 0.00	0.01 ± 0.01	0.50 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + TAb	1.00 ± 0.00	64.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	1.00 ± 0.00
LTN + CO	0.14 ± 0.01	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.30 ± 0.01
LTN + CDe	1.00 ± 0.00	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.96 ± 0.01	0.98 ± 0.01
LTN + RPB	1.00 ± 0.00	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.96 ± 0.01	0.98 ± 0.01
LTN + COCOCO	1.00 ± 0.00	53.50 ± 1.96	0.90 ± 0.01	1.00 ± 0.00	1.96 ± 0.01	0.98 ± 0.01

Table 45: E-prediction theoretical coverage target ($1 - \mathbb{E}[\alpha]$, $1 - \mathbb{E}[\beta]$, respectively) results on CHX, averaged over 10 seeds (mean $\pm$ std). Const. is the imposed set-size constraint (– if unconstrained); Size is the average prediction-set size; Cov. is the empirical coverage.

	Concepts				Labels			
Method	Const.	Size	$1 - \mathbb{E}[\tilde{\beta}]$	Cov.	Const.	Size	$1 - \mathbb{E}[\tilde{\alpha}]$	Cov.
Concept-fixed	5	3.37 ± 0.30	0.90 ± 0.00	0.86 ± 0.02	—	1.43 ± 0.08	—	—
Label-fixed	—	2.42 ± 0.45	—	—	2	1.72 ± 0.12	0.82 ± 0.04	0.90 ± 0.02
Both-fixed	5	3.57 ± 0.73	0.77 ± 0.13	0.89 ± 0.03	2	1.80 ± 0.06	0.82 ± 0.04	0.91 ± 0.01



(a) Label-fixed: $|\Upsilon^{\text{rev}}(\mathbf{x})| \leq 2$

(b) Concept-fixed: $|\Gamma^{\text{rev}}(\mathbf{x})| \leq 5$

Figure 3: E-conformal coverage under single-sided constraints on CHX averaged over 10 seeds. Empirical coverage closely tracks the theoretical guarantee in both regimes.

Table 46: Full results over the 9 datasets for DPL. □ indicates no concept supervision.

		Concepts			Labels		
	Method	Const.	Size	Cov.	Const.	Size	Cov.
MN-Add	DPL □	**1.00 ± 0.00**	1.00 ± 0.00	0.98 ± 0.00	**1.00 ± 0.00**	1.00 ± 0.00	0.98 ± 0.00
	DPL + RPB □	1.00 ± 0.00	1.00 ± 0.00	**0.90 ± 0.01**	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**0.99 ± 0.00**	**0.90 ± 0.01**	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**
MN-H	DPL □	**1.00 ± 0.00**	1.00 ± 0.00	0.50 ± 0.03	**1.00 ± 0.00**	1.00 ± 0.00	0.99 ± 0.01
	DPL + RPB □	0.09 ± 0.01	25.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.89 ± 0.05**	**0.89 ± 0.05**
	DPL + COCOCO □	**1.00 ± 0.00**	**2.20 ± 0.14**	0.89 ± 0.05	**1.00 ± 0.00**	**0.89 ± 0.05**	**0.89 ± 0.05**
MN-SP	DPL □	**1.00 ± 0.00**	1.00 ± 0.00	0.03 ± 0.02	**1.00 ± 0.00**	1.00 ± 0.00	0.99 ± 0.00
	DPL + RPB □	0.73 ± 0.19	46.81 ± 36.72	**0.93 ± 0.05**	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**27.92 ± 12.05**	0.84 ± 0.05	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**
MN-EO	DPL □	**1.00 ± 0.00**	1.00 ± 0.00	0.13 ± 0.08	**1.00 ± 0.00**	1.00 ± 0.00	0.72 ± 0.01
	DPL + RPB □	0.12 ± 0.01	100.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.55 ± 0.18**	**0.90 ± 0.02**
	DPL + COCOCO □	**1.00 ± 0.00**	**12.13 ± 1.37**	0.90 ± 0.02	**1.00 ± 0.00**	**1.55 ± 0.18**	**0.90 ± 0.02**
CHX	DPL	**1.00 ± 0.00**	1.00 ± 0.00	0.47 ± 0.11	**1.00 ± 0.00**	1.00 ± 0.00	0.72 ± 0.01
	DPL + RPB	0.55 ± 0.05	5.30 ± 1.01	**0.91 ± 0.03**	**1.00 ± 0.00**	**1.60 ± 0.18**	**0.88 ± 0.03**
	DPL + COCOCO	**1.00 ± 0.00**	**2.90 ± 0.56**	0.83 ± 0.04	**1.00 ± 0.00**	**1.60 ± 0.18**	**0.88 ± 0.03**
	DPL □	**1.00 ± 0.00**	1.00 ± 0.00	0.45 ± 0.09	**1.00 ± 0.00**	1.00 ± 0.00	0.73 ± 0.01
	DPL + RPB □	0.52 ± 0.12	6.85 ± 2.53	**0.91 ± 0.02**	**1.00 ± 0.00**	**1.66 ± 0.22**	**0.90 ± 0.04**
	DPL + COCOCO □	**1.00 ± 0.00**	**3.36 ± 0.80**	0.84 ± 0.04	**1.00 ± 0.00**	**1.66 ± 0.22**	**0.90 ± 0.04**
BDD-OIA-FS	DPL	0.61 ± 0.04	1.00 ± 0.00	0.03 ± 0.00	0.61 ± 0.04	1.00 ± 0.00	0.74 ± 0.00
	DPL + RPB	0.79 ± 0.04	187.49 ± 48.31	**0.92 ± 0.01**	**1.00 ± 0.00**	**1.50 ± 0.03**	**0.86 ± 0.01**
	DPL + COCOCO	**1.00 ± 0.00**	**144.35 ± 34.56**	0.87 ± 0.01	**1.00 ± 0.00**	**1.50 ± 0.03**	**0.86 ± 0.01**
	DPL □	0.61 ± 0.04	1.00 ± 0.00	0.04 ± 0.00	0.61 ± 0.04	1.00 ± 0.00	0.74 ± 0.00
	DPL + RPB □	0.80 ± 0.04	192.39 ± 49.91	**0.92 ± 0.01**	**1.00 ± 0.00**	**1.49 ± 0.03**	**0.86 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**149.99 ± 35.63**	0.87 ± 0.01	**1.00 ± 0.00**	**1.49 ± 0.03**	**0.86 ± 0.01**
BDD-OIA-L	DPL	0.67 ± 0.01	1.00 ± 0.00	0.21 ± 0.02	0.67 ± 0.01	1.00 ± 0.00	0.74 ± 0.00
	DPL + RPB	0.82 ± 0.03	26.55 ± 7.91	**0.92 ± 0.01**	**1.00 ± 0.00**	**1.36 ± 0.03**	**0.86 ± 0.01**
	DPL + COCOCO	**1.00 ± 0.00**	**21.95 ± 7.37**	0.64 ± 0.02	**1.00 ± 0.00**	**1.36 ± 0.03**	**0.86 ± 0.01**
	DPL □	0.67 ± 0.01	1.00 ± 0.00	0.21 ± 0.02	0.67 ± 0.01	1.00 ± 0.00	0.74 ± 0.00
	DPL + RPB □	0.82 ± 0.03	25.98 ± 7.55	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.35 ± 0.03**	**0.86 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**21.41 ± 7.02**	0.64 ± 0.02	**1.00 ± 0.00**	**1.35 ± 0.03**	**0.86 ± 0.01**
BDD-OIA-R	DPL	0.32 ± 0.03	1.00 ± 0.00	0.19 ± 0.02	0.32 ± 0.03	1.00 ± 0.00	0.73 ± 0.00
	DPL + RPB	0.84 ± 0.02	28.23 ± 8.76	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.38 ± 0.03**	**0.86 ± 0.01**
	DPL + COCOCO	**1.00 ± 0.00**	**23.64 ± 7.69**	0.77 ± 0.01	**1.00 ± 0.00**	**1.38 ± 0.03**	**0.86 ± 0.01**
	DPL □	0.32 ± 0.03	1.00 ± 0.00	0.19 ± 0.02	0.32 ± 0.03	1.00 ± 0.00	0.73 ± 0.00
	DPL + RPB □	0.84 ± 0.02	27.69 ± 8.18	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.38 ± 0.03**	**0.86 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**23.20 ± 7.25**	0.77 ± 0.01	**1.00 ± 0.00**	**1.38 ± 0.03**	**0.86 ± 0.01**
CIF10	DPL	1.00 ± 0.00	1.00 ± 0.00	0.34 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.66 ± 0.00
	DPL + RPB	0.91 ± 0.02	20.75 ± 2.33	**0.93 ± 0.01**	**1.00 ± 0.00**	**1.76 ± 0.06**	**0.90 ± 0.01**
	DPL + COCOCO	**1.00 ± 0.00**	**18.87 ± 2.12**	0.88 ± 0.01	**1.00 ± 0.00**	**1.76 ± 0.06**	**0.90 ± 0.01**
	DPL □	0.97 ± 0.05	1.00 ± 0.00	0.34 ± 0.02	0.97 ± 0.05	1.00 ± 0.00	0.66 ± 0.00
	DPL + RPB □	0.79 ± 0.05	57.60 ± 24.96	**0.97 ± 0.01**	**1.00 ± 0.00**	**1.83 ± 0.08**	**0.90 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**45.43 ± 19.68**	0.90 ± 0.01	**1.00 ± 0.00**	**1.83 ± 0.08**	**0.90 ± 0.01**
RIV10	DPL	1.00 ± 0.00	1.00 ± 0.00	0.43 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.80 ± 0.00
	DPL + RPB	0.94 ± 0.01	6.33 ± 0.65	**0.92 ± 0.01**	**1.00 ± 0.00**	**1.20 ± 0.02**	**0.89 ± 0.01**
	DPL + COCOCO	**1.00 ± 0.00**	**5.85 ± 0.62**	0.84 ± 0.01	**1.00 ± 0.00**	**1.20 ± 0.02**	**0.89 ± 0.01**
	DPL □	1.00 ± 0.00	1.00 ± 0.00	0.41 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	0.78 ± 0.00
	DPL + RPB □	0.92 ± 0.03	8.11 ± 2.22	**0.93 ± 0.01**	**1.00 ± 0.00**	**1.23 ± 0.02**	**0.90 ± 0.01**
	DPL + COCOCO □	**1.00 ± 0.00**	**7.31 ± 1.77**	0.86 ± 0.01	**1.00 ± 0.00**	**1.23 ± 0.02**	**0.90 ± 0.01**
CEB	DPL	**1.00 ± 0.00**	1.00 ± 0.00	0.18 ± 0.02	**1.00 ± 0.00**	1.00 ± 0.00	0.58 ± 0.03
	DPL + RPB	0.78 ± 0.06	75.06 ± 38.35	**0.90 ± 0.04**	**1.00 ± 0.00**	**2.31 ± 0.16**	**0.87 ± 0.01**
	DPL + COCOCO	**1.00 ± 0.00**	**53.49 ± 25.93**	0.83 ± 0.03	**1.00 ± 0.00**	**2.31 ± 0.16**	**0.87 ± 0.01**
	DPL □	**1.00 ± 0.00**	1.00 ± 0.00	0.00 ± 0.00	**1.00 ± 0.00**	1.00 ± 0.00	0.61 ± 0.01
	DPL + RPB □	0.74 ± 0.27	907.88 ± 76.19	**0.97 ± 0.02**	**1.00 ± 0.00**	**3.59 ± 1.49**	**0.94 ± 0.06**
	DPL + COCOCO □	**1.00 ± 0.00**	**670.15 ± 252.46**	0.92 ± 0.06	**1.00 ± 0.00**	**3.59 ± 1.49**	**0.94 ± 0.06**

Table 47: Full results over the 9 datasets for LTN. □ indicates no concept supervision.

	Method	Concepts			Labels		
		Const.	Size	Cov.	Const.	Size	Cov.
MN-Add	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.97 ± 0.00	**1.00 ± 0.00**	1.00 ± 0.00	0.97 ± 0.00
	LTN + RPB □	**1.00 ± 0.00**	1.00 ± 0.00	**0.90 ± 0.01**	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**	**1.00 ± 0.00**	**0.90 ± 0.01**	**0.90 ± 0.01**
MN-H	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.50 ± 0.03	**1.00 ± 0.00**	1.00 ± 0.00	0.83 ± 0.12
	LTN + RPB □	0.64 ± 0.46	19.48 ± 1.70	**0.87 ± 0.04**	**1.00 ± 0.00**	**5.60 ± 3.97**	**0.99 ± 0.02**
	LTN + COCOCO □	**1.00 ± 0.00**	**12.77 ± 9.35**	0.86 ± 0.04	**1.00 ± 0.00**	**5.60 ± 3.97**	**0.99 ± 0.02**
MN-SP	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.03 ± 0.02	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.98 ± 0.00**
	LTN + RPB □	0.50 ± 0.00	100.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.98 ± 0.00**
	LTN + COCOCO □	**1.00 ± 0.00**	**50.00 ± 0.00**	0.98 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.98 ± 0.00**
MN-EO	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.94 ± 0.01**
	LTN + RPB □	0.08 ± 0.00	100.00 ± 0.00	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.94 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**8.05 ± 0.05**	0.94 ± 0.01	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.94 ± 0.01**
CHX	LTN	**1.00 ± 0.00**	1.00 ± 0.00	0.46 ± 0.09	**1.00 ± 0.00**	1.00 ± 0.00	0.74 ± 0.02
	LTN + RPB	**1.00 ± 0.00**	6.58 ± 1.64	**0.92 ± 0.01**	**1.00 ± 0.00**	**3.53 ± 0.35**	**0.99 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**6.56 ± 1.66**	**0.92 ± 0.01**	**1.00 ± 0.00**	**3.53 ± 0.35**	**0.99 ± 0.01**
	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.45 ± 0.10	**1.00 ± 0.00**	1.00 ± 0.00	0.73 ± 0.02
	LTN + RPB □	**1.00 ± 0.00**	7.58 ± 2.49	**0.91 ± 0.02**	**1.00 ± 0.00**	**3.68 ± 0.44**	**0.98 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**7.54 ± 2.51**	**0.91 ± 0.02**	**1.00 ± 0.00**	**3.68 ± 0.44**	**0.98 ± 0.01**
BDD-OIA-FS	LTN	0.00 ± 0.00	1.00 ± 0.00	0.04 ± 0.02	0.00 ± 0.00	1.00 ± 0.00	0.46 ± 0.04
	LTN + RPB	**1.00 ± 0.00**	**71.15 ± 8.81**	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.95 ± 0.01**	**0.92 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**71.15 ± 8.81**	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.95 ± 0.01**	**0.92 ± 0.01**
	LTN □	0.00 ± 0.00	1.00 ± 0.00	0.02 ± 0.01	0.00 ± 0.00	1.00 ± 0.00	0.46 ± 0.04
	LTN + RPB □	**1.00 ± 0.00**	**73.92 ± 8.94**	**0.90 ± 0.01**	**1.00 ± 0.00**	**1.95 ± 0.01**	**0.92 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**73.92 ± 8.94**	**0.90 ± 0.01**	**1.00 ± 0.00**	**1.95 ± 0.01**	**0.92 ± 0.01**
BDD-OIA-L	LTN	**1.00 ± 0.00**	1.00 ± 0.00	0.03 ± 0.01	**1.00 ± 0.00**	1.00 ± 0.00	0.28 ± 0.01
	LTN + RPB	**1.00 ± 0.00**	**12.01 ± 2.42**	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.92 ± 0.02**	**0.95 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**12.01 ± 2.42**	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.92 ± 0.02**	**0.95 ± 0.01**
	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.03 ± 0.01	**1.00 ± 0.00**	1.00 ± 0.00	0.28 ± 0.01
	LTN + RPB □	**1.00 ± 0.00**	**12.01 ± 2.42**	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.92 ± 0.02**	**0.95 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**12.01 ± 2.42**	**0.91 ± 0.01**	**1.00 ± 0.00**	**1.92 ± 0.02**	**0.95 ± 0.01**
BDD-OIA-R	LTN	**1.00 ± 0.00**	1.00 ± 0.00	0.01 ± 0.01	**1.00 ± 0.00**	1.00 ± 0.00	0.30 ± 0.01
	LTN + RPB	**1.00 ± 0.00**	**53.50 ± 1.96**	**0.90 ± 0.01**	**1.00 ± 0.00**	**1.96 ± 0.01**	**0.98 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**53.50 ± 1.96**	**0.90 ± 0.01**	**1.00 ± 0.00**	**1.96 ± 0.01**	**0.98 ± 0.01**
	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.01 ± 0.01	**1.00 ± 0.00**	1.00 ± 0.00	0.30 ± 0.01
	LTN + RPB □	**1.00 ± 0.00**	**53.50 ± 1.96**	**0.90 ± 0.01**	**1.00 ± 0.00**	**1.96 ± 0.01**	**0.98 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**53.50 ± 1.96**	**0.90 ± 0.01**	**1.00 ± 0.00**	**1.96 ± 0.01**	**0.98 ± 0.01**
CIF10	LTN	0.98 ± 0.00	1.00 ± 0.00	0.31 ± 0.01	0.98 ± 0.00	1.00 ± 0.00	0.58 ± 0.01
	LTN + RPB	0.93 ± 0.02	19.65 ± 2.28	**0.93 ± 0.01**	**1.00 ± 0.00**	**2.06 ± 0.14**	**0.89 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**18.22 ± 1.79**	0.88 ± 0.01	**1.00 ± 0.00**	**2.06 ± 0.14**	**0.89 ± 0.01**
	LTN □	0.98 ± 0.00	1.00 ± 0.00	0.13 ± 0.04	0.98 ± 0.00	1.00 ± 0.00	0.50 ± 0.03
	LTN + RPB □	0.94 ± 0.03	28.83 ± 3.66	**0.93 ± 0.01**	**1.00 ± 0.00**	**3.08 ± 0.17**	**0.90 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**27.09 ± 2.77**	0.88 ± 0.01	**1.00 ± 0.00**	**3.08 ± 0.17**	**0.90 ± 0.01**
RIV10	LTN	0.99 ± 0.00	1.00 ± 0.00	0.41 ± 0.02	0.99 ± 0.00	1.00 ± 0.00	0.72 ± 0.01
	LTN + RPB	0.96 ± 0.01	9.41 ± 0.72	**0.92 ± 0.01**	**1.00 ± 0.00**	**1.33 ± 0.02**	**0.89 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**8.85 ± 0.68**	0.85 ± 0.01	**1.00 ± 0.00**	**1.33 ± 0.02**	**0.89 ± 0.01**
	LTN □	0.99 ± 0.02	1.00 ± 0.00	0.14 ± 0.11	0.99 ± 0.02	1.00 ± 0.00	0.56 ± 0.12
	LTN + RPB □	0.99 ± 0.00	12.00 ± 1.78	**0.92 ± 0.01**	**1.00 ± 0.00**	**2.05 ± 0.30**	**0.89 ± 0.02**
	LTN + COCOCO □	**1.00 ± 0.00**	**11.65 ± 1.74**	0.85 ± 0.01	**1.00 ± 0.00**	**2.05 ± 0.30**	**0.89 ± 0.02**
CEB	LTN	**1.00 ± 0.00**	1.00 ± 0.00	0.24 ± 0.01	**1.00 ± 0.00**	1.00 ± 0.00	0.63 ± 0.01
	LTN + RPB	**1.00 ± 0.00**	36.19 ± 4.07	**0.90 ± 0.02**	**1.00 ± 0.00**	**3.13 ± 0.13**	**0.96 ± 0.01**
	LTN + COCOCO	**1.00 ± 0.00**	**36.05 ± 4.08**	**0.90 ± 0.02**	**1.00 ± 0.00**	**3.13 ± 0.13**	**0.96 ± 0.01**
	LTN □	**1.00 ± 0.00**	1.00 ± 0.00	0.00 ± 0.00	**1.00 ± 0.00**	1.00 ± 0.00	0.62 ± 0.01
	LTN + RPB □	**1.00 ± 0.00**	**349.70 ± 34.12**	**0.91 ± 0.02**	**1.00 ± 0.00**	**4.31 ± 0.18**	**0.99 ± 0.01**
	LTN + COCOCO □	**1.00 ± 0.00**	**349.70 ± 34.12**	**0.91 ± 0.02**	**1.00 ± 0.00**	**4.31 ± 0.18**	**0.99 ± 0.01**

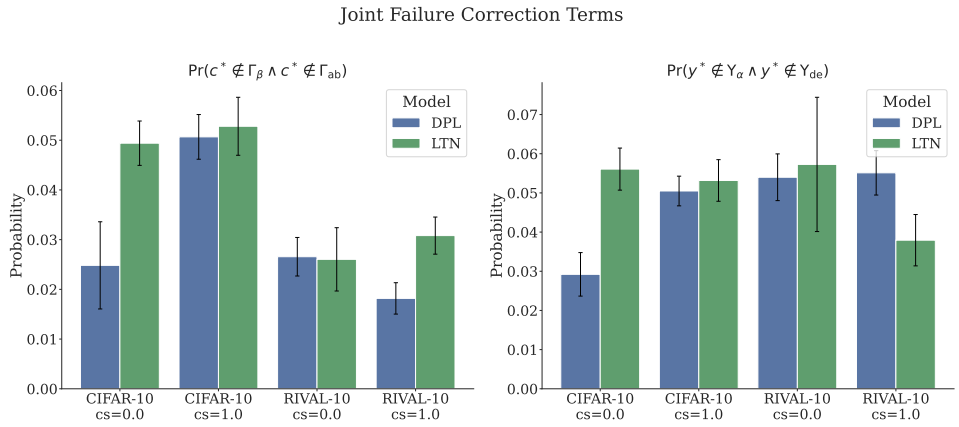


Figure 4: Joint failure probabilities on `CIF10` and `RIV10`. Even if low, this estimate contributes to tightening the lower bound in Proposition 4.2 and Proposition B.10.

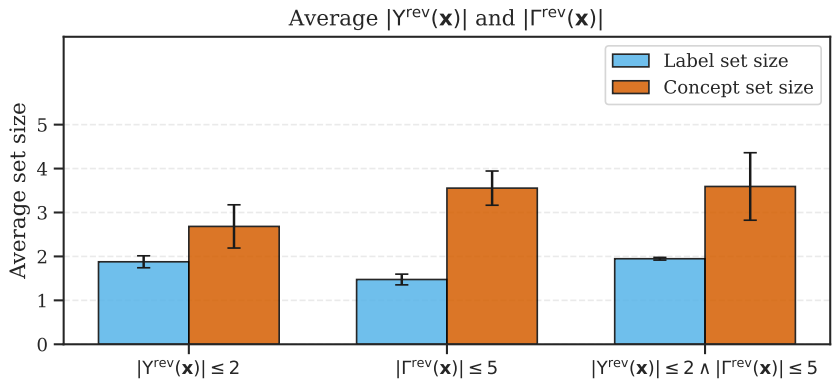


Figure 5: Average prediction-set sizes across constraint regimes; error bars show ± 1 std across 10 seeds.

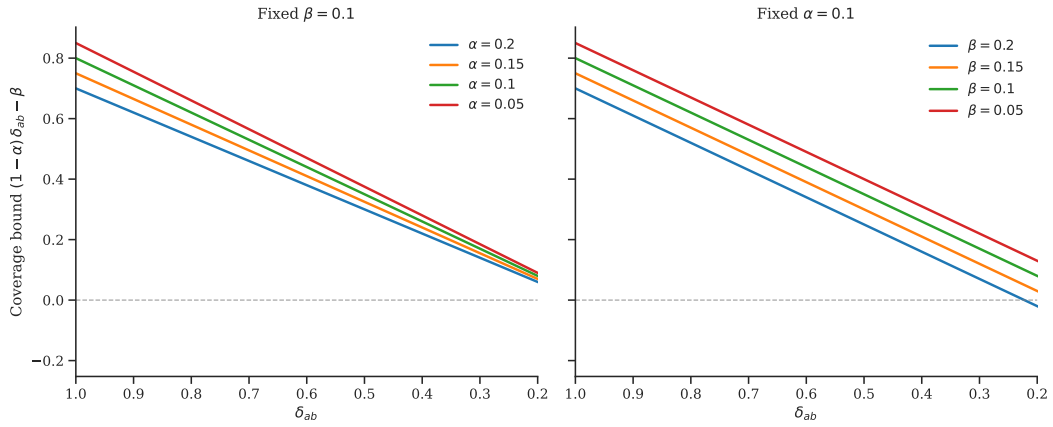


Figure 6: Synthetic experiment showing $(1-\alpha)\delta_{ab} - \beta$ for varying α (left) and β (right) starting at nominal guarantee of $1-\alpha = 1-\beta = 0.9$. The bound decrease linearly depending on the quality of the inference layer g.

E.1 Scaling to BDD-OIA

BDD-OIA [118, 94] has 21 binary concepts, so $\mathbf{\Gamma}^{\times}(\mathbf{x})$ can contain up to 2^{21} worlds, making naive enumeration infeasible. Following Bortolotti et al. [94], we exploit the structure of K to decompose the multi-label task into three independent sub-tasks: forward/stop (F-S), left (L), and right (R), which depend on only 9, 6, and 6 concepts, respectively. This reduces the number of worlds to enumerate to $2^9 = 512$, $2^6 = 64$, and $2^6 = 64$ for the three sub-tasks. Table 48 and Table 49 reports DPL results with $\alpha = \beta = 0.1$, averaged over 10 seeds.

Table 48: Concept-level results for DPL on BDD-OIA, per action. C-SUP: concept supervision during training.

		F-S			L			R		
METHOD	C-SUP	COV.	SIZE	CONST.	COV.	SIZE	CONST.	COV.	SIZE	CONST.
RPB	✓	0.92 ± 0.01	187.5 ± 47.8	0.79 ± 0.04	0.92 ± 0.01	26.5 ± 7.5	0.82 ± 0.03	0.91 ± 0.01	28.2 ± 8.3	0.84 ± 0.01
COCOCO	✓	0.87 ± 0.01	144.4 ± 32.8	1.00	0.64 ± 0.02	21.9 ± 7.0	1.00	0.77 ± 0.01	23.6 ± 7.3	1.00
RPB	✗	0.92 ± 0.01	192.4 ± 47.3	0.80 ± 0.04	0.91 ± 0.01	26.0 ± 7.2	0.82 ± 0.03	0.91 ± 0.01	27.7 ± 7.8	0.84 ± 0.01
COCOCO	✗	0.87 ± 0.01	150.0 ± 33.8	1.00	0.64 ± 0.02	21.4 ± 6.7	1.00	0.77 ± 0.01	23.2 ± 6.9	1.00

Table 49: Label-level results for DPL on BDD-OIA, per action. Label sets of COCOCO and RPB coincide by construction. C-SUP: concept supervision during training.

		F-S			L			R		
METHOD	C-SUP	COV.	SIZE	CONST.	COV.	SIZE	CONST.	COV.	SIZE	CONST.
COCOCO/ RPB	✓	0.86 ± 0.01	1.5 ± 0.0	1.00	0.86 ± 0.01	1.4 ± 0.0	1.00	0.86 ± 0.01	1.4 ± 0.0	1.00
COCOCO/ RPB	✗	0.86 ± 0.01	1.5 ± 0.0	1.00	0.86 ± 0.01	1.4 ± 0.0	1.00	0.86 ± 0.01	1.4 ± 0.0	1.00

COCOCO is the only approach achieving perfect concept consistency (1.00, compared with 0.79–0.84 for RPB), while consistently producing smaller concept sets (e.g., $187.5 \to 144.4$ for F-S with concept supervision). The lower concept coverage observed on L and R is predicted by Proposition 4.2, as the knowledge encoded by BDD-OIA is not abductively sound [94]. Specifically, with $\hat{\delta}_{\text{ab}}^{\text{L}} = 0.728$ and $\hat{\delta}_{\text{ab}}^{\text{R}} = 0.906$, and estimated joint-failure terms of 0.026 and 0.008, respectively, the bound gives coverage guarantees of 0.58 and 0.72. Both are satisfied by the corresponding empirical coverages of 0.64 and 0.77.

E.2 Estimating δ_{ab} and δ_{de}

We estimate the soundness gaps empirically as

$$\hat{\delta}_{\text{de}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x},\mathbf{c}^*,\mathbf{y}^*)\in\mathcal{D}_{\text{test}}} \mathbb{1}\{g^{\dagger}(\mathbf{c}^*) = \mathbf{y}^*\}, \tag{141}$$

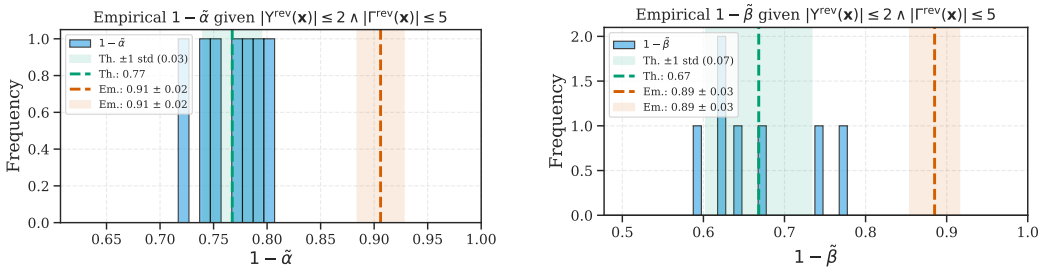
$$\hat{\delta}_{\text{ab}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x},\mathbf{c}^*,\mathbf{y}^*)\in\mathcal{D}_{\text{test}}} \mathbb{1}\{\mathbf{c}^* \in g^{-\dagger}(\mathbf{y}^*)\}. \tag{142}$$

where $\mathcal{D}_{\text{test}}$ is the test set. RIV10 and CIF10 are constructed such that 4 of 10 classes share concepts with another class, so $\delta_{\text{de}} = 0.8$ is a structural property of the dataset by design.

Table 50: Empirical estimates of δ_{ab} and δ_{de} on the test set, averaged over 10 seeds (mean $\pm$ std).

DATASET	$\hat{\delta}_{\text{ab}}$	$\hat{\delta}_{\text{de}}$
MN-Add	1.00 ± 0.00	1.00 ± 0.00
MN-H	1.00 ± 0.00	1.00 ± 0.00
MN-SP	1.00 ± 0.00	1.00 ± 0.00
MN-EO	1.00 ± 0.00	1.00 ± 0.00
CHX	1.00 ± 0.00	1.00 ± 0.00
CEB	1.00 ± 0.00	1.00 ± 0.00
CIF10	1.00 ± 0.00	0.79 ± 0.01
RIV10	1.00 ± 0.00	0.79 ± 0.01

E.3 Product vs. Average e-value Aggregation: when Assumptions do not hold



(a) Label coverage

(b) Concept coverage

Figure 7: COCOCO* on CHX with DPL under average size budgets $|\Upsilon^{\text{rev}}(\mathbf{x})| \leq 2$ and $|\Gamma^{\text{rev}}(\mathbf{x})| \leq 5$, bootstrapped over 100 calibration resamples and averaged across 10 seeds. Empirical coverage (orange) exceeds the theoretical target (green), while sizes remain *strictly* below the imposed budgets. Results using **product** as e-value aggregator.

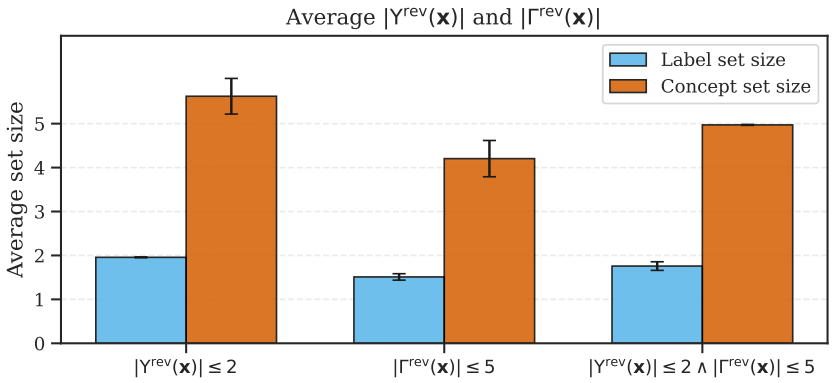


Figure 8: Average set-sizes using COCOCO* and **product** as e-value aggregator.

As an additional ablation, we examine what happens when there is data leakage in the calibration process, as well as when we employ the *product* as an e-value aggregator. Specifically, we used the entire test set for evaluation. In this case, leakage occurs because the model is selected via early stopping on the validation set and is therefore influenced by it, which violates the *exchangeability* assumption.

We evaluate COCOCO* on CHX, fixing the label and concept budgets to 2 and 5, respectively, and using **product** as the e-value aggregator. This approach is theoretically justified only under the assumption of *conditional independence* of the per-concept e-values given $(\mathbf{x}, \mathcal{D}_{\text{cal}})$, an assumption that does not hold in practice. For this analysis, we performed an extensive search over the space of α and β – the label and concept miscoverage rates, respectively, and ran 100 bootstrap iterations. Specifically, we analyze the following combinations: $(\{0.10, 0.11, \ldots, 0.30\}, \{0.10, 0.30, 0.31, \ldots, 0.60\})$. Surprisingly, despite these violations, COCOCO* exhibits stable behavior on the test set. As shown in Fig. 8, the average set sizes are 1.74 for labels and 4.97 for concepts, both strictly below their respective budgets. The empirical coverages are 0.91 and 0.89, substantially exceeding the theoretical targets $1 - \mathbb{E}[\tilde{\alpha}] = 0.77$ and $1 - \mathbb{E}[\tilde{\beta}] = 0.67$ (see Section E.3).

E.4 Splitting the Miscoverage Budget between Concepts and Labels

By setting $\alpha = \beta = 0.1$, Proposition 4.2 guarantees a nominal coverage of $1 - \alpha - \beta = 0.8$ for COCOCO, while RPB concept set has a nominal coverage of 0.9. To compare the methods under matched nominal coverage guarantees, we perform an additional ablation in which we fix $\alpha + \beta = 0.1$ and progressively shift the error budget from labels to concepts: $(\alpha, \beta) \in$

$(0.01, 0.09), (0.03, 0.07), (0.05, 0.05), (0.07, 0.03), (0.09, 0.01)$. Table 51 reports the results for DPL on CHX across 10 random seeds.

Table 51: COCOCO with DPL on CHX under different splits of the miscoverage budget 0.1. C-SUP: concept supervision during training. The first row of each block is the default setting of Table 2.

			CONCEPTS			LABELS		
C-SUP	(α, β)	NOMINAL	CONST.	SIZE	COV.	CONST.	SIZE	COV.
✓	$(0.10, 0.10)$	0.8	1.00	2.90 ± 0.56	0.83 ± 0.04	1.00	1.60 ± 0.18	0.88 ± 0.03
✓	$(0.01, 0.09)$	0.9	1.00	4.57 ± 0.86	0.91 ± 0.03	1.00	2.57 ± 0.35	0.97 ± 0.01
✓	$(0.03, 0.07)$	0.9	1.00	4.23 ± 0.59	0.91 ± 0.02	1.00	2.19 ± 0.12	0.95 ± 0.01
✓	$(0.05, 0.05)$	0.9	1.00	4.09 ± 0.55	0.90 ± 0.02	1.00	1.97 ± 0.14	0.94 ± 0.02
✓	$(0.07, 0.03)$	0.9	1.00	4.08 ± 0.57	0.89 ± 0.02	1.00	1.82 ± 0.14	0.92 ± 0.02
✓	$(0.09, 0.01)$	0.9	1.00	4.18 ± 0.79	0.89 ± 0.03	1.00	1.68 ± 0.17	0.90 ± 0.03
✗	$(0.10, 0.10)$	0.8	1.00	3.36 ± 0.80	0.84 ± 0.04	1.00	1.66 ± 0.22	0.90 ± 0.04
✗	$(0.01, 0.09)$	0.9	1.00	5.84 ± 2.31	0.91 ± 0.02	1.00	2.72 ± 0.57	0.98 ± 0.01
✗	$(0.03, 0.07)$	0.9	1.00	5.24 ± 1.38	0.91 ± 0.02	1.00	2.29 ± 0.19	0.96 ± 0.02
✗	$(0.05, 0.05)$	0.9	1.00	5.18 ± 1.23	0.91 ± 0.02	1.00	2.09 ± 0.20	0.95 ± 0.02
✗	$(0.07, 0.03)$	0.9	1.00	4.93 ± 1.07	0.90 ± 0.02	1.00	1.88 ± 0.18	0.93 ± 0.02
✗	$(0.09, 0.01)$	0.9	1.00	4.81 ± 0.93	0.89 ± 0.03	1.00	1.75 ± 0.20	0.91 ± 0.03

Across all configurations, consistency is perfect, and empirical coverage meets the corresponding nominal guarantee (**D1**–**D2**). As α increases from 0.01 to 0.09, the error budget shifts from labels to concepts: label sets shrink under concept supervision (from 2.57 to 1.68, compared with 1.60 for RPB), while concept sets remain relatively stable (from 4.57 to 4.18) and consistently smaller than those of RPB (5.30). Thus, at matched nominal guarantees, COCOCO achieves comparable concept coverage with smaller, fully consistent concept sets, while producing label sets comparable in size to RPB with a higher label-coverage guarantee (0.9 vs. 0.8). More generally, treating α and β as separate knobs gives users ex-ante control over where to allocate the error budget and, consequently, which side receives the stronger guarantee.

E.5 Size of the Concept-Annotated Calibration Set

Conformalizing concepts requires concept annotations, but *only on the calibration set*: the NeSy-CBM itself can be trained using label supervision alone. Table 52 evaluates COCOCO on CHX using models trained *without* concept supervision, with calibration sets of 43, 87, and 131 examples (5%, 10%, and 15% of the test set, respectively; the main experiments use 175), with $\alpha = \beta = 0.1$, averaged over 10 seeds.

Table 52: COCOCO on CHX (no concept supervision) for varying calibration-set sizes $|\mathcal{D}_{\text{cal}}|$.

		CONCEPTS		LABELS	
MODEL	$\lvert\mathcal{D}_{\text{cal}}\rvert$	COV.	SIZE	COV.	SIZE
DPL	43	0.80 ± 0.03	2.67 ± 0.70	0.86 ± 0.04	1.57 ± 0.20
DPL	87	0.82 ± 0.04	3.10 ± 0.70	0.88 ± 0.03	1.59 ± 0.20
DPL	131	0.85 ± 0.02	3.37 ± 0.70	0.90 ± 0.03	1.67 ± 0.20
LTN	43	0.84 ± 0.07	5.91 ± 2.99	0.95 ± 0.05	3.21 ± 0.65
LTN	87	0.88 ± 0.03	6.68 ± 2.71	0.98 ± 0.02	3.50 ± 0.54
LTN	131	0.90 ± 0.01	7.41 ± 2.36	0.98 ± 0.01	3.64 ± 0.42

As shown by Table 52, with as few as 43 concept-annotated examples, used only for calibration, COCOCO already attains coverage at or above the nominal $1 - \alpha - \beta = 0.8$ guarantee.

E.6 The Full Spectrum of NeSy Conformal Prediction

In the main text we compared COCOCO only against RPB, the strongest competitor. Here we report the full ablation against the four natural conformalization strategies introduced in Section A: TO, TAb, CO, and CDe. The goal is threefold: (i) confirm that the two-sided joint revision of COCOCO is genuinely necessary and that no simpler one-sided strategy achieves the same trade-off across **D1**–**D3**; (ii) quantify the failure modes of each baseline across the eight data sets; and (iii) ground the design choices of COCOCO empirically. The full numerical results are Tables 9 to 31; we summarise the qualitative results below. Throughout, the nominal coverage targets are $1 - \alpha = 1 - \beta = 0.9$.

TO **offers conformalized labels but uncalibrated, inconsistent concepts**. TO applies CP to the label distribution $p_\theta(\mathbf{Y} \mid \mathbf{x}, \mathsf{K})$ directly. As predicted by Eq. (4), label coverage hits the nominal $1-\alpha = 0.9$ target on every data set (e.g., 0.90 on MN-Add, 0.89 on CHX with DPL, 0.90 on RIV10, CEB, and CIF10). On the concept side, however, TO returns the single argmax concept (size 1), so concept coverage collapses to the base concept accuracy of the underlying NeSy-CBM: as low as 0.18 on CEB (DPL, supervised), 0.00 on CEB unsupervised (DPL and LTN), 0.00 and 0.13 on MN-EO (LTN and DPL, respectively) and 0.03 on MN-SP (DPL and LTN). Worse, label consistency drops well below 1 whenever the conformal label set is non-trivial: e.g., 0.20 on CHX (LTN), 0.20-0.42 on CEB, and 0.62 on CHX (DPL, supervised). TO thus violates both **D1** (consistency) and **D2** (concept-side coverage).

CO **achieves conformalized concepts but uncalibrated labels**. CO applies CP at the concept level using the product construction of Eq. (5) with Bonferroni's correction (Corollary B.4). Concept coverage matches the target on most data sets (0.87-1.00). However, CO returns the single argmax label, so the label side inherits the base label accuracy. Label consistency drops: 0.08-0.40 on MN-EO, MN-H, CEB, and CHX. Bonferroni's correction (Corollary B.4) inflates the concept set as the number of concepts grows: 75.06 on CEB (DPL, supervised), 57.60 on CIF10 (DPL, unsupervised), 100.00, the entire product space, on MN-SP (LTN) and MN-EO (LTN and DPL). CO thus partially satisfies **D2** on the concept side but fails both **D1** and **D3**.

TAb **achieves consistency and coverage but not conciseness**. TAb constructs a concept set via abduction, $\Gamma^{\mathsf{ab}}(\mathbf{x}) = g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$. This recovers both coverage and consistency at the concept level by construction: on every data set, TAb attains high concept coverage ≥ 0.89 and 1.00 consistency by construction. The caveat is concept-size. Because abduction indiscriminately includes every concept vector consistent with some label in $\Upsilon_\alpha(\mathbf{x})$, the resulting set is an order of magnitude larger than any competitor: 504.38 and 742.55 on CEB for DPL, supervised and unsupervised setting respectively, 1024 for LTN instead (the entire concept space), 171.66 and 288.11 on CIF10 for DPL and LTN unsupervised, 2.20-50.00 on the MNIST variants. TAb therefore satisfies **D1**–**D2** on the concept side but fails **D3** dramatically.

CDe **achieves consistency but inflated label sets**. CDe constructs a label set by deducing $g^\dagger(\Gamma_\beta(\mathbf{x}))$ from the concept set. It achieves both label and concept coverage near the nominal level and 1.00 consistency by construction (Proposition 3.3). The failure mode is on the label side: the deduced label set can be substantially larger than TO. On CEB (DPL, supervised), as an example, CDe produces label sets of size 3.71 vs. 2.41 of TO and on MN-H (DPL) 9.00 vs. 0.89. And the same trend could be observed in every tested datasets. The issue is that every element of $\mathbf{\Gamma}^\times(\mathbf{x})$, even low-probability ones, contributes a label to $\Upsilon^{\mathsf{de}}(\mathbf{x})$. CDe is therefore a reasonable baseline when label size is not a binding constraint, but it inflates the label set failing **D3** on the labels.

RPB **closes label-side consistency but leaves the concept set inconsistent**. RPB [57] revises the label set as $\Upsilon^{\mathsf{rev}}(\mathbf{x}) = \Upsilon_\alpha(\mathbf{x}) \cap g^\dagger(\mathbf{\Gamma}^\times(\mathbf{x}))$, but leaves $\mathbf{\Gamma}^\times(\mathbf{x})$ untouched. It therefore produces *identical* label sets to COCOCO by construction, but underperforms on the concept side: concept consistency is typically 0.50-0.94 across the eight data sets (vs. 1.00 for COCOCO), and the concept set is at least as large as COCOCO in every case where the two differ. Representative gaps include CEB (DPL, supervised), $0.78 \to 1.00$ consistency, $75.06 \to 53.49$ size, CHX (DPL, supervised) $0.55 \to 1.00$, $5.30 \to 2.90$, MN-EO (DPL) $0.12 \to 1.00$, $100.00 \to 12.13$, MN-EO (LTN) $0.08 \to 1.00$, $100.00 \to 8.05$ and MN-H (DPL) $0.09 \to 1.00$, $25.00 \to 2.20$. The takeaway is that revising only the label side leaves the concept side inconsistent with the deduced labels; the symmetric revision in COCOCO is what closes this gap and delivers **D1** on *both* sides while preserving **D2**.

Summary. Each one-sided baseline achieves one desideratum at the cost of another: TO ensures label coverage but sacrifices concept coverage *and* consistency; CO ensures concept coverage but sacrifices consistency and concept conciseness; TAb ensures concept coverage and consistency but sacrifices concept conciseness; CDe ensures consistency but sacrifices label conciseness; RPB ensures label consistency but leaves the concept set inflated and inconsistent. COCOCO is the only method to revise both sides symmetrically in a single deduction–abduction step (Proposition B.7) and to attain **D1**, **D2**, and **D3** simultaneously across all eight data sets and both NeSy backbones.

E.7 Time Overhead Analysis

A key point of COCOCO is that the joint revision operator reaches a fixed point in one step (Proposition 4.1), making the additional cost a single computation of the abductive and deductive images, i.e., $\Gamma^{\mathsf{ab}}(\mathbf{x}) = g^{-\dagger}(\Upsilon_\alpha(\mathbf{x}))$ and $\Upsilon^{\mathsf{de}}(\mathbf{x}) = g^{\dagger}(\Gamma_\beta(\mathbf{x}))$.

In practice, as shown by Section E.7 and Fig. 10, COCOCO is comparable in runtime to all other ablations. While it is naturally slower than TO and RPB since it requires additional revision steps, Proposition B.7 ensures it remains tractable thanks to the ***fixed-point*** theorem (Proposition B.7).

Looking at the aggregate results in Section E.7, COCOCO exhibits an overhead of approximately 160% relative to DPL and LTN when considered individually in the unsupervised setting (100% horizontal line), and around 120% in the supervised setting. This is comparable to RPB (140% and 110%, respectively) and to CDe ($130 - 150\%$ unsupervised), and only moderately higher than TO, which is the closest to the baseline.

Looking at individual datasets in Fig. 10, the overhead of COCOCO is dataset-dependent, as it depends on the number of concepts and labels produced by the conformal method, but remains consistently bounded, except for CEB, where the number of concept combinations is $4\times$ higher than the baseline due to the large number of generated concepts and labels. Notably, in CHX and RIV10, the overhead is almost non-existent, demonstrating that in practical applications the overhead is negligible. In fact, COCOCO is well-suited to settings where a human is in the loop and the number of concepts and produced labels is close to one, precisely the scenario in which COCOCO excels.

Furthermore, when employing COCOCO with e-values (COCOCO*), as well as in other ablations, the computational overhead is approximately $200 - 250\%$ in the unsupervised setting and close to 150% in the supervised setting. This is worse than COCOCO even with the speed-up arising from avoiding the Bonferroni correction (Proposition B.9), as it eliminates the need to enumerate the full product set (Eq. (5)). COCOCO*, used in a p-value-like fashion, produces consistently larger concept and label sets, except in CHX and RIV10. Nevertheless, unlike all other approaches, COCOCO* can be fixed in size and thus in computational overhead. In fact, with appropriate choices of $\hat{\beta}$ and $\hat{\alpha}$, namely those that keep the set size below a given threshold (as shown in Section 6), we can cap the computational cost, since we know in advance the size of the instances on which the inference layer g will be applied.

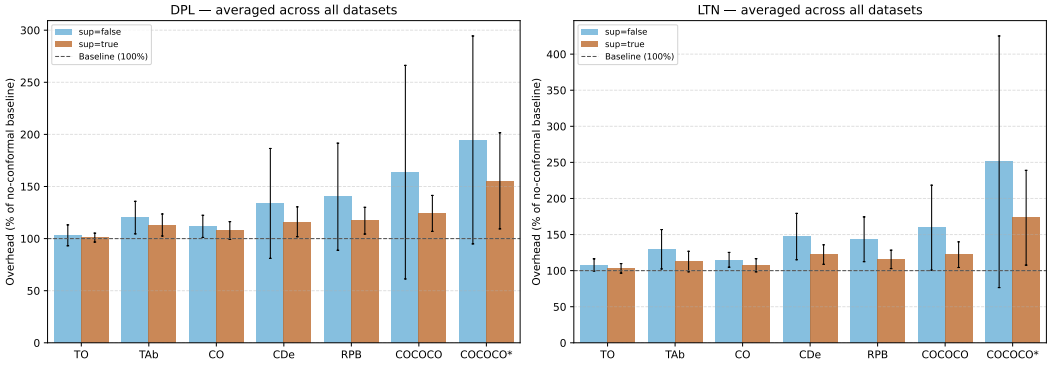


Figure 9: Time overhead (%) relative to DPL (left) and LTN (right) baseline (dashed line, 100%) for all methods (TO, TAb, CO, CDe, RPB, COCOCO and COCOCO*) averaged across all 10 seeds and all datasets. Blue shows the case where concept supervision was absent, and orange shows the case where it was present. Error bars indicate standard deviation.

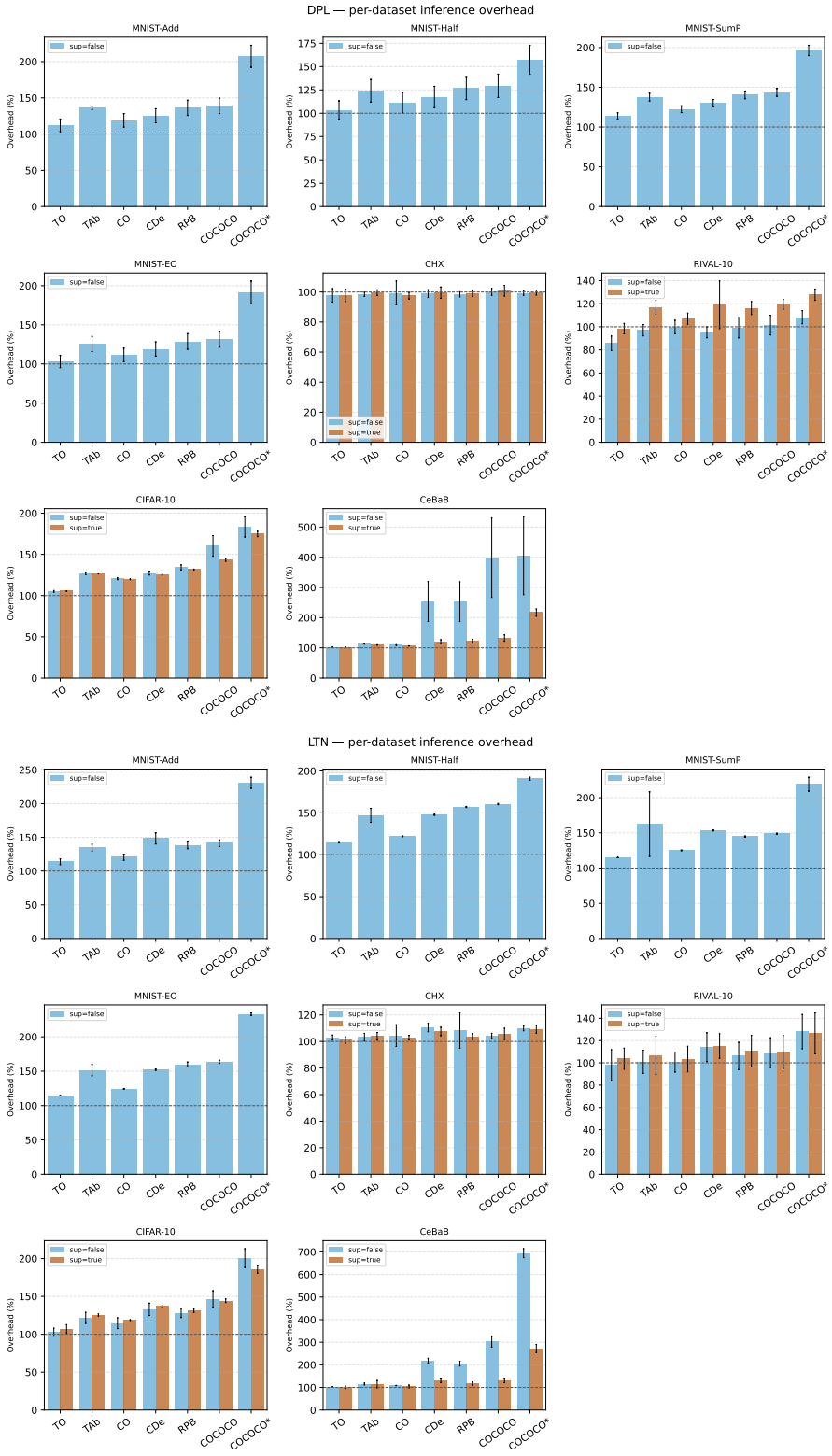


Figure 10: Time overhead (%) relative to the DPL and LTN baseline (dashed line, 100%) for all methods (TO, TAb, CO, CDe, RPB, COCOCO, and COCOCO*), across all 9 datasets, averaged over 10 seeds. The top panel reports results with the DPL baseline, and the bottom panel with the LTN baseline. Blue indicates absence of concept supervision, while orange indicates its presence. Error bars denote standard deviation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the main claims made in the abstract and introduction accurately reflect the papers contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provided a paragraph discussing limitation in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The theory is based on existing research on conformal predictions [13] and e-conformal predictions [18]. All the proofs of the proposition stated in the text are in the appendix Section B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental details are discussed in Section 6. The dataset details are described in Section C.5, metrics details in Section C.2, Neuro-Symbolic architecture details Section C.4 and neural backbone details in Section D.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release an open-source implementation of our code upon paper acceptance. Reviewers can inspect the code in the supplementary materials.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Dataset details are descibed in Section C.5, hyperparameters and backbones are described in Section D.1 and Section D.2. Neuro-Symbolic models implementation is discussed in Section C.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For each experiment in Section 6, we report error bars, standard deviations or confidence intervals, depending on the table/plot. The same applies to additional experiments in Section E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the information related to the computational resources is reported in Section C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impact in an apposite section at the beginning of the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The authors of the datasets and pre-trained models used in this paper are properly cited in the main text, and all asset licenses have been respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowd-sourcing experiments nor research involving human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not describe LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.