NEURAL CONCEPT VERIFIER: SCALING PROVER-VERIFIER GAMES VIA CONCEPT ENCODINGS

Anonymous authors

000

001

002003004

006

008 009

010 011

012

013

014

015

016

017

018

019

021

023

025

026027028

029

031

033

034

037

040

041

042

043

044

046

047

048

051

052

Paper under double-blind review

ABSTRACT

While Prover-Verifier Games (PVGs) offer a promising and much needed path toward verifiability in nonlinear classification models, they have not yet been applied to complex inputs such as high-dimensional images. Conversely, Concept Bottleneck Models (CBMs) effectively translate such data into interpretable concepts but are limited by their reliance on low-capacity linear predictors. In this work, we push towards real-world verifiability by combining the strengths of both approaches. We introduce Neural Concept Verifier (NCV), a unified framework combining PVGs for formal verifiability with concept encodings to handle complex, high-dimensional inputs in an interpretable way. NCV achieves this by utilizing recent minimally supervised concept discovery models to extract structured concept encodings from raw inputs. A prover then selects a subset of these encodings, which a verifier, implemented as a nonlinear predictor, uses exclusively for decision-making. Our evaluations show that NCV outperforms CBM and pixel-based PVG classifier baselines on high-dimensional, logically complex datasets and also helps mitigate shortcut behavior. Overall, we demonstrate NCV as a promising step toward performative, verifiable AI.

1 Introduction

Deep learning has achieved remarkable predictive performances, but often at the expense of interpretability and trustworthiness (Rudin, 2019). However, particularly in high-stakes applications, it is critical that models provide verifiable justifications for their decisions (Irving et al., 2018; Fok & Weld, 2023). Prover-Verifier Games (PVGs), introduced by Anil et al. (2021), formalize such justifications via a gametheoretic approach: a prover provides evidence to convince a verifier, who accepts only verifiable proofs. A prominent instantiation of PVGs is the *Merlin-Arthur Classifier* (Wäldchen et al., 2024), i.e., a classifier guided by cooperative and adversarial provers, offering formal interpretability guarantees through information-

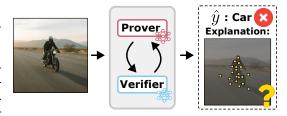


Figure 1: Challenges of Prover-Verifier Games (PVGs) in image classification: (i) It is non-trivial to scale up for high-dimensional data. (ii) Furthermore, the learned explanation masks on the pixel level remain difficult for humans to understand.

theoretic bounds. However, the Merlin-Arthur framework faces significant scalability challenges when applied to high-dimensional real-world data, as explanations based on raw pixels are both computationally difficult to optimize and offer limited human understandability (*cf.* Fig. 1; Wäldchen et al. (2024)).

Concurrently, *Concept Bottleneck Models* (CBMs) have emerged as a powerful framework for interpretable machine learning, structuring predictions through intermediate interpretable concept encodings (Koh et al., 2020; Stammer et al., 2021). Despite their advantages, CBMs typically employ *linear classifiers* on top of concept encoding layers, thereby potentially restricting expressivity and failing on tasks requiring nonlinear interactions among concepts (e.g., XOR problems, counting or permutation invariance, *cf.* Suppl. B) (Mahinpei et al., 2021; Kimura et al., 2024; Lee et al., 2019).

In this work, we combine the best of both worlds by introducing the **Neural Concept Verifier (NCV)**, a novel framework integrating concept-based representations into PVGs in the form of Merlin-Arthur Classifiers. NCV shifts the prover–verifier interaction from the image level to a structured, symbolic concept level, overcoming both the scalability limitations encountered by Merlin-Arthur Classifiers in high-dimensional settings and the expressivity constraints inherent to linear CBMs. Through extensive evaluations on controlled synthetic benchmarks such as CLEVR-Hans, as well as large-scale real-world datasets including CIFAR-100 and ImageNet-1k, we demonstrate that NCV successfully scales PVGs to complex, high-dimensional classification tasks. At the same time, NCV enables verifiable, performant nonlinear classifiers on top of concept extractors, effectively narrowing the interpretability–accuracy gap present in standard, linear CBMs. Lastly, our framework improves robustness to shortcut learning, thereby enhancing the generalizability and trustworthiness of predictions, particularly crucial in high-stakes applications.

Our contributions can be summarised as: (i) We propose *Neural Concept Verifier* (NCV), a framework combining concept-based models with Prover–Verifier Games (PVGs). (ii) NCV scales PVGs to high-dimensional image data by operating on compact concept encodings. (iii) It enables expressive yet interpretable classification via sparse, nonlinear reasoning over concepts. (iv) We validate NCV on synthetic and real-world benchmarks, demonstrating strong accuracy and verifiability. (v) We highlight that NCV improves generalization under spurious correlations, indicating increased robustness.

The remainder of the paper is structured as follows. We begin with a review of related work, highlighting recent developments in the field. We then introduce our proposed Neural Concept Verifier framework, and present its formal description. This is followed by a comprehensive experimental evaluation that investigates key aspects of the framework. Finally, we discuss our findings and conclude the paper.

2 Background

Prover-Verifier Games. PVGs were introduced by Anil et al. (2021) as a game-theoretic framework to encourage learning agents to produce *testable* justifications through interactions between an untrusted prover and a trusted verifier. Their work showed that under suitable game settings, the verifier can learn robust decision rules even when the prover actively attempts to persuade it of arbitrary outputs. Wäldchen et al. (2024) extended this idea to the *Merlin-Arthur Classifier* (MAC), which provides formal interpretability guarantees by bounding the mutual information between the selected features and the ground-truth label. Recently, Kirchner et al. (2024) applied a PVG-inspired approach to improve legibility of Large Language Model (LLM) outputs and Amit et al. (2024) introduced *self-proving models* that leverage interactive proofs to formally verify the correctness of model outputs. PVG-style setups have also been explored in safety-focused learning protocols (Irving et al., 2018; Brown-Cohen et al., 2024; Głuch et al., 2024). These developments reflect a broader trend of utilising multi-agent learning (Pruthi et al., 2022; Schneider & Vlachos, 2024; Du et al., 2024; Nair et al., 2023; Stammer et al., 2024a). Our work builds on this line of research by embedding PVGs into a concept-based classification framework, addressing the dimensionality bottleneck that limits PVGs in high-dimensional settings.

Concept Representations for Interpretability. The introduction of Concept Bottleneck Models (CBMs) (Koh et al., 2020; Delfosse et al., 2024), but also concept-based explanations (Kim et al., 2018; Crabbé & van der Schaar, 2022; Poeta et al., 2023; Lee et al., 2025) was an important moment in the growing interest in AI interpretability research. The particular appeal of CBMs lies in the promise of interpretable predictions and a controllable, structured interface for human interactions (Stammer et al., 2021). While initial CBMs relied on fully supervised concept annotations, subsequent research has relaxed this requirement by leveraging pretrained vision-language models like CLIP for concept extraction (Bhalla et al., 2024; Yang et al., 2023; Oikarinen et al., 2023; Panousis et al., 2024; Steinmann et al., 2025), or employing fully unsupervised concept discovery methods (Ghorbani et al., 2019; Stammer et al., 2024b; Schut et al., 2025; Sawada & Nakamura, 2022). While previous work has focused on enhancing the concept bottleneck itself, by reducing supervision requirements, mitigating concept leakage (Havasi et al., 2022), or dynamically expanding the concept space (Shang et al., 2024), the classifier component of CBMs, has received little attention. We argue that while a

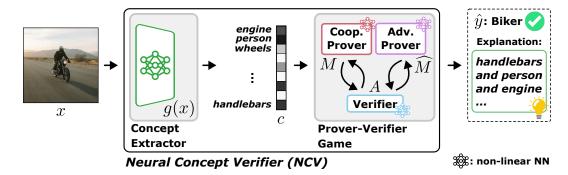


Figure 2: Overview of the Neural Concept Verifier (NCV). The input image is first processed by a concept extractor to produce symbolic concept encodings. A prover–verifier game is then played over these encodings: a cooperative prover selects a sparse concept subset supporting the true class, while an adversarial prover selects misleading concepts. Finally, the nonlinear verifier makes a prediction based only on these selected concepts, ensuring verifiable and robust classification.

linear layer offers interpretability, its limited expressiveness makes it unsuitable for handling complex, compositional, and object-centric tasks. Our proposed NCV tackles this limitation.

Shortcut Learning. Independent of their interpretability, training deep models can often lead to unwanted artifacts and side effects. Shortcut learning describes the problem of models learning to rely on unwanted and unintended features to resolve a task (Geirhos et al., 2020). This is a common problem when training purely deep learning models (Lapuschkin et al., 2019; Schramowski et al., 2020) or neuro-symbolic models (Marconato et al., 2023; Bortolotti et al., 2025), and, if not taken care of, can lead to predictions being right for the wrong reasons (Ross et al., 2017). There have been various methods developed to tackle this problem, from careful dataset curation (Ahmed et al., 2021) to modified model training (Friedrich et al., 2023), *cf.* (Steinmann et al., 2024) for a comprehensive overview. However, these mitigation methods rely on several important assumptions about the training data and potential shortcuts to mitigate, and can affect model performance (Sagawa et al., 2019). While NCV is not specifically designed to mitigate shortcuts, we show that our setup can intrinsically mitigate the impact of shortcuts in the data.

3 NEURAL CONCEPT VERIFIER (NCV)

In this section, we introduce the *Neural Concept Verifier* (NCV), a novel framework for training verifiable nonlinear classifiers capable of handling high-dimensional inputs. NCV combines concept-based representations with the Prover-Verifier Game setup of Merlin-Arthur Classifiers (MAC) (Wäldchen et al., 2024), enabling models whose predictions are certified to rely on a sparse set of high-level concepts. NCV consists of two main components (*cf.* Fig. 2): (i) a minimally or weakly supervised *concept extractor* that transforms input data into interpretable concept encodings, and (ii) a nonlinear MAC that selects and verifies sparse concept-based inputs to make final predictions. After providing background notations and a high-level overview of the NCV framework below, we provide detailed descriptions of each main component as well as training and inference details in the following.

3.1 PROBLEM SETUP AND NOTATION

Let $\mathcal{X} \in \mathbb{R}^{N \times D}$ denote a dataset of N inputs (e.g., images), each of dimension D, and $\mathcal{Y} \in \mathbb{R}^{N \times (K+1)}$ the corresponding labels for K classes plus one additional *rejection class*. The rejection class allows the model to abstain from a decision when uncertain, which is important for enforcing interpretability guarantees in adversarial setups (Wäldchen et al., 2024). Our overall goal is to learn a model $f: \mathcal{X} \to \mathcal{Y}$ that maps an input $x \in \mathcal{X}$ to a prediction $\hat{y} \in \mathcal{Y}$, where the prediction is based on a small, interpretable subset of high-level concepts.

3.2 THE NCV FRAMEWORK

NCV next decomposes f into the following components (cf. Fig. 2 for an illustrative overview):

- 1. A **concept extractor** $g: \mathcal{X} \to \mathcal{C}$, which maps each input x to a high-level concept encoding $\mathbf{c} \in \mathbb{R}^C$, where $C \in \mathbb{N}$ is the number of discovered concepts.
- 2. A pair of **prover agents** $M, \widehat{M} : \mathcal{C} \to \{0,1\}^C$ that produce sparse binary masks selecting m concepts each. M (Merlin, *cooperative Prover*) aims to help classification; \widehat{M} (Morgana, *adversarial Prover*) aims to mislead, which is crucial for overall robustness.
- 3. A nonlinear **verifier** (Arthur) $A : \mathbb{R}^C \to \mathcal{Y}$, which predicts a label based only on the masked concepts.

The three agents are trained jointly, where the interaction between these agents encourages the verifier to rely only on robust, informative concept features. Let us now provide details on these individual components.

3.3 CONCEPT EXTRACTION

The concept extractor $g: \mathcal{X} \to \mathbb{R}^C$ transforms raw input data into interpretable, high-level concept representations, where each dimension corresponds to a semantically meaningful concept. The resulting concept encoding $\mathbf{c} \in \mathbb{R}^C$ serves as the input to the PVG.

While conceptually simple, a careful combination of PVGs and concept-based models is necessary. The concept extractor must satisfy three key requirements: interpretability, expressiveness, and modularity. Concepts should correspond to human-understandable features that can serve as meaningful explanations, the concept space should capture sufficient information for the target task without creating information bottlenecks, and the extractor should operate independently of the prover-verifier components. Unlike traditional concept extractor approaches that enforce sparsity constraints directly on c, our framework delegates sparsity to the prover-verifier interaction. This allows the concept space to remain dense and expressive while achieving interpretable sparsity through downstream concept selection. Further, NCV can accommodate different supervision paradigms for concept extraction: supervised methods that leverage predefined concept vocabularies, self-supervised methods that exploit multi-modal correspondences (e.g., vision-language alignment), or unsupervised methods that discover latent conceptual structures can all be used as concept extractors.

Overall, NCV requires that g produces consistent, interpretable encodings while maintaining sufficient information for accurate classification. In our evaluations, we instantiate NCV's concept extractor via the recent unsupervised and object-centric NCB framework (Stammer et al., 2024b) and the multi-modal, CLIP-based SpLiCE (Bhalla et al., 2024) approach. Operating in concept space rather than raw input space provides: (i) scalability through dimensionality reduction and (ii) explanations based on human-interpretable concepts.

3.4 VERIFIABLE CLASSIFICATION VIA THE MERLIN-ARTHUR CLASSIFIERS

The second core component of NCV is a verifiable classifier of the Merlin-Arthur setup (Wäldchen et al., 2024) originally inspired by Interactive Proof Systems (Goldwasser et al., 1985). This setup generally formalizes the idea of proving that a classification decision is supported by a sparse and informative set of features.

In NCV, specifically, two competing provers, **Merlin** and **Morgana**, select concept subsets either to support or mislead classification, respectively. The verifier, Arthur, then makes predictions based solely on these masked concepts without knowledge of the prover's intent. Formally, each prover outputs a sparse binary mask with m active entries, producing selected subsets:

$$S = M(\mathbf{c}) \odot \mathbf{c}, \quad \widehat{S} = \widehat{M}(\mathbf{c}) \odot \mathbf{c},$$

where c is the concept encoding from extractor g, and \odot denotes element-wise masking. Notably, all three agents represent differentiable models, with Arthur specifically representing a nonlinear model.

This interactive setup enables two key metrics: (i) **completeness**, *i.e.*, the probability that Arthur correctly classifies the input when using Merlin's selected features and (ii) **soundness**, *i.e.*, the

probability that Arthur avoids incorrect classification when using Morgana's features, either by giving the correct class or opting for a rejection. Overall, these components in NCV ensure that only sparse, faithful concept subsets are used and can be inspected for final predictions.

Sparsity has recently become central in concept-based models, as large concept spaces require sparse predictions for interpretability. Prior work (Bhalla et al., 2024; De Santis et al., 2025) achieves this by regularizing the concept space itself, restricting the number of active concepts before training the classifier. In contrast, NCV keeps the concept space fully expressive and enforces sparsity only in the concepts passed to the classifier. At inference, Arthur predicts solely from the masked concepts selected by Merlin, ensuring sparse, interpretable predictions without limiting the richness of the concept space.

3.5 Training and Inference

The training step of NCV incorporates updating only the parameters of the three agents M, \widehat{M} , and A as g represents a pretrained model that is subsequently frozen upon NCV's multi-agent training step. Thus, the three agents are jointly trained by optimizing a three-agent game, which encourages Arthur to rely on concepts selected by Merlin, while being robust to potentially misleading concepts selected by Morgana. Given a concept encoding $\mathbf{c} \in \mathbb{R}^C$, label $y \in \mathcal{Y}$ and cross-entropy function $CE(\cdot,\cdot)$, we define:

- Merlin's loss: $L_M = CE(A(S), y)$, where S is the sparse concept subset selected by Merlin. This loss encourages Arthur to classify correctly based on Merlin's input.
- Morgana's loss: $L_{\widehat{M}} = CE(A(\widehat{S}), y)$, where \widehat{S} is Morgana's adversarial concept subset. Here, the loss is interpreted as the classifier's inability to be misled by deceptive inputs¹.

Overall, Arthur's loss combines both objectives with a hyperparameter $\gamma \in \mathbb{R}_{\geq 0}$, controlling the emphasis on predictive performance (*completeness*) versus robustness (*soundness*):

$$L_A = (1 - \gamma) L_M + \gamma L_{\widehat{M}}, \tag{1}$$

In detail, the three agents are updated jointly in a two-phase min-max optimization. First, the prover agents are updated where Merlin minimizes L_M and Morgana maximizes $L_{\widehat{M}}$; then Arthur is updated by minimizing L_A on the sparse selected concepts chosen by the provers. This scheme incentivizes Arthur to base its predictions on informative, task-relevant, and verifiably robust concept subsets.

At inference time, only the cooperative prover M is used to select a sparse subset of concepts, based on the input's concept encoding. The verifier A then predicts a label or rejects based solely on this selected subset.

Overall, by integrating concept-extractor modules and leveraging the Merlin–Arthur framework, NCV emphasizes faithfulness and interpretability while preserving nonlinear modeling capabilities, and shifts the min–max optimization into a lower-dimensional concept space, improving efficiency, scalability, and stability in high-dimensional settings, a common challenge in min–max optimization for deep learning (Mescheder et al., 2018; Nagarajan & Kolter, 2017).

4 EXPERIMENTAL EVALUATIONS

In this section, we present a comprehensive evaluation of Neural Concept Verifier (NCV) on both synthetic and real-world high-dimensional image datasets. We evaluate based on two instantiations of NCV that utilize different concept extractors: a CLIP-based extractor and the Neural Concept Binder (NCB). We assess predictive performance and interpretability across multiple datasets, compare against several baselines and examine scalability and robustness against shortcut learning.

Our evaluation is structured around the following research questions: (Q1) Does shifting Prover-Verifier Games (PVGs) to concept-encodings via NCV lead to performative classifiers on high-dimensional synthetic and real-world images (*i.e.*, high completeness and soundness)? (Q2) Does NCV reduce the "interpretability-accuracy gap" in the context of CBMs? (Q3) Does NCV allow

¹In practice, the CE loss of Morgana is a slightly modified CE loss, cf. Wäldchen et al. (2024).

for more detailed explanations over pixel-based PVGs? Finally, (Q4) Can training via NCV reduce shortcut learning?

4.1 EXPERIMENTAL SETUP

Datasets. We investigate NCV on CLEVR-Hans3 and CLEVR-Hans7 (Stammer et al., 2021), synthetic benchmarks derived from CLEVR (Johnson et al., 2017) that capture complex object compositions and include visual shortcuts. CLEVR-Hans3 features three compositional classes, while CLEVR-Hans7 increases the complexity to seven, with all images rendered at 128×128 pixels. The training and validation sets contain spurious correlations between attributes and labels (e.g., gray cubes linked to a specific class), which are absent in the test set, making the datasets well-suited for studying shortcut behavior. Models that exploit such correlations often fail under the decorrelated test distribution. We first report results on non-confounded versions of these datasets, where feature distributions are consistent across splits, and later return to the confounded versions for shortcut mitigation. To assess scalability and generalization to natural images, we additionally evaluate on ImageNet-1k (Deng et al., 2009) with 1.2M high-resolution images across 1,000 classes (resized to 224×224 pixels), and on CIFAR-100 (Krizhevsky, 2009) with 60,000 low-resolution 32×32 images across 100 fine-grained categories. Lastly, we perform experiments on COCOLogic (Steinmann et al., 2025), a recent benchmark combining real-world images with complex, compositional class rules.

Baseline Models. We compare our framework against several representative baselines, with training details provided in Suppl. C. As a strong but non-interpretable baseline, we use a standard ResNet-18 (He et al., 2016) for evaluations on CLEVR-Hans, and a ResNet-50 for CIFAR-100, CO-COLogic and ImageNet-1k, each trained end-to-end on raw images. We further evaluate a pixel-based MAC (Wäldchen et al., 2024) (denoted as *Pixel-MAC*), an instantiation of the Prover-Verifier Game in which the verifier is initialized from a pretrained ResNet-18, while both provers (Merlin and Morgana) are U-Net models (Ronneberger et al., 2015) that output continuous feature-importance masks over the input image. These masks are discretized using Top-k selection to define the features visible to the verifier, and all agents are jointly fine-tuned; the resulting explanations (*i.e.*, certificates) correspond to masks in pixel space (see (Wäldchen et al., 2024) for further details). Lastly, we compare to a vanilla Concept Bottleneck Model (Koh et al., 2020) (denoted as *CBM*), where a linear classifier predicts from concept features extracted by either NCB (Stammer et al., 2024b) for CLEVR-Hans or SpLiCE (Bhalla et al., 2024) for CIFAR-100, ImageNet-1k and COCOLogic.

NCV Instantiations. For CLEVR-Hans3 and CLEVR-Hans7, we instantiate NCV with NCB (Stammer et al., 2024b) as the concept extractor, using models pretrained on CLEVR (Johnson et al., 2017). A permutation-invariant Set Transformer (Lee et al., 2019) serves as the verifier (Arthur) to process the unordered NCB encodings. The provers (Merlin and Morgana) are independent Set Transformers that take the full concept-slot encodings as input and output a sparse mask of 12 active concepts for the verifier. All components are jointly trained with the Adam optimizer (Kingma, 2014). Further details and ablations are provided in Suppl. D. For ImageNet-1k, CIFAR-100 and COCOLogic, we use a CLIP-based concept extractor (Radford et al., 2021), following the approach of SpLiCE (Bhalla et al., 2024) to compute image—text similarity scores with a fixed concept vocabulary. Unlike SpLiCE, which performs per-sample optimization, our method (denoted as *CLIP-Sim*) retains the full activation vector and delegates concept selection to the provers, avoiding expensive inference-time optimization and enabling scalability. Here, the verifier and both provers are two-layer MLPs; the provers output sparse masks of 32 concepts per example. All modules are trained with Adam. Additional details and ablations are given in Suppl. E.

Metrics. All methods are evaluated for completeness and, where applicable, soundness (Sec. 3.4). Hereby, completeness corresponds to classification accuracy under cooperative feature selection, while soundness measures robustness to adversarial prover selections. We use 20 random seeds for CLEVR-Hans and 10 for ImageNet-1k, CIFAR-100 and COCOLogic, reporting mean and standard deviation across all seeds. For CLEVR-Hans shortcut learning, we additionally report the *validation–test gap*, i.e., the difference between validation accuracy on a confounded split and test accuracy on a non-confounded split; smaller gaps indicate better generalization and reduced reliance on spurious correlations.

Table 1: NCV delivers high predictive performance and robustness through verifiable, concept-based reasoning. We report completeness and soundness scores for ResNet, Pixel-MAC, CBM, and NCV across synthetic (CLEVR-Hans3, CLEVR-Hans7) and real-world (CIFAR-100, ImageNet-1k, COCOLogic) datasets. NCV consistently matches or outperforms baselines in completeness while offering strong soundness guarantees.

Model	Feature Space	Completeness (Accuracy)	Soundness (Robustness)	Completeness (Accuracy)	Soundness (Robustness)	Completeness (Accuracy)	Soundness (Robustness)
		CIFA	R-100	Image	Net-1k	COCO	Logic
ResNet-50	pixel space	81.45 ± 0.60	n/a	76.01 ± 0.02	n/a	65.80 ± 3.41	n/a
Pixel-MAC CBM NCV (ours)	pixel space SpLiCE CLIP-Sim	$15.27 \pm 4.78 \\ 75.42 \pm 0.04 \\ 83.32 \pm 0.28$	96.31 ± 4.12 n/a 99.99 ± 0.01	35.06 ± 3.20 68.59 ± 0.01 67.04 ± 0.16	99.65 ± 0.26 n/a 99.94 ± 0.02	$42.57 \pm 3.13 \\ 58.84 \pm 0.09 \\ \textbf{75.42} \pm 3.21$	97.70 ± 0.61 n/a 97.87 ± 0.47
		CLEVR-Hans3		CLEVR-Hans7			
ResNet-18	pixel space	97.87 ± 0.24	n/a	98.71 ± 0.24	n/a		
Pixel-MAC CBM NCV (ours)	pixel space NCB NCB	$\begin{array}{c} 96.59 \pm 0.72 \\ 95.44 \pm 0.08 \\ 98.92 \pm 0.32 \end{array}$	99.99± 0.01 n/a 100.00± 0.00	$97.61 \pm 0.38 \\ 89.12 \pm 0.12 \\ 97.89 \pm 0.31$	99.88± 0.28 n/a 100.00± 0.00		

4.2 EVALUATIONS

Scaling PVGs to High Dimensions (Q1). In our first evaluation, we examine whether shifting the Prover–Verifier Game (PVG) to concept encodings enables NCV to scale to high-dimensional image domains while achieving strong performance in terms of completeness and soundness. We hereby compare NCV against two key baselines: (1) a black-box ResNet classifier (ResNet-18 for CLEVR-Hans and ResNet-50 for CIFAR-100, ImageNet-1k and COCOLogic), and (2) Pixel-MAC, a nonlinear PVG model operating in raw pixel space. Tab. 1 summarizes results across synthetic (CLEVR-Hans3, CLEVR-Hans7) and real-world (CIFAR-100, ImageNet-1k, COCOLogic) benchmarks. Each model's feature space is indicated for clarity. On the synthetic CLEVR-Hans benchmarks, we observe that NCV consistently achieves the highest completeness scores, surpassing Pixel-MAC and even ResNet-18 on CLEVR-Hans3, while also attaining perfect soundness. This demonstrates that NCV not only matches or exceeds the performance of strong black-box classifiers but also maintains faithful and certifiable decision-making. Pixel-MAC performs well in these settings but falls slightly short in completeness and cannot match NCV's zero-error soundness.

On the more challenging real-world datasets, Pixel-MAC either fails entirely or performs poorly. In contrast, NCV successfully scales to these datasets, achieving superior completeness and near-perfect soundness. Notably, NCV surpasses ResNet-50 even in raw accuracy for CIFAR-100 and COCOLogic, providing both higher predictive performance while retaining interpretability. In summary, NCV generalizes well across domains: it scales beyond the limitations of pixel-based PVGs, delivers competitive accuracy even on large-scale and complex datasets, and retains soundness throughout. These findings affirm that shifting PVGs to concept space enables interpretable and verifiable classifiers that remain performant and scalable in high-dimensional synthetic and real-world visual environments. We therefore answer Q1 affirmatively.

Narrowing the Interpretability–Accuracy Gap (Q2). In Tab. 1, we further examine whether NCV can overcome a central limitation of standard Concept Bottleneck Models (CBMs): the interpretability–accuracy gap resulting from their use of constrained linear classifiers (cf. Suppl. B for a discussion). Related to this, we observe that NCV consistently narrows and in some cases even closes this gap, while maintaining formal verifiability across all evaluated datasets. Specifically, on CLEVR-Hans3, the baseline CBM trails the opaque ResNet-18 by over 2 percentage points in completeness, whereas NCV matches or exceeds the ResNet's performance while retaining perfect soundness. The benefit is even more pronounced on CLEVR-Hans7: CBM underperforms ResNet-18 by nearly 10 percentage points, while NCV narrows the gap to just 1 percentage point. This trend persists on real-world datasets. On CIFAR-100, NCV outperforms the base CBM and even slightly exceeds ResNet-50's performance. This is even more pronounced on COCOLogic, where NCV outperforms both the base CBM and ResNet-50 by a large margin. As the additional benefits of a nonlinear classifier are quite small on ImageNet-1k, the additional training complexity of NCV results in a slightly worse performance compared to the base CBM there. Overall, NCV improves over linear CBMs in both accuracy and robustness, especially on tasks requiring complex concept

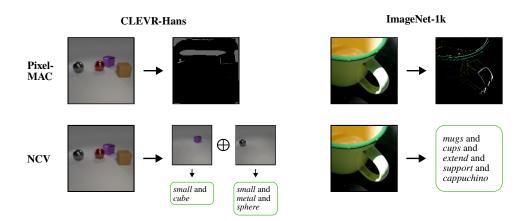


Figure 3: Comparison of explanations from NCV vs. Pixel-MAC. (**top**) Merlin–Arthur training on pixel space yields uninformative masks. (**bottom**) MAC on concept encodings via NCV translates into combinations of high-level concepts and, in turn, in an interpretable prediction.

reasoning. At the same time, it can match or even surpasses the accuracy of the opaque ResNet models, demonstrating that interpretable, concept-level reasoning via Prover–Verifier Games can deliver competitive performance without sacrificing transparency or verifiability. We therefore answer Q2 affirmatively.

More Detailed Explanations (Q3) We next investigate the resulting explanations produced by our NCV framework, with a focus on explanatory clarity. Since our goal is to improve over classic vision-based Prover–Verifier Games, we compare against pixel-level MAC explanations. Fig. 3 illustrates a qualitative example from both the CLEVR-Hans3 (cf. Fig. 4 for more examples) and ImageNet-1k datasets. Notably, under Pixel-MAC, the Prover–Verifier setup operates directly on pixels, yielding broad, diffuse explanation masks that often cover entire objects or irrelevant background regions, arguably providing limited insight regarding which exact features drive the verifier's final decision. In contrast, NCV leverages its internal concept encodings to pinpoint the precise attributes relevant for a class decision, recovering the exact class rule for CLEVR-Hans (i.e. small cube and small metal sphere), and providing a meaningful concept explanation for class coffee-mug of ImageNet-1k.² For ImageNet-1k, the mask size is set to 32 concepts, but for clarity we visualize only the top 5 most frequent concepts across 32 samples. Overall, these examples highlight that NCV offers substantially more detailed and understandable explanations than pixel-based PVG baselines, highlighting that concept-level PVGs yield verifiable and interpretable decisions even for complex, high-dimensional data. This leads us to answer Q3 affirmatively.

Mitigating Shortcut Learning (Q4) Lastly, to assess whether NCV can mitigate shortcut learning in image classification, we train models on different versions of CLEVR-Hans3 and CLEVR-Hans7 with varying ratios of clean samples (i.e., without shortcut) in the training and validation sets. We then measure validation accuracy with shortcuts and test accuracy on a held-out, clean data split. This setup allows us to track both predictive performance and robustness to shortcut learning. Tab. 2 reports results for three model types: a linear CBM, a nonlinear CBM, and our instantiation of NCV using NCB as concept extractor. We observe that while NCV achieves the highest test accuracy among all models in the 0% clean data setting, it still exhibits a sizeable validation-test gap, indicating a strong influence of the underlying shortcuts. As the amount of clean samples is progressively increased, test accuracy and test-validation gap improves across all models. However, NCV consistently achieves the highest test accuracy in every setting, and its validation-test gap decreases more rapidly than for either CBM variant. This trend indicates that NCV is not only better at leveraging clean supervision when available, but is also more robust to shortcut learning. Together, these results demonstrate that concept-level Prover-Verifier Games in NCV encourage models to rely on robust, task-relevant features, making NCV more resilient to shortcut learning, even with limited amounts of clean data.

²The availability of object-level concepts in NCV depends on the underlying concept extractor. For CLEVR-Hans, we use NCB, which provides such object-based explanations.

Table 2: NCV shows increased robustness to shortcuts on CLEVR-Hans3 and CLEVR-Hans7. We report validation accuracy on a set with shortcuts, and test accuracy on a set without shortcuts, as well as the validation–test gap (lower is better) across models trained with varying amounts of clean data.

Ratio	Model	CLEVR-Hans3			CLEVR-Hans7		
Non-Conf. (Samples)	Wiodei	Val Acc (w/ shortcut)	Test Acc (w/o shortcut)	Val-Test Gap (↓)	Val Acc (w/ shortcut)	Test Acc (w/o shortcut)	Val-Test Gap (↓)
0%	CBM (lin.) CBM (non-lin.) NCV	$\begin{array}{c} 95.65 \!\pm\! 0.09 \\ 98.70 \!\pm\! 0.32 \\ \textbf{99.44} \!\pm\! 0.15 \end{array}$	$90.54 \pm 0.09 \\ 95.04 \pm 0.96 \\ 94.21 \pm 1.41$	5.11 3.66 5.23	$90.37 \pm 0.10 \\ 98.09 \pm 0.24 \\ 98.38 \pm 0.18$	$85.27 \pm 0.15 \\ 90.69 \pm 1.17 \\ 92.23 \pm 0.67$	5.10 7.40 6.15
1% (105)	CBM (lin.) CBM (non-lin.) NCV	$\begin{array}{c} 96.28 \pm 0.16 \\ 99.10 \pm 0.27 \\ \textbf{99.37} \pm 0.18 \end{array}$	$\begin{array}{c} 91.03 \pm 0.31 \\ 94.84 \pm 0.98 \\ 97.11 \pm 0.98 \end{array}$	5.25 4.26 2.26	$90.74 \pm 0.12 \\ 98.17 \pm 0.17 \\ 98.19 \pm 0.24$	85.41 ± 0.17 92.65 ± 1.31 94.68 ± 0.64	5.33 5.52 3.51
5% (525)	CBM (lin.) CBM (non-lin.) NCV	$\begin{array}{c} 95.38 \pm 0.37 \\ 98.41 \pm 0.55 \\ 99.59 \pm 0.19 \end{array}$	$\begin{array}{c} 93.34 \!\pm\! 0.51 \\ 96.13 \!\pm\! 0.71 \\ 98.88 \!\pm\! 0.37 \end{array}$	2.04 2.28 0.71	$90.37 \pm 0.15 \\ 98.32 \pm 0.22 \\ 98.47 \pm 0.24$	$86.37 \pm 0.18 \\ 95.19 \pm 0.80 \\ 96.24 \pm 0.71$	4.00 3.13 2.23
20% (2100)	CBM (lin.) CBM (non-lin.) NCV	$\begin{array}{c} 95.67 \pm 0.28 \\ 99.15 \pm 0.21 \\ \textbf{99.37} \pm 0.28 \end{array}$	93.46 ± 0.23 98.09 ± 0.51 98.82 ± 0.67	2.21 1.06 0.55	89.93 ± 0.29 98.21 ± 0.29 98.63 ± 0.13	87.21 ± 0.31 97.00 ± 0.49 97.74 ± 0.28	2.72 1.21 0.89

5 DISCUSSION

Overall, our results show that shifting Prover–Verifier Games (PVGs) to the concept level yields a powerful and scalable framework for verifiable, interpretable classification. By operating on symbolic concept embeddings, NCV avoids the computational cost of per-sample inference in pixel space, yet matches or surpasses pixel-based baselines in both completeness and soundness. It reduces the performance gap typical of Concept Bottleneck Models (CBMs), achieving parity with opaque models on synthetic tasks and even surpassing them on natural images. Additionally, concept-level outputs offer concise, human-readable explanations. Finally, NCV exhibits a resilience to spurious correlations, generalizing from confounded training splits and closing the generalization gap with minimal available clean data.

That said, NCV has several limitations. Its effectiveness depends on the quality of the underlying concept extractor: noisy or entangled concept spaces can reduce both accuracy and human understandability. The increased training complexity introduced by the three-agent PVG setup also results in greater computational cost and training instability, e.g., compared to linear CBMs. Moreover, when using pretrained models like CLIP for concept discovery, NCV inherits their biases and inconsistencies to some extent (Birhane et al., 2021; Gehman et al., 2020; Bhalla et al., 2024). Finally, recent work (Debole et al., 2025) shows that such concept spaces can diverge from expert semantics, even when yielding strong downstream performance.

6 Conclusion

In this work, we have introduced the Neural Concept Verifier (NCV), a unified framework that brings together Prover–Verifier Games and concept-level representations for interpretable classification at scale. Through extensive experiments on CLEVR-Hans, CIFAR-100, ImageNet-1k, and COCOLogic, we have shown that NCV achieves high completeness and soundness, reduces the interpretability–accuracy gap of concept bottleneck models, delivers detailed concept-based explanations, and effectively mitigates shortcut learning. Thus, NCV paves the way for deploying trustworthy and transparent models in domains where both predictive performance and verifiability are essential.

Future work should explore how concept encodings can be integrated into alternative PVG-style setups, where structured representations may improve performance or reduce communication overhead. It is also promising to investigate applications beyond vision, such as natural language processing and structured data, where interpretable verification may be equally valuable. Finally, while existing information-theoretic guarantees, such as those introduced by Wäldchen et al. (2024), focus on binary classification under specific assumptions, extending such guarantees to high-dimensional, multi-class settings remains an important open direction for formal interpretability at scale.

REFERENCES

- Kaoutar Ben Ahmed, Gregory M Goldgof, Rahul Paul, Dmitry B Goldgof, and Lawrence O Hall. Discovery of a generalization gap of convolutional neural networks on covid-19 x-rays classification. *Ieee Access*, 9:72970–72979, 2021.
- Noga Amit, Shafi Goldwasser, Orr Paradise, and Guy Rothblum. Models that prove their own correctness. *CoRR*, abs/2405.15722, 2024.
- Cem Anil, Guodong Zhang, Yuhuai Wu, and Roger Grosse. Learning to give checkable answers with prover-verifier games. *CoRR*, abs/2108.12099, 2021.
 - Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.
 - Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *CoRR*, abs/2110.01963, 2021.
 - Samuele Bortolotti, Emanuele Marconato, Paolo Morettin, Andrea Passerini, and Stefano Teso. Shortcuts and identifiability in concept-based models from a neuro-symbolic lens. *CoRR*, abs/2502.11245, 2025.
 - Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Scalable ai safety via doubly-efficient debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.
 - Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. Advances in Neural Information Processing Systems, pp. 2590– 2607, 2022.
 - Francesco De Santis, Philippe Bich, Gabriele Ciravegna, Pietro Barbiero, Danilo Giordano, and Tania Cerquitelli. Towards better generalization and interpretability in unsupervised concept-based models. *arXiv preprint arXiv:2506.02092*, 2025.
 - Nicola Debole, Pietro Barbiero, Francesco Giannini, Andrea Passeggini, Stefano Teso, and Emanuele Marconato. If concept bottlenecks are the question, are foundation models the answer? *CoRR*, abs/2504.19774, 2025.
 - Quentin Delfosse, Sebastian Sztwiertnia, Mark Rothermel, Wolfgang Stammer, and Kristian Kersting. Interpretable concept bottlenecks to align reinforcement learning agents. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
 - Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255, 2009.
 - Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 21–27 Jul 2024.
 - Raymond Fok and Daniel S Weld. In search of verifiability: Explanations rarely enable complementary performance in ai-advised decision making. *AI Magazine*, 2023.
 - Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology for exploring the mitigation of shortcut behaviour. *Nature Machine Intelligence*, 5(3):319–330, 2023.
 - Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *CoRR*, abs/2009.11462, 2020.

- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias
 Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.
 - Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
 - Grzegorz Głuch, Berkant Turan, Sai Ganesh Nagarajan, and Sebastian Pokutta. The Good, the Bad and the Ugly: Watermarks, Transferable Attacks and Adversarial Defenses. *CoRR*, abs/2410.08864, 2024.
 - S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pp. 291–304. Association for Computing Machinery, 1985.
 - Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. *Advances in Neural Information Processing Systems*, 35:23386–23397, 2022.
 - 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 (CVPR)*, June 2016.
 - Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). CoRR, abs/1606.08415, 2016.
 - Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. volume abs/1805.00899, 2018.
 - Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
 - Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning (ICML)*, pp. 2668–2677. PMLR, 2018.
 - Masanari Kimura, Ryotaro Shimizu, Yuki Hirakawa, Ryosuke Goto, and Yuki Saito. On permutation-invariant neural networks. *CoRR*, abs/2403.17410, 2024.
 - Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
 - Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of llm outputs. *CoRR*, abs/2407.13692, 2024.
 - 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*, pp. 5338–5348. PMLR, 2020.
 - Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
 - Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1–8, 2019.
 - Jae Hee Lee, Sergio Lanza, and Stefan Wermter. From neural activations to concepts: A survey on explaining concepts in neural networks. *Neurosymbolic Artificial Intelligence*, 1:NAI–240743, 2025.
 - Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019.

- Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *CoRR*, abs/2106.13314, 2021.
 - Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. *Advances in Neural Information Processing Systems*, 36:72507–72539, 2023.
 - Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3481–3490. PMLR, 2018.
 - Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent gan optimization is locally stable. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5591–5600, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
 - Varun Nair, Elliot Schumacher, Geoffrey Tso, and Anitha Kannan. Dera: enhancing large language model completions with dialog-enabled resolving agents. *CoRR*, abs/2303.17071, 2023.
 - Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *CoRR*, abs/2304.06129, 2023.
 - Konstantinos P. Panousis, Dino Ienco, and Diego Marcos. Coarse-to-fine concept bottleneck models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 105171–105199. Curran Associates, Inc., 2024.
 - Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
 - Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey. *CoRR*, abs/2312.12936, 2023.
 - Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022.
 - Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.
 - Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI* 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pp. 234–241. Springer, 2015.
 - Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2662–2670, 2017.
 - Cynthia Rudin. Stop explaining black box models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.
- Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust
 neural networks for group shifts: On the importance of regularization for worst-case generalization.
 CoRR, abs/1911.08731, 2019.
 - Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.

- Johannes Schneider and Michalis Vlachos. Reflective-net: Learning from explanations. *Data Mining and Knowledge Discovery*, 38(5):2975–2996, 2024.
- Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.
- Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *CoRR*, abs/2111.02114, 2021.
- Lisa Schut, Nenad Tomašev, Thomas McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human–ai knowledge gap through concept discovery and transfer in alphazero. *Proceedings of the National Academy of Sciences*, 122(13), 2025.
- Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11030–11040, 2024.
- Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Conference on Computer Vision and Pattern Recognition CVPR*, pp. 3619–3629, 2021.
- Wolfgang Stammer, Felix Friedrich, David Steinmann, Manuel Brack, Hikaru Shindo, and Kristian Kersting. Learning by self-explaining. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856.
- Wolfgang Stammer, Antonia Wüst, David Steinmann, and Kristian Kersting. Neural concept binder. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 71792–71830. Curran Associates, Inc., 2024b.
- 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. *CoRR*, abs/2412.05152, 2024.
- David Steinmann, Wolfgang Stammer, Antonia Wüst, and Kristian Kersting. Object centric concept bottlenecks. *CoRR*, abs/2505.24492, 2025.
- Stephan Wäldchen, Kartikey Sharma, Berkant Turan, Max Zimmer, and Sebastian Pokutta. Interpretability guarantees with Merlin-Arthur classifiers. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pp. 1963–1971. PMLR, 2024.
- Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19187–19197, 2023.

Supplementary Materials

A USE OF LARGE LANGUAGE MODELS

Large language models were used to support this work by assisting with text refinement, implementation of code components (including methods and plot generation), and by providing input during idea development and approach refinement.

B WHY LINEAR CLASSIFIERS FALL SHORT IN CBMS

While linear classifiers are generally considered to be interpretable, these models are not suited to solve arbitrarily complex problems. A linear classifier is only able to capture linear relationships between inputs and output features and cannot model complex, non-linear relationships. In the context of CBMs, this problem is usually tackled by utilizing a linear classifier to predict the output based on the detected concepts. The concepts themselves can be detected using non-linear models, and only the classification based on these concepts is done with a linear model. However, this is not always sufficient, as there are also simple examples where non-linear relationships between concepts and the output exist, for example thresholds detection (three out of five symptoms need to be present to indicate an illness) or multiplicative effects (crop yield is the result of a multiplicative relationship between rain and fertility).

To illustrate the problem in a simple experimental setup, let us assume we have a dataset of simple shapes and every image contains between one and four of these shapes. The shapes are either a square or a circle and either orange or blue. We consider two simple classification scenarios for this dataset.

• XOR: This setting classification follows the traditional XOR problem: We want to classify images that contain either an orange square or a blue circle as class one and all other images as class two.

• Counting: This setting includes object counting and illustrates that even for classification based on a single attribute, a linear layer can be insufficient. Here, we want to classify all images with exactly one blue shape as class one, and all other images as class two.

We evaluate a linear layer and a simple MLP on this toy dataset. To further simplify things, we assume that our concept encoder is able to perfectly detect the concepts in the image, thus providing for each element the information whether there is an object and if so, its shape and color. We randomly generate 5000 samples of the dataset and train the models on a train split of 80% and evaluate on the remaining 20%. The MLP has one hidden layer of size 16 and uses ReLU activation functions.

 The results of this evaluation are shown in Tab. 3. In both scenarios, the linear classification layer is not able to solve the task, despite the deceptively simple relationship between concepts and output classes. On the other hand, the MLP achieves close to perfect accuracy on both settings.

So far, we have argued that not every task can be solved with a CBM and a linear classification layer. However, this is not

Table 3: A linear prediction layer cannot solve *XOR* or *counting*. Even with the assumption of a perfect concept encoder, the linear layer fails.

Model	XOR (Acc)	Counting (Acc)
Linear Layer	0.766 ± 0.011	0.677 ± 0.006
MLP	0.953 ± 0.053	0.982 ± 0.015

entirely accurate. In principle, any task can be solved linearly—provided that we define the right *linear-sufficient* concepts. For instance, in the XOR setting, detecting the concepts "orange square and no blue circle" and "blue circle and no orange square" would allow a linear classifier to solve the task. Similarly, in the counting task, introducing a concept such as "exactly one blue object" would make linear classification trivial.

That said, the assumption that such sufficient concepts are always available is not realistic. First, designing or discovering these concepts often makes concept detection considerably more difficult. Second, as concepts become increasingly specific and compositional, they tend to lose interpretability.

Finally, requiring tailored concepts for every individual task does not scale. Returning to our example, the concept "exactly one blue object" might help with task two but is essentially useless for task one.

Taken together, this illustrates why relying solely on linear classifiers in CBMs is often impractical. To address such cases, non-linear classifiers should also be considered.

C EXPERIMENTAL DETAILS: BASELINES

In this section we provide training details for the three introduced baselines *ResNet-18*, *ResNet-50*, *Pixel-MAC* and *CBM*.

C.1 RESNET-18 AND RESNET-50

We initialize the framework with a pretrained ResNet-18 model and employ the Adam optimizer across all experiments. On the CLEVR datasets, the model is trained with a batch size of 128 for 30 epochs using a learning rate of 10^{-4} and weight decay of 10^{-4} , repeated across 20 random seeds with early stopping based on validation loss.

On CIFAR-100, we use a ResNet-50 trained for 100 epochs with a learning rate of 10^{-4} , weight decay of 10^{-5} , and a batch size of 128, averaged over 10 random seeds with early stopping. The ResNet-50 baseline on ImageNet is evaluated directly using pretrained PyTorch (Paszke et al., 2019) weights without further finetuning. For COCOLogic, a ResNet-50 is trained for 300 epochs with a batch size of 256, learning rate of 10^{-4} , and weight decay of 10^{-2} , again averaged over 10 random seeds with early stopping.

In the Pixel-MAC setup, a separate ResNet-18 is trained under the same configuration as above but with a reduced learning rate of 10^{-5} , while keeping the batch size, weight decay, and early stopping criterion unchanged. All Pixel-MAC results are obtained from these ResNet-18 checkpoints.

C.2 PIXEL-MAC

In this setup, we apply Merlin-Arthur training on pixel space by utilizing the pretrained ResNet-18 models as classifiers and U-Net architectures for both Merlin and Morgana. Throughout all experiments, we employ the Adam optimizer for both classifier and feature selector optimization, with $\gamma=0.5$ to ensure high soundness.

For the CLEVR datasets, we train with a batch size of 128 for 40 epochs, using a learning rate of 10^{-5} and weight decay of 10^{-6} for the classifier optimization. The U-Net architectures are trained with a learning rate of 10^{-4} , weight decay of 10^{-5} and an L1 penalty coefficient of 0.1. We set the mask size to 1500, meaning that the U-Nets select a subset of 1500 pixels per sample (out of 128×128 pixels).

For CIFAR-100, we reduce the batch size to 64 and train for 100 epochs. The classifier is optimized with a learning rate of 10^{-5} and weight decay of 10^{-4} . Both Merlin and Morgana are trained using a learning rate of 10^{-3} and a reduced mask size of 32 pixels. We use an L1 penalty of 0.01 and apply early stopping across 10 random seeds.

For ImageNet and COCOLogic, we further reduce the batch size to 32 due to memory constraints and train for 80 epochs using the same learning rates and hyperparameters as in the CIFAR-100 setting. The mask size is set to 1000 pixels per image. As with CIFAR-100, early stopping is applied across 10 random seeds, and training is initialized from the pretrained ResNet-18 backbone.

C.3 CBM WITH LINEAR CLASSIFIER

Next, we present the implementation details for the CBM baseline, where a linear classifier operates on concept features obtained from the concept extractor.

For the CLEVR datasets, we train a linear classifier on concepts extracted by the Neural Concept Binder. The training process employs a batch size of 128, a learning rate of 10^{-3} , and weight decay of 10^{-4} . The model is trained for 60 epochs on CLEVR-Hans3 and 30 epochs on CLEVR-Hans7, using early stopping based on validation loss, repeated across 20 different random seeds.

For CIFAR-100 and ImageNet, we train a linear classifier on sparse SpLiCE encodings using a dictionary size of 10,000. In both cases, we use a batch size of 4096 and train for 250 epochs with early stopping, a learning rate of 10^{-3} , and no weight decay. A hidden layer with 512 units is used, and an L1 penalty of 0.2 is applied within SpLiCE to encourage sparsity in the concept representations. All results are averaged over 10 random seeds.

D NCB-BASED NEURAL CONCEPT VERIFIER EXPERIMENTS

In the following, we provide details on NCB-based NCV, experimental evaluations as well as additional evaluations.

D.1 PRETRAINING

Before training NCV, we first pretrain the models without the feature selectors. The corresponding results for the pretraining are shown in Tab. 4, where we evaluate on 20 random seeds. These pretrained models are then used as initialization for the subsequent NCV training. For the pretraining, we use a Set Transformer with two stacked multi-head attention blocks, a hidden dimension of 128 and four attention heads. We use a batch size of 128, 30 epochs and the Adam optimizer with a learning rate of 10^{-3} for both datasets, applying early stopping based on validation loss.

Table 4: Pretraining results on the CLEVR-Hans3 and CLEVR-Hans7 datasets without shortcuts

CLEVR-Hans3 CLEVR-Hans7 Val. Accuracy Test Accuracy 99.02 ± 0.31 98.13 ± 0.37 98.08 ± 0.24 97.83 ± 0.25

D.2 NCV TRAINING

For the experiments presented in our main results in Tab. 1, the experimental details for both datasets are as follows:

Model Architecture. The verifier is implemented as a pretrained Set Transformer consisting of two stacked multi-head attention blocks with hidden dimension 128, four attention heads, and layer normalization. Merlin and Morgana are implemented as independent neural networks, each parameterized by a Set Transformer with two stacked attention blocks with hidden dimensions 256, four attention heads, and layer normalization. The provers receive the full concept slot matrix as input and output a sparse selection mask with exactly 12 nonzero entries (out of 64 total features), indicating the active blocks provided to the verifier.

Training Details. All components are jointly trained using the Adam optimizer with a learning rate of 10^{-3} and weight decay of 10^{-4} . Models are trained for 50 epochs and a batch size of 512 is used throughout. For the Merlin and Morgana provers, a hard selection constraint is enforced, limiting the number of selected concepts to a fixed budget of 12 block-encodings per sample. To ensure high soundness, we set $\gamma=0.5$, giving equal weight to both feature selector losses in the total loss computation. We train our models using 20 random seeds.

Extended Results. Additionally, we evaluated the NCV framework with varying mask sizes and an alternative model architecture for the feature selectors. The results are presented in Tab. 5 for the CLEVR-Hans3 dataset and Tab. 6 for the CLEVR-Hans7 dataset, where we evaluate both the validation set and the test set. The alternative architecture implements a MLP with two hidden layers and ReLU activation functions for the feature selectors, while maintaining a pretrained Set Transformer as the classifier across all experiments. Our results reveal that the Set Transformer feature selector consistently outperforms the MLP feature selector on the test set, particularly with smaller mask sizes such as 4 and 6. Furthermore, this configuration maintains high completeness (>96%) and soundness (>99%), even with a reduced number of selected features.

Table 5: Completeness and soundness on the CLEVR-Hans3 dataset without shortcuts for different mask sizes and feature selector architectures. The highlighted values are used for Table 1.

		Validation		Test		
Mask Size	Feature Selector	Completeness	Soundness	Completeness	Soundness	
4	Set Transformer MLP	98.35 ± 0.31 96.51 ± 1.18	99.88 ± 0.28 99.81 ± 0.27	97.69 ± 0.63 95.54 ± 1.37	99.82 ± 0.23 99.85 ± 0.28	
6	Set Transformer MLP	98.71 ± 0.52 96.21 ± 0.89	99.87 ± 0.13 99.96 ± 0.03	98.11 ± 0.62 94.78 ± 1.24	99.88 ± 0.19 99.97 ± 0.06	
12	Set Transformer MLP	99.20 ± 0.11 99.28 ± 0.11	100.00 ± 0.00 99.98 ± 0.06	98.92 ± 0.32 98.89 ± 0.21	$ \begin{array}{c} 100.00 \pm 0.00 \\ 99.99 \pm 0.07 \end{array} $	

Table 6: Completeness and soundness on the CLEVR-Hans7 dataset without shortcuts for different mask sizes and feature selector architectures. The highlighted values are used for Table 1.

		Validation		Test		
Mask Size	Feature Selector	Completeness	Soundness	Completeness	Soundness	
4	Set Transformer MLP	96.69 ± 1.28 92.63 ± 1.24	99.93 ± 0.09 99.89 ± 0.12	96.71 ± 1.37 92.71 ± 1.31	99.91 ± 0.09 99.87 ± 0.13	
6	Set Transformer MLP	97.32 ± 0.42 95.43 ± 1.48	99.98 ± 0.02 99.88 ± 0.13	97.14 ± 0.51 95.12 ± 1.48	99.98 ± 0.02 99.86 ± 0.14	
12	Set Transformer MLP	98.13 ± 0.11 97.41 ± 1.07	100.00 ± 0.00 99.99 ± 0.03	97.89 ± 0.31 97.01 ± 0.93	100.00 ± 0.00 99.99 ± 0.04	

D.3 EXPLANATIONS

Here, we present supplementary examples of explanations generated by both Pixel-MAC and NCV on the CLEVR-Hans3 dataset in Fig. 4. These results further substantiate our claim that NCV provides significantly more transparent and interpretable explanations compared to the pixel-based PVG baseline.

E EXPERIMENTAL DETAILS FOR CLIP-BASED NCV

In the following section, we present the implementation details of CLIP-based NCV training.

E.1 PRETRAINING

Once more, before starting with the actual NCV training, we first pretrain the models without the provers (Merlin and Morgana). The corresponding results for the pretraining are shown in Tab. 7. As textual concept descriptions T, we used the top 10,000 most frequent one- and two-word phrases from LAION (Schuhmann et al., 2021) captions, following the setup of Bhalla et al. (2024). For pretraining the verifier, we use a two-layer multilayer perceptron (MLP) with a hidden dimension of 512 and GELU activations (Hendrycks & Gimpel, 2016) on CIFAR-100, ImageNet-1k, and COCOLogic. On CIFAR-100 and ImageNet, training uses a batch size of 4096 and a learning rate of 10^{-4} , with dropout (0.3), weight decay of 10^{-4} , and early stopping (patience 10). On COCOLogic, we instead train for 100 epochs with a batch size of 512, learning rate of 10^{-4} , and weight decay of 10^{-2} , using a learning-rate scheduler (plateau, patience 5, factor 10^{-3} , minimum learning rate 10^{-6}) and no early stopping. All pretraining is conducted without provers, and the resulting verifiers are used to initialize the CLIP-based NCV training.

E.2 NCV TRAINING

For the experiments presented in our main results in Tab. 1, we detail the training setup separately for CIFAR-100 and ImageNet.

Class Rules	Pixel-MAC	Neural Concept Verifier
Large gray cube and large cylinder		large and gray and cube
Small metal cube and small metal sphere		small and cube sphere
Large blue sphere and small yellow sphere		blue and sphere yellow and sphere

Figure 4: Comparison of explanations from NCV vs. Pixel-MAC for CLEVR-Hans3 images of all three classes. (a) Merlin–Arthur training on pixel space yields uninformative masks. (b) NCV provides clear explanations by highlighting object features corresponding to the class rule. The single-object images are reconstructions from the respective slots selected by Merlin (prover).

Table 7: Pretraining accuracy of the verifier (without provers) for CLIP-based NCV on CIFAR-100, COCOLogic and ImageNet-1k.

Dataset	Accuracy (%)
CIFAR-100	85.96
COCOLogic	81.39
ImageNet-1k	77.07

Model Architecture. The verifier (Arthur) is initialized as the pretrained two-layer multilayer perceptron (MLP) described above. Merlin and Morgana are implemented as independent neural networks, each parameterized by a two-layer MLP with hidden dimension 512 and ReLU activations. Both provers receive the full concept activation vector as input and output a sparse selection mask indicating the active concepts that are passed to the verifier.

CIFAR-100. For CIFAR-100, all components are trained jointly for 100 epochs using the Adam optimizer. We set the verifier learning rate to 10^{-4} , and use 5×10^{-4} for both Merlin and Morgana. A batch size of 256 is used throughout. Weight decay is set to 0.1, and a hard mask size of 32 concepts is enforced per input. To incentivize sparse masks, an L1 penalty of 0.1 is applied to the provers. A learning rate scheduler (plateau-based) is employed with a patience of 5, minimum learning rate of 10^{-6} , and decay factor of 0.001. Early stopping is disabled, and all results are averaged over 10 random seeds.

ImageNet. The ImageNet setup mirrors CIFAR-100 in most aspects. We again train for 100 epochs with a batch size of 256, using the same learning rates for verifier (10^{-4}) and provers (5×10^{-4}) , mask size of 32 features, and L1 penalty (0.1). Weight decay is reduced to 0.005 to improve generalization. The same learning rate scheduler and seed setup are used as in the CIFAR-100 experiments.

COCOLogic. Training on COCOLogic follows the CIFAR-100 configuration with minor adjustments: models are trained for 100 epochs with a batch size of 512, verifier learning rate of 10^{-4} , and prover learning rates of 5×10^{-4} . We use a weight decay of 0.01, a mask size of 32, and an L1 penalty of

Table 8: Ablation results for our method on CIFAR-100 with varying mask sizes. We report mean \pm std over 10 seeds.

Mask Size	Completeness Train	Completeness Validation	Soundness Train	Soundness Validation
4	$74.97 {\pm} 4.45$	$69.32 {\pm} 2.66$	$99.85 {\pm} 0.07$	99.85 ± 0.09
8	87.42 ± 2.60	$71.20 {\pm} 22.43$	$99.95 {\pm} 0.03$	$99.95 {\pm} 0.03$
16	$94.08 {\pm} 1.46$	$81.82 {\pm} 0.50$	$99.96 {\pm} 0.03$	$99.97 {\pm} 0.02$
64	$97.65 {\pm} 0.47$	$84.01 {\pm} 0.31$	$100.00 {\pm} 0.00$	$100.00 {\pm} 0.00$

Table 9: Ablation results for our method on ImageNet with varying mask sizes. We report mean \pm std over 10 seeds.

Mask Size	Completeness Train	Completeness Validation	Soundness Train	Soundness Validation
4	$59.30 {\pm} 0.34$	$55.96 {\pm} 0.30$	$99.81 {\pm} 0.03$	99.83 ± 0.06
8	$64.84 {\pm} 0.41$	$60.98 {\pm} 0.30$	$99.94 {\pm} 0.01$	$99.94 {\pm} 0.03$
16	$68.85 {\pm} 0.34$	$64.60 {\pm} 0.12$	99.96 ± 0.03	$99.96 {\pm} 0.03$
64	$73.35 {\pm} 0.39$	$69.03 {\pm} 0.18$	$99.97 {\pm0.01}$	$99.97 {\pm} 0.02$

0.1 on the provers. As with CIFAR-100, early stopping is disabled, and learning rate scheduling and seed averaging remain unchanged.