FROM RULES TO VISION: SYMBOLIC KNOWLEDGE INJECTION AND RULE EXTRACTION UNDER LIMITED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Neurosymbolic learning combines neural representation learning with explicit symbolic knowledge. Among its foundational approaches, rule-based knowledge injection offers a direct way to incorporate structured prior knowledge into a network's topology and initial weights. While this idea has shown value in other domains, its potential for modern image classification remains underexplored, although symbolic rules could provide useful prior structure when labelled data are scarce and support more inspectable models. We therefore adapt the classical knowledge-based artificial neural network (KBANN) framework to image classification and ask whether symbolic knowledge can compensate for very limited labelled data, and whether learned knowledge can later be extracted as readable rules. We implement the full injection-refinement-extraction lifecycle: predicates are grounded as deterministic functions of pixels, rules are translated into neural structure and initial weights, the network is refined through gradient learning, and rules are extracted after training. Controlled experiments separate four factors: rule source, predicate grounding, injection strength, and how the rule pathway is attached to the network. We also compare classical KBANN compilation with a signed projection that maps predicates directly to classes. With one labelled image per class, ten MNIST images in total, supplied digit rules raise balanced accuracy from 0.5022 for a plain CNN to 0.6183 with classical compilation and 0.6847 with the signed projection. Prior benefit tracks standalone rule strength ($r = 0.97$), while permuted rules underperform the plain CNN and rules induced from the same ten labels provide no comparable gain. After training, extracted rules preserve the injected structure and reproduce network predictions with fidelity of 0.9814–0.9903, but recover no new antecedents. Knowledge injection can therefore reduce dependence on labelled data and leave inspectable structure behind, while its effectiveness depends strongly on how that knowledge is represented and integrated.

1 INTRODUCTION

Deep image classifiers learn everything they know from pixels, and they need many labelled examples to do it. Neurosymbolic learning asks whether explicit symbolic knowledge can supply what labels cannot: fewer annotations for the same accuracy, a prediction a person can inspect, and a theory that learning can repair where it is wrong (Towell & Shavlik, 1994; Xu et al., 2018; Manhaeve et al., 2021). The integration of learning and reasoning has been described as the third wave of the field (d'Avila Garcez & Lamb, 2023) and as the defining problem of its current period (Kautz, 2022). Both promises matter in vision, where annotation is expensive and a trained model is hard to audit after the fact.

A substantial literature now combines logical knowledge with image classifiers. Constraints regularize a network with first-order knowledge during training (Roychowdhury et al., 2021), semantic loss places constraints in the objective (Xu et al., 2018), probabilistic logic programs make inference differentiable (Manhaeve et al., 2021), differentiable rule layers learn Boolean operators over learned features (Reimann et al., 2022), and concept bottleneck models route prediction through named concepts so a human can intervene on them (Koh et al., 2020). The survey of Nicolau et al. (2026)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

organizes more than sixty such methods into external solvers, differentiable approximations, probabilistic methods, graph methods, and logical constraints, and names perception–symbol integration, dependence on hand-written knowledge, benchmark scope, and reproducibility as the concerns recurring across all of them.

One long-standing family is missing from that taxonomy. Every surveyed method keeps symbolic knowledge *active* during learning or inference, inside an objective, a solver, a graph, or a reasoning layer. Rule-to-weight compilation does not: it translates an approximately correct propositional theory into topology and large signed initial weights, then releases those parameters to ordinary gradient descent (Towell & Shavlik, 1994). The knowledge is asserted once and never enforced again. Despite the appeal of that arrangement–a rule set consumes no labels, adds no inference-time cost, and can be read back out of the trained weights–rule–based knowledge injection appears in none of the vision methods the survey collects.

Other fields report clear benefits from exactly this mechanism. Hooshyar et al. (2024) apply the full injection–refinement–extraction lifecycle to educational data, and Responsible-DKT continues that program with grounded weighted relational rules (Hooshyar et al., 2026); earlier work examined how rule-based initialization affects convergence, accuracy, and network size (Andrews & Geva, 1999). What these settings share is that their variables arrive already named and already meaningful. In vision they do not. A rule such as "a zero has a closed loop" is written over named predicates while a convolutional network receives an array of pixels, and between the two lies a grounding step classical KBANN never had to take.

This study investigates whether the KBANN lifecycle transfers to images, keeping two questions in front of the rest. Can supplied symbolic knowledge compensate for very limited labelled data, the regime where a prior is most likely to be worth its cost? And can the knowledge the network holds after training be extracted as rules a person can read? Answering either requires separating four factors usually bundled into a single reported gain: *provenance*, whether a theory was written by hand or induced from the same labels the network trains on; *grounding*, what the predicates supply on their own before any rule is applied; *initialization scale*, the effect of large stamped weights independently of what they encode; and *attachment*, whether the symbolic path replaces the visual predictor, supplements it, or merely reads it.

Two comparisons recur in every table, so we name them here. The first asks how a theory should become weights. **Compilation** is the classical KBANN answer: each clause gets its own conjunctive unit and a class's clauses are combined through a disjunction, so the logical structure is literally present in the topology. The **signed projection** discards that structure and writes the same theory into one linear predicate-to-class map, $+1$ per positive antecedent and -1 per negated one. Both read the same predicates, start at the same logit scale, and stay equally trainable, so a difference between them is attributable to the representation rather than to the knowledge. The second asks what a rule effect is measured against. The **scrambled** arm keeps the topology and the stamp magnitude but permutes which predicate feeds which unit, holding everything fixed except the written predicate–class correspondences. We treat it as a floor rather than a neutral control: a permuted theory is actively misleading, scoring below a plain CNN given no rules at all, so correct-minus-scrambled adds the harm of wrong rules to the benefit of right ones. Content is quoted against the plain CNN throughout.

We contribute the following. (i) A complete visual KBANN pipeline with negation, disjunction, computed or learned predicates, and M-of-N extraction, in which supplied rules substitute for scarce supervision: at one image per class they improve balanced accuracy by 0.116 over a plain CNN once predicate access and head capacity are matched. (ii) Evidence that extraction after learning returns a readable reconstruction of the trained rule pathway at high predictive fidelity, with a precise statement of what that does and does not establish. (iii) Matched controls for predicate access, stamp scale, content, and attachment, which locate a cost in the compilation step itself that is measurable before training: clause and disjunction layers can discard more predicate information than correct content restores, and which representation wins is predicted by the two branches' standalone accuracy at initialization. The positive low-shot result is MNIST-specific and uses rules written before scoring but not under clean-room conditions; rules induced from the same ten labels do not reproduce it.

2 Related Work

From KBANN to contemporary knowledge injection. KBANN maps rules into network structure and large signed weights, refines the resulting network by backpropagation, and translates trained weights back into rules (Towell & Shavlik, 1994; 1993). RULEIN subsequently examined the effect of rule-based initialization on convergence, accuracy, and network size (Andrews & Geva, 1999). The educational adaptation of Hooshyar et al. (2024) is the implementation lineage reproduced here. Responsible-DKT instead represents weighted first-order rules in a grounded relational computation graph (Hooshyar et al., 2026); its rules remain part of the model computation, whereas a KBANN stamp may drift after initialization. This difference in the duration of symbolic influence motivates our controls.

Rules in neural vision. Constraint-based approaches keep first-order knowledge active during learning or inference and have been evaluated with CNNs (Roychowdhury et al., 2021); related mechanisms include rule distillation (Hu et al., 2016), semantic loss (Xu et al., 2018), DeepProbLog (Manhaeve et al., 2021), Logic Tensor Networks (Serafini & d'Avila Garcez, 2016), DL2 (Fischer et al., 2019), Deep Logic Models (Marra et al., 2020), and straight-through constraint injection (Yang et al., 2022). Neural Logic Rule Layers learn differentiable Boolean operators and can incorporate expert rules (Reimann et al., 2022), and Concept Logic Trees support user-editable transparent image classification (Rodríguez et al., 2024). Concept-RuleNet is closest to our hybrid in combining neural and rule-based scores, but obtains its concepts, rules, and verification from pretrained vision-language and language-model agents (Sinha et al., 2026). Our measured pixel predicates, from-scratch CNN, and trainable stamp address a different question: what remains when a classical compiler supplies the starting point and no continuing semantic constraint.

Concept-mediated prediction and semantic controls. Both representation capacity and intervention procedure affect what a concept bottleneck reveals (Koh et al., 2020; Espinosa Zarlenga et al., 2022; Shin et al., 2023). Deep Concept Reasoners construct differentiable rules over concept truth values (Barbiero et al., 2023), and Self-Explaining Neural Networks emphasize explicitness, faithfulness, and stability (Alvarez-Melis & Jaakkola, 2018). Recent work shows that concept prediction need not restrict a representation to concept information (Almudévar et al., 2026; Debot & Marra, 2025), and that consecutive linear maps can bypass the intended bottleneck (Tapli et al., 2026). Concept Flow Models use random concepts and hierarchies to test semantic utility (Wang & Paschke, 2026); our antecedent permutation follows the same evaluative principle and is not claimed as new, applying the control to the initial weights of a rule compiler.

3 Method

3.1 Rules as Initial Weights

We compile propositional rules into neural weights following KBANN (Towell & Shavlik, 1994). Let P_r and N_r be the positive and negated antecedents of clause r, and let $z_i \in [0, 1]$ denote a grounded predicate. The initialized clause unit is

$$h_r(z) = \sigma\left(\omega \sum_{i \in P_r} z_i - \omega \sum_{i \in N_r} z_i - \omega(|P_r| - \tfrac{1}{2})\right), \tag{1}$$

with $\omega = 4$. The threshold counts positive antecedents; counting every literal, as in the available educational implementation, makes clauses with negation systematically too difficult to activate. For class c, multiple clauses are combined as

$$g_c(z) = \max_{r \in \mathcal{R}_c} \{\omega h_r(z)\} - \omega/2. \tag{2}$$

The maximum avoids an arity-dependent soft-OR bias. All compiled parameters remain trainable, so these equations describe an initialization, not a constraint maintained during learning.

The *free* arm keeps the compiled topology with default random weights; the *scrambled* arm keeps topology and stamp scale while permuting antecedent connections, as defined in Section 1.

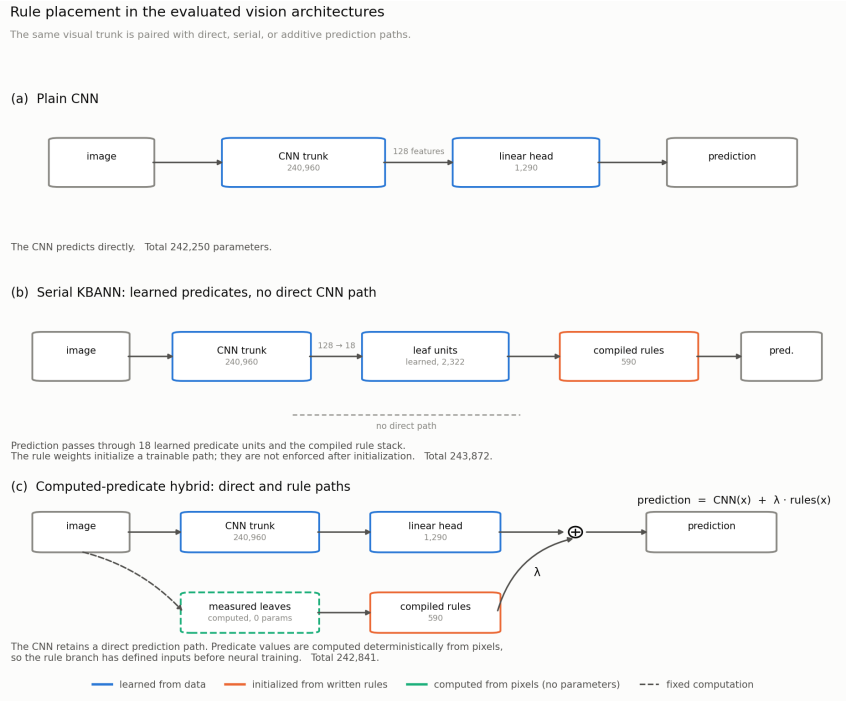

Figure 1: Rule placement in the principal architectures. Serial KBANN makes the learned predicate and rule stack the only route to the prediction. The hybrid retains a direct CNN route and adds a trainable rule branch over predicates computed from pixels. A stamp specifies initial weights; it does not enforce the rules after training.

3.2 Computed Predicates

We evaluate on MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and FER2013 (Goodfellow et al., 2013); CIFAR-10 (Krizhevsky, 2009) appears only in a negative imbalance experiment. Predicates are deterministic functions of pixels rather than human annotations: 22 binary measurements on MNIST spanning topology, bounding-box shape, ink distribution, symmetry, skeleton structure, run profiles, and strokes, with analogous statistics for the other datasets. Percentile thresholds are fitted on a 20,000-image pool and frozen for test, which needs no concept labels but does use additional unlabeled images (Appendix A.8).

We distinguish two rule provenances. *Induced* rules are greedy DNF fits to labeled predicate vectors, so every label used to select them counts against the supervision budget. *Supplied* MNIST rules encode familiar digit-shape statements, were fixed before scoring, and consumed no labels; they are not a clean-room artifact, since their author had previously seen a predicate-prevalence table. We treat that result as controlled but dataset-specific evidence rather than independent expert validation.

3.3 Vocabulary Probe

The vocabulary probe is multinomial logistic regression over the full measured vocabulary with balanced class weights, fitted on approximately 1,000 labels and evaluated on the test split by balanced accuracy and macro one-vs-rest ROC-AUC. These are achieved classifier scores, not information-theoretic ceilings or label-free diagnostics (Appendix A.5).

3.4 Attachment Modes

We evaluate five attachments. In *serial* KBANN the rule stack is the only route to the class output; a *residual* branch adds to CNN logits; a *parallel* branch reads detached CNN features and cannot alter

Table 1: MNIST balanced accuracy, mean (sample SD) over ten seeds, with supplied rules. Balanced-accuracy chance is 0.100 on this ten-way task. CNN+PRED receives the same computed predicates through a learned MLP; PRED uses only those predicates. RESNET-18 is trained from scratch on the same seed-specific subsets under the same update schedule; it is reported so that backbone capacity can be read off the same protocol, and it is not the comparator the rule claims rest on.

Images/class	CNN	ResNet-18	CNN+Pred	Pred only	Hybrid	Scrambled
1	0.502 (0.047)	0.207 (0.049)	0.504 (0.045)	0.461 (0.078)	**0.618 (0.039)**	0.406 (0.058)
2	0.650 (0.037)	0.279 (0.050)	0.642 (0.042)	0.562 (0.036)	**0.710 (0.019)**	0.569 (0.044)
3	0.732 (0.027)	0.372 (0.042)	0.720 (0.040)	0.593 (0.035)	**0.757 (0.021)**	0.644 (0.037)
5	0.801 (0.027)	0.479 (0.050)	0.799 (0.021)	0.640 (0.018)	**0.810 (0.019)**	0.739 (0.029)
10	0.885 (0.011)	0.623 (0.034)	**0.888 (0.010)**	0.683 (0.011)	0.879 (0.009)	0.849 (0.012)
100	0.970 (0.002)	0.966 (0.003)	**0.971 (0.002)**	0.759 (0.004)	0.967 (0.002)	0.963 (0.003)

CNN training. The *hybrid* computes

$$f_{\mathrm{hyb}}(x) = f_{\mathrm{cnn}}(x) + \lambda g(z(x)), \tag{3}$$

where g starts from the compiled rules and λ is trainable. The *stamped CBM* keeps the same image–predicate–linear-class topology as its CBM control and changes only the initial class-head direction, asking whether a signed normalized rule projection is a useful prior without the disjunction layer.

Unless stated otherwise the image encoder has four Conv–BatchNorm–ReLU–pool blocks of widths 32, 64, 128, and 128 followed by adaptive pooling, trained from scratch with Adam at learning rate 10^{-3} and batch size 128. Low-shot runs use seeds 42–51, resample a class-balanced subset per seed, share it across arms, and receive at least 300 updates; serial and residual experiments use approximately 1,000 labels and 30 epochs. We report balanced accuracy, sample standard deviations, and paired tests, with p-values descriptive where several budgets or contrasts were inspected.

4 EXPERIMENTS

4.1 SUPPLIED RULES HELP ONLY IN THE EXTREME LOW-SHOT REGIME

Table 1 gives the equal-neural-label experiment. Budgets are quoted as images per class throughout, so one image per class is ten labels on this ten-way task and a hundred is $1{,}000$. The plain CNN, feature control, and hybrid train on the same seed-specific images. The supplied rules used no labels during their construction, subject to the prior-exposure caveat in Section 3; the hybrid evaluates them over deterministic predicates. At one image per class, it improves balanced accuracy from 0.5022 to 0.6183, a paired difference of 0.1161 with a nominal 95% confidence interval of $[0.0936, 0.1386]$. The advantage narrows to 0.0601 at two images per class and changes sign by ten. In this setting, prior information substitutes for a small number of labels; it does not improve the mature classifier. A ResNet-18 trained from scratch on the same subsets stays far below the compact CNN until a hundred images per class, and the computed predicates alone outscore it at every budget below that, so the low-shot baseline is not an artifact of an undersized backbone.

The matched-feature control sharpens attribution within this experiment. Giving a non-rule CNN the same predicates changes the one-shot mean by only $+0.0016$, while the hybrid is $+0.1145$ higher than that control (nominal paired 95% CI $[0.0893, 0.1397]$). Head capacity does not explain it either: a parameter count alone would not settle this, since at ten labels a larger head is a harder estimation problem rather than a stronger control, so we measure it. Replacing the control's 10,058-parameter nonlinear head with a 1,470-parameter linear one moves the one-shot mean by $+0.0004$ ($p = 0.95$). A scrambled-rule branch reaches 0.4064, which is 0.0958 *below* the plain CNN; for the reason given in Section 1 we read content against the CNN rather than against this floor. The rules alone score 0.4421, and rules induced from the same ten labels score 0.4981, against 0.5022 for the CNN. The positive result therefore uses information beyond the ten labels and is not reproduced by our same-budget induction procedure; Appendix A.8 accounts for every label each arm consumes.

Table 2: Signed projection minus compiled stack, balanced accuracy, mean over ten seeds. **Positive means compiling costs accuracy.** All arms share the trunk, the computed predicates, the label budget, and the branch logit scale (≈ 1.37). "Solo" is each branch's own balanced accuracy before training. Hand-written rules exist only for MNIST.

Rules	Dataset	Solo compiled	Solo signed	1 img/cls	2	5	100
Hand-written	MNIST	0.4421	**0.5004**	$+0.0664$	$+0.0469$	$+0.0313$	$+0.0022$
Induced	MNIST	**0.5979**	0.3988	-0.0422	-0.0151	$+0.0040$	$+0.0049$
Induced	Fashion	**0.4307**	0.3345	-0.0204	$+0.0044$	$+0.0112$	$+0.0060$

4.2 Compiling the Theory Is a Representation Choice

The experiment above shows that a theory helps, not that *compiling* it helps. KBANN's contribution is a specific representation: clauses become conjunctive units, multiple clauses per class become a disjunction, and the whole stack is stamped. We isolate that choice by injecting the same theory two ways. The compiled branch is Equations 1–2. The *signed projection* is a single linear map $W[i, c]$ that adds $+1$ for each positive occurrence of predicate i in a clause with head c and -1 for each negated one, with no clause layer and no disjunction. Both are trainable, both read the same computed predicates, and both take no gradient into the trunk. We normalize each class column to a fixed L_2 norm so the two branches start at comparable logit scale; Appendix A.9 sweeps that norm.

Table 2 reports the difference. It changes sign with rule provenance. For the supplied hand-written theory, compiling costs 0.0664 at one image per class and remains negative at every budget through $1{,}000$ labels. For rules induced by greedy DNF search, compiling pays at extreme low shot and the advantage reverses by five images per class.

The mechanism is visible before training. Our induction procedure selects the conjunction with the best F_1, then the one that most improves the F_1 of the disjunction of the two, so an induced theory arrives with its disjunctive structure already fitted while a hand-written theory carries no calibrated disjunction. Let solo be a branch's balanced accuracy on its own, before any gradient step. The sign of $\mathrm{solo}(\text{signed}) - \mathrm{solo}(\text{compiled})$ agrees with the sign of the trained difference in all three cells–an observed agreement, not a validated predictor.

Permuted rules are an adversarial control, not a neutral one. We permute which predicate carries which weight, holding each class column's multiset of weights and its norm fixed. The branch drops to 0.1375 balanced accuracy against a 0.100 chance rate, and the full model falls 0.0417 below the plain CNN ($p = 6.0 \times 10^{-4}$). The same holds in all three cells, so a contrast against this arm sums two effects of opposite sign. We also relocate a controlled fraction of the theory's antecedents, holding each unit's arity and positive antecedent count invariant. Degradation is monotone, and a prior's benefit tracks its standalone strength across those doses ($r = 0.97$, $p = 0.006$, $n = 5$). The correct theory lies on that relationship rather than above it. Permuting only the clause-to-class assignment, which preserves every clause intact, lands 0.055 below it. We therefore treat permutation arms as floors.

The measured vocabulary bounds what any theory over it can express and differs sharply across datasets (0.7865 probe balanced accuracy on MNIST against 0.2288 on FER2013); we treat this as a moderating condition on every result below, not a contribution, and report it in Appendix A.5. The compact four-block CNN is not a weak comparator either: under the same schedule it beats ResNet-18 at one-shot MNIST and in matched $1{,}000$-label runs on both datasets (Appendix A.6).

4.3 An Architecture-Matched Concept-Bottleneck Comparison

The same representational cost appears at a third attachment, and against a published baseline rather than an ablation of our own. The serial rescue of Section 4.4 is measured against random initialization of the same compiled topology, which leaves open whether that topology is worth having at all. We therefore add an architecture-matched concept bottleneck: identical trunk, identical measured predicates, identical auxiliary predicate loss, and a single linear predicate-to-class layer in place of the compiled clause and disjunction layers. This is Koh's joint bottleneck (Koh et al., 2020) over our predicates, and it is not stamped.

Table 3: Architecture-matched comparison at approximately 1,000 labels, balanced accuracy as mean (sample SD) over ten seeds. All arms share the trunk, the measured predicates, and the auxiliary predicate loss. CBM replaces the compiled clause and disjunction layers with one linear layer and is not stamped; FREE is the compiled topology left unstamped.

Dataset	Plain CNN	CBM	Serial KBANN	Scrambled	Free
MNIST	**0.9633 (0.0024)**	0.9396 (0.0089)	0.8981 (0.0121)	0.5702 (0.0993)	0.1002 (0.0004)
Fashion-MNIST	**0.8359 (0.0024)**	0.7999 (0.0059)	0.7184 (0.0223)	0.4379 (0.0778)	0.1006 (0.0019)
FER2013	**0.2554 (0.0400)**	0.2410 (0.0319)	0.2278 (0.0236)	0.1824 (0.0173)	0.1464 (0.0075)

Table 4: The same initialization has opposite effects under different attachments. Values are balanced accuracy, mean (sample SD) over ten seeds. "Random" denotes an unstamped branch with the same topology.

Attachment	Dataset	Random	Scrambled	Correct
Serial	MNIST	0.1002 (0.0004)	0.5702 (0.0993)	**0.8981 (0.0121)**
Serial	Fashion	0.1006 (0.0019)	0.4379 (0.0778)	**0.7184 (0.0223)**
Residual	MNIST	**0.9626 (0.0023)**	0.9594 (0.0035)	0.9532 (0.0080)
Residual	Fashion	**0.8353 (0.0021)**	0.8308 (0.0024)	0.8278 (0.0032)

Table 3 reports the comparison. The bottleneck exceeds the stamped serial stack by 0.0415 on MNIST $p = 1.3 \times 10^{-7}$) and 0.0815 on Fashion-MNIST ($p = 4.5 \times 10^{-7}$); on FER2013 the difference is 0.0131 and not significant ($p = 0.31$), consistent with a vocabulary whose probe barely exceeds chance. The bottleneck uses fewer parameters on the class path than the compiled stack (180 against 570), so the difference is not capacity. Stamping the bottleneck's own head, rather than replacing it with the compiled stack, improves slightly over random initialization but does not distinguish correct from scrambled rules ($p = 0.575$); Appendix A.4 reports that experiment.

Read with the free and scrambled columns, this decomposes the serial result. Relative to the plain CNN, routing prediction through the predicate bottleneck costs 0.0236 on MNIST, replacing that bottleneck's single layer with the compiled clause and disjunction layers costs a further 0.0415, and leaving those layers unstamped costs the remaining 0.7979. The stamp rescues a difficulty the compiled topology introduces; it does not recover the accuracy the simpler bottleneck already reached over the same predicates.

4.4 SERIAL STAMPING RESCUES A DIFFICULT OPTIMIZATION

The serial symbolic stack shows how sharply a stamp can improve over the same architecture with random rule-layer weights. With 1,000 labels, the unstamped stack remains at chance: 0.1002 balanced accuracy on MNIST and 0.1006 on Fashion-MNIST. Stamping the correct rules raises performance to 0.8981 and 0.7184 respectively. Scrambled rules, which preserve the large weight scale while corrupting semantic content, reach 0.5702 and 0.4379.

The scrambled arm recovers a substantial part of the gap while carrying no intended class–predicate correspondence, which is consistent with scale and topology making the serial sigmoid stack easier to optimize. We do not attribute the effect solely to gradient scale, because the present diagnostic does not separate class-loss from auxiliary predicate-loss gradients. Either way the comparison shows a constrained architecture being rescued, not superiority to a plain CNN or CBM.

4.5 ATTACHMENT AND RULE QUALITY BOUND THE GAIN

The same stamp behaves differently when a CNN already has a direct route from pixels to classes. Table 4 contrasts the serial and residual cases. On MNIST, residual stamping reduces the mean from 0.9626 to 0.9532 (paired $p = 0.0043$); on Fashion-MNIST, it reduces the mean from 0.8353 to 0.8278 ($p = 2.4 \times 10^{-4}$). Scrambling is also harmful, but less so. This suggests that a persistent, unannealed branch can disturb an already effective predictor.

Improving symbolic fit does not solve this. Validation selection expands the MNIST theory from two two-literal clauses per class to three three-literal ones, raising rules-only test balanced accuracy from 0.5979 to 0.6900, yet learned serial KBANN moves from 0.8981 to 0.8907. The refined theory compiles to a larger stack and is stored as a separate result set, so the two are never pooled.

Readout and extraction. A rule branch reading detached CNN features agrees with the CNN on 0.9698 of MNIST test predictions against 0.6862 for a scrambled control, and the M-of-N procedure of Towell & Shavlik (1993) recovers the injected first-layer edges with retention 1.0000 and clustered-network fidelity 0.9814–0.9903. Both are content-dependent, and neither is a faithfulness result: the branch is trained on labels, not on CNN decisions, and extraction returns the stamped topology with no new antecedents recovered. Appendix A.7 gives both in full, together with a negative class-imbalance study.

5 Discussion and Conclusions

Symbolic knowledge helped where it was most needed and least expensive. With one labelled image per class, a supplied digit theory raised balanced accuracy from 0.5022 to 0.6183, while the advantage disappeared by ten images per class. This suggests that the prior mainly substitutes for a small number of labels rather than improving a classifier once sufficient supervision is available. The way the theory was represented also mattered. The same knowledge performed better as a flat signed projection, reaching 0.6847, than when compiled into clause and disjunction layers, indicating that representing a theory through additional logical structure can introduce a cost of its own. Architecture further shaped the effect. The same stamp that raised a serial symbolic stack from chance to 0.8981 degraded a residual CNN that already predicted well. After training, extraction recovered the injected edges with predictive fidelity between 0.9814 and 0.9903, but recovered no new antecedents. The knowledge therefore transferred, although its compilation did not always help, and what was extracted was a reconstruction of the injected structure rather than newly discovered symbolic knowledge.

This extends classical KBANN to a setting that differs fundamentally from the named variables on which the original framework was developed (Towell & Shavlik, 1994; Hooshyar et al., 2024). In vision, symbolic predicates must first be grounded in pixels, and this grounding stage can determine whether the rules contain usable information at all. A probe over the full predicate vocabulary reaches 0.7865 balanced accuracy on MNIST but only 0.2288 on FER2013, meaning that no theory expressed through the latter vocabulary can contain more class information than its predicates make available. This connects directly to concerns identified by Nicolau et al. (2026), including perception–symbol integration, hand-written knowledge, benchmark scope, and reproducibility. Rather than treating these only as caveats, our experiments separate them into factors within a common injection–refinement–extraction lifecycle.

The role of symbolic knowledge here also differs from contemporary constraint- and reasoning-based systems. Such approaches keep symbolic structure active in the objective or forward computation during learning (Xu et al., 2018; Roychowdhury et al., 2021; Manhaeve et al., 2021; Reimann et al., 2022). A KBANN stamp, in contrast, is released to gradient descent after initialization, and nothing subsequently enforces that structure. For this reason, initialization with rules should not by itself be interpreted as continued reasoning with rules; what remains after training has to be examined explicitly.

The results across label budgets show why this distinction matters. A prior competes with the information already available from labelled examples. At ten total labels, the supplied theory carries more class information than the CNN can estimate from images alone. By one hundred labels per class, however, the CNN has learned better statistics than the theory encodes, and the rule branch becomes a constraint on an already adequate predictor. Architecture dependence follows the same logic. A serial sigmoid stack with small random weights cannot train successfully, so stamping primarily provides useful optimization structure. This explains why even scrambled rules recover much of the gap, improving performance from 0.1002 to 0.5702. A residual CNN already provides an effective path from pixels to classes, so a persistent, unannealed symbolic branch can instead perturb a representation that already works. The same nominal theory can therefore act as semantic

information, optimization structure, or harmful bias depending on what the underlying architecture already provides.

Improving the symbolic theory does not necessarily resolve this tension. Expanding the MNIST rules increased rules-only accuracy from 0.5979 to 0.6900, yet the learned serial KBANN decreased slightly from 0.8981 to 0.8907. Symbolic fit and neural utility are therefore not equivalent objectives. This is also visible in the comparison between classical compilation and direct signed projection. Compilation preserves logical structure in the network topology, which can be useful when that structure has itself been fitted. An induced theory arrives with its disjunction already optimized, so the compiler implements a combination that induction has selected, explaining why compilation can help induced rules in the extreme low-shot setting. A hand-written theory, however, does not contain a calibrated disjunction, and in that case the clause and disjunction layers can discard predicate information faster than correct rule content restores it.

This difference is visible even before neural training. The standalone accuracies of the two branches, measured on held-out predicates before any gradient step, order the two representations in the same direction as the trained outcome in all three provenance and dataset cells we examine. Three cells constitute an observed agreement rather than a validated predictor, but the measurement is inexpensive and the design decision it informs is consequential. A practitioner holding a domain theory can therefore score it both ways and choose a representation before committing to a training run. The architecture-matched concept bottleneck supports the same reading against a published baseline rather than an ablation of our own. This is not a defeat by an older method, since a bottleneck is a standing architectural constraint while a stamp is only an initialization, and the two answer different questions. Its role here is to keep the claim falsifiable, because without it the improvement of a serial stack from 0.1002 to 0.8981 would read as an unqualified success.

The extraction results should be read with similar care. Named predicates make a pathway inspectable, extraction makes the learned weights expressible, and predictive agreement checks that the recovered rules reproduce the model's outputs. All three succeed here, and both measurements depend on rule content rather than on the procedure alone. None of them, however, establishes faithfulness. The rule branch is trained on labels rather than on the decisions of the CNN, and extraction returns the stamped topology without proposing any amendment to it. What the results support is reconstructibility, in the sense that the trained rule pathway can be written down and read again. A causal claim would additionally require interventions whose effects follow the extracted program, together with evidence that the named predicates mean what they claim on target-domain images. Interpretability in this setting is therefore a sequence of tests, and the present work passes the earlier ones.

Taken together, these results indicate when symbolic knowledge injection is worth its cost in vision. Where a reliable domain theory is available and labelled examples are very scarce, injecting that theory is beneficial, and the simplest representation is often the most effective one. The benefit should be expected to fade as labels accumulate, and to depend on what the architecture can already do without assistance, since a stamp is most valuable where a network cannot otherwise train and least valuable where it already performs well. Where the objective is intervention or a standing guarantee rather than a useful starting point, a concept bottleneck remains the more appropriate design, because an initialization stops constraining a model as soon as training begins. Porting KBANN from named tabular variables to images is thus not a mechanical substitution of inputs. The vision model must define predicates, ground them in pixels, preserve their information through the logical operators, and attach the resulting computation without disabling a neural path that already works. Under those conditions, symbolic knowledge does what neurosymbolic learning promises: it substitutes for scarce supervision and leaves behind a structure that can be read back out at high fidelity. Neurosymbolic vision should therefore evaluate rule provenance, predicate grounding, content controls, attachment, and intervention faithfulness as separate questions, rather than infer knowledge transfer from the presence of symbolic structure.

5.1 Limitations

The strongest equal-budget content result is obtained on MNIST alone. The supplied rules were fixed before scoring and consumed no labels, but their author had previously inspected a predicate-prevalence table, so clean-room rule writing remains untested, and rules induced from the same ten

labels score 0.4981 against 0.5022 for the plain CNN. Fashion-MNIST and FER2013 replicate the attachment, vocabulary, and backbone trends, yet their low-shot rules were induced from additional labels, so the study does not establish general label efficiency.

Computed predicates avoid manual concept annotation, but they require hand-designed feature functions and an unlabeled calibration pool, which suits simple shapes and is weak for natural images. Learned soft predicates can also carry information beyond their names, and our measurements test neither concept interventions, nor minimality, nor causal faithfulness. The claim of full transparency therefore remains open until interventions change predictions in accordance with the extracted rules.

The architecture sweep is diagnostic rather than exhaustive. The residual coefficient is not annealed, the stamped concept-bottleneck head is an additive projection rather than an exact executor of the disjunctive normal form, and rule refinement explores only a small grid. Nominal p-values are not adjusted across all budgets and contrasts inspected. Finally, the measured cost of compilation is consistent with the disuse of the method in vision, but it does not explain that disuse, since compute, data scale, and end-to-end training all changed at the same time.

AI USE STATEMENT

Generative AI tools assisted literature discovery, code navigation and implementation, experimental design, debugging, statistical checks, result interpretation, figure preparation, and manuscript drafting and revision. They were not used to generate synthetic datasets, formulate or prove theoretical claims, or translate text; other required-disclosure categories are not applicable. Numerical claims are linked to saved experiment outputs, and bibliography entries were checked against publisher or archival metadata. The authors reviewed the AI-assisted work and take responsibility for the final text, claims, code, and artifacts.

ETHICS STATEMENT

This work uses standard image-classification benchmarks and computed predicates. The main ethical risk is not direct deployment harm but over-interpretation: rule-based explanations can appear more faithful or more general than the underlying vocabulary permits. The paper therefore reports scrambled controls, fidelity separately from accuracy, and limitations of the predicate vocabulary.

REPRODUCIBILITY STATEMENT

The experiments use balanced accuracy, fixed seeds, deterministic training where recorded, and post-correction results only. The experiment code and saved outputs record the commands and files associated with each claim; the appendix gives rule provenance, the predicate vocabulary, hyperparameters, seed protocol, extraction definitions, and an artifact map.

REFERENCES

Antonio Almudévar, José Miguel Hernández-Lobato, and Alfonso Ortega. There was never a bottleneck in concept bottleneck models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Fy7V5dalvX.

David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html.

Robert Andrews and Shlomo Geva. On the effects of initialising a neural network with prior knowledge. In *Proceedings of the 6th International Conference on Neural Information Processing*, volume 1, pp. 251–256, 1999. doi: 10.1109/ICONIP.1999.843995.

Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Mateo Espinosa Zarlenga, Lucie Charlotte Magister, Alberto Tonda, Pietro Lió, Frederic Precioso, Mateja Jamnik, and Giuseppe Marra. Interpretable neural-symbolic concept reasoning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1801–1825, 2023. URL https://proceedings.mlr.press/v202/barbiero23a.html.

Artur d'Avila Garcez and Luís C. Lamb. Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

David Debot and Giuseppe Marra. Quantifying the accuracy-interpretability trade-off in concept-based sidechannel models. *arXiv preprint arXiv:2510.05670*, 2025.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/867c06823281e506e8059f5c13a57f75-Abstract-Conference.html.

Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin Vechev. DL2: Training and querying neural networks with logic. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1931–1941, 2019.

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. *arXiv preprint arXiv:1307.0414*, 2013.

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*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Danial Hooshyar, Roger Azevedo, and Yeongwook Yang. Augmenting deep neural networks with symbolic educational knowledge: Towards trustworthy and interpretable AI for education. *Machine Learning and Knowledge Extraction*, 6(1):593–618, 2024. doi: 10.3390/make6010028.

Danial Hooshyar, Gustav Šír, Yeongwook Yang, Tommi Kärkkäinen, Raija Hämäläinen, Ekaterina Krivich, Mutlu Cukurova, Dragan Gašević, and Roger Azevedo. Neural-symbolic knowledge tracing: Injecting educational knowledge into deep learning for responsible learner modelling. *arXiv preprint arXiv:2604.08263*, 2026.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2410–2420. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1228. URL https://aclanthology.org/P16-1228/.

Henry Kautz. The third AI summer: AAAI robert s. engelmore memorial lecture. *AI Magazine*, 43 (1):105–125, 2022.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. Technical report; no DOI or arXiv record exists for this reference.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence*, 298:103504, 2021. doi: 10.1016/j.artint.2021.103504.

Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Integrating learning and reasoning with deep logic models. In *Machine Learning and Knowledge Discovery in Databases*, pp. 517–532. Springer International Publishing, 2020. doi: 10.1007/978-3-030-46147-8_31.

Marcio Nicolau, Claudio R. Jung, and Luis C. Lamb. Neurosymbolic computer vision: a survey and perspective. *Neural Computing and Applications*, 38:574, 2026. doi: 10.1007/s00521-026-12063-6.

Jan Niclas Reimann, Andreas Schwung, and Steven X. Ding. Neural logic rule layers. *Information Sciences*, 596:185–201, 2022. doi: 10.1016/j.ins.2022.03.021.

David M. Rodríguez, Manuel P. Cuéllar, and Diego P. Morales. Concept logic trees: Enabling user interaction for transparent image classification and human-in-the-loop learning. *Applied Intelligence*, 54(5):3667–3679, 2024. doi: 10.1007/s10489-024-05321-4.

Soumali Roychowdhury, Michelangelo Diligenti, and Marco Gori. Regularizing deep networks with prior knowledge: A constraint-based approach. *Knowledge-Based Systems*, 222:106989, 2021. doi: 10.1016/j.knosys.2021.106989.

Luciano Serafini and Artur S. d'Avila Garcez. Learning and reasoning with logic tensor networks. In *AI*IA 2016 Advances in Artificial Intelligence*, pp. 334–348. Springer International Publishing, 2016. doi: 10.1007/978-3-319-49130-1_25.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31504–31520, 2023. URL https://proceedings.mlr.press/v202/shin23a.html.

Sanchit Sinha, Guangzhi Xiong, Zhenghao He, and Aidong Zhang. Concept-RuleNet: Grounded multi-agent neurosymbolic reasoning in vision language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29678–29686, 2026. doi: 10.1609/aaai.v40i35.40212.

Merve Tapli, Quentin Bouniot, Wolfgang Stammer, Zeynep Akata, and Emre Akbas. Rethinking concept bottleneck models: From pitfalls to solutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9901–9910, 2026.

Geoffrey G. Towell and Jude W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13(1):71–101, 1993. doi: 10.1023/a:1022683529158.

Geoffrey G. Towell and Jude W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1–2):119–165, 1994. doi: 10.1016/0004-3702(94)90105-8.

Ya Wang and Adrian Paschke. Concept flow models: Anchoring concept-based reasoning with hierarchical bottlenecks. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=TNYLf65I3I.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5502–5511, 2018. URL https://proceedings.mlr.press/v80/xu18h.html.

Zhun Yang, Joohyung Lee, and Chiyoun Park. Injecting logical constraints into neural networks via straight-through estimators. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25096–25122, 2022.

A Additional Experimental Details

A.1 Data, Predicates, and Supervision

MNIST and Fashion-MNIST use their standard 60,000/10,000 train/test splits. FER2013 uses 28,709 training images and the combined 7,178-image public/private test split; its training classes range from 436 examples for disgust to 7,215 for happy. For every dataset, predicate thresholds are fitted once on a random 20,000-image training pool (or the available pool when smaller) and reused

unchanged at test time. Low-shot neural subsets are sampled from this pool. Thus the images used for calibration are not additional labeled examples, but they are additional unlabeled inputs available to the method. The positive MNIST experiment uses supplied rules and does not use labels to fit them; induced-rule experiments use class labels explicitly and are reported separately.

The standardized vocabulary diagnostic fits on approximately 1,000 labeled predicate vectors. It reports both balanced accuracy and macro one-vs-rest class ROC-AUC, so FER2013's imbalance cannot be hidden by majority-class accuracy. The FER2013 vocabulary combines 19 band-statistic predicates with 19 coarse shape predicates. These measurements are less class-aligned than MNIST topology: they do not explicitly encode facial action units, landmarks, or localized muscle movement, and the experiment therefore tests a substantially weaker grounding interface despite using more predicates.

The 22 MNIST predicates, in their fixed array order, are: `has_closed_loop`, `has_two_loops`, `hole_upper`, `hole_lower`, `hole_big`, `is_narrow`, `is_wide`, `low_ink`, `high_ink`, `top_heavy`, `bottom_heavy`, `lr_symmetric`, `tb_symmetric`, `few_endpoints`, `many_endpoints`, `has_junction`, `top_runs2`, `mid_runs2`, `mid_runs3`, `bot_runs2`, `has_top_bar`, and `has_left_edge`. Their implementation thresholds grayscale ink, computes connected holes and skeleton statistics, and derives shape and mass measurements inside the foreground bounding box. The source file gives every equation and fitted percentile.

A.2 Supplied MNIST Theory

The supplied theory has two clauses per class and uses 18 predicates. It was written before evaluation but, as stated in the main text, not under clean-room conditions. The complete theory is:

```
d0 :- hole_big, few_endpoints.
d0 :- has_closed_loop, not has_junction.
d1 :- is_narrow, not has_closed_loop.
d1 :- low_ink, not has_junction.
d2 :- is_wide, bottom_heavy.
d2 :- bottom_heavy, not has_closed_loop.
d3 :- many_endpoints, not has_closed_loop.
d3 :- not has_left_edge, not has_closed_loop.
d4 :- has_junction, not has_closed_loop.
d4 :- many_endpoints, mid_runs2.
d5 :- has_top_bar, bottom_heavy.
d5 :- has_top_bar, not has_closed_loop.
d6 :- hole_lower, not hole_upper.
d6 :- has_closed_loop, bottom_heavy.
d7 :- has_top_bar, top_heavy.
d7 :- not has_closed_loop, not bot_runs2.
d8 :- hole_upper, hole_lower.
d8 :- has_two_loops, lr_symmetric.
d9 :- hole_upper, not hole_lower.
d9 :- has_closed_loop, top_heavy.
```

A.3 Training and Statistical Protocol

The low-shot MNIST experiments use seeds 42–51 and independently sample the same number of images from each class for every seed. All compared arms receive the same seed-specific subset. Batch size is 128 and the epoch count is chosen to provide at least 300 optimizer updates. Rule-only evaluation applies compiled clauses to computed test predicates without neural training.

The serial experiments use 1,001 stratified training images, 30 epochs, binary cross-entropy for class outputs, and auxiliary binary cross-entropy on measured predicates. CNN, residual, hybrid, and CBM predictions use softmax cross-entropy; learned-predicate variants additionally optimize predicate binary cross-entropy. The free, scrambled, and KBANN serial arms share the convolutional trunk, predicate width, compiled layers, optimizer, and data. Scrambling permutes antecedent connections independently by seed while preserving weight values and layer dimensions.

Table 5: Architecture-matched MNIST CBMs, mean ± sample SD over ten seeds. The 10-label row uses supplied rules; the 1,000-label row uses refined induced rules and is a separate experiment.

Labels	CBM	Stamped	Scrambled	Prior
10	0.4062 ± 0.0433	0.4384 ± 0.0443	0.4288 ± 0.0330	0.4386 ± 0.0436
1,000	0.9431 ± 0.0065	0.9456 ± 0.0065	0.9438 ± 0.0075	0.9452 ± 0.0038

Table 6: Dataset characteristics and full-vocabulary probes. All images are grayscale. The probes fit multinomial logistic regression with balanced class weights on approximately 1,000 labels; AUC is macro one-vs-rest class ROC-AUC. "Majority" is the largest class share, a plain-accuracy floor; "BA chance" is $1/k$, the floor for the balanced accuracy actually reported. "Gain" is exploratory one-shot balanced-accuracy gain from rules induced with 20,000 labels, not an equal-budget comparison.

Dataset	Image type	Classes	Majority	BA chance	Predicates	Probe BA	AUC	Gain
MNIST	28×28 digits	10	0.100	0.100	22	0.7865	0.9701	+0.226
Fashion-MNIST	28×28 clothing	10	0.100	0.100	22	0.5785	0.9132	+0.061
FER2013	48×48 expressions	7	0.247	0.143	38	0.2288	0.6011	+0.021

The architecture-matched CBM experiment uses all 22 predicates; at ten labels it uses 300 one-batch epochs, and at 1,000 labels it uses 30 epochs. The ResNet-18 control uses the standard small-image adaptation: its first convolution is 3×3 with stride one, the initial max-pool is removed, and all weights are randomly initialized. Runs use deterministic PyTorch operations and no pretrained weights.

Balanced accuracy is the unweighted mean of per-class recall. Low-shot uncertainty is computed from paired seed-wise differences because the sampled subset and model seed are shared across arms. The one-shot handwritten-rule interval in the main text uses a Student-t interval with nine degrees of freedom. The complete rule files, seed-level rows, result provenance, and figure-generation commands are included in the accompanying repository.

A.4 Architecture Matching Does Not Resolve a Content Effect

The low-shot hybrid has a direct CNN path, whereas a strict CBM predicts only through learned predicate logits (Koh et al., 2020). We therefore match the CBM topology and stamp only its predicate-to-class head. Table 5 shows a small one-shot improvement over the random CBM ($+0.0322$, paired $p = 0.0197$ nominal, which does not survive correction for the arms and budgets inspected here), and the experiment does not distinguish correct from scrambled stamps ($+0.0095$, $p = 0.5749$). A decaying penalty toward the initial rule matrix gives a similar mean. The improvement over random initialization therefore cannot be attributed specifically to rule semantics. A separate 1,000-label run with data-refined rules likewise does not distinguish the conditions. Because its rule source differs, we do not present the two rows as a learning curve.

A.5 Dataset and Vocabulary Determine the Available Signal

Table 6 makes the cross-dataset boundary explicit. Class count alone does not determine difficulty: FER2013 has seven classes rather than ten, yet its low-resolution natural faces require distinctions among subtle expressions, its test split is imbalanced, and global brightness, edge, region, and coarse shape predicates carry little class information. Its larger 38-predicate vocabulary reaches only 0.2288 balanced accuracy and 0.6011 macro one-vs-rest ROC-AUC. MNIST's topological predicates are much better aligned with digit identity; Fashion-MNIST lies between these cases. Thus a weak FER2013 result is evidence about this vocabulary–dataset interface, not evidence that adding more predicates or compiling rules can overcome a poorly expressed task.

The probe and induced-rule gain have the same ordering, but this three-point observation is descriptive. Dataset, vocabulary, and rule provenance vary together; the induction budget invalidates

Table 7: Scratch-backbone benchmark, balanced accuracy as mean (sample SD) over ten seeds. "Pred. aux." denotes the matched auxiliary predicate loss used by both backbones in the 1,000-label protocol.

Dataset	Protocol	Labels	Compact CNN	ResNet-18
MNIST	Class only	10	**0.5022 (0.0470)**	0.2073 (0.0490)
MNIST	Pred. aux.	1,001	**0.9633 (0.0024)**	0.9408 (0.0475)
Fashion-MNIST	Pred. aux.	999	**0.8359 (0.0024)**	0.8057 (0.0046)

Table 8: Detached readout results, mean (sample SD) over five seeds. The branch is trained on labels; agreement is the fraction of test predictions shared with the CNN.

Dataset	CNN	Rule branch	Agreement	Scrambled agreement
MNIST	0.9708 (0.0017)	0.9556 (0.0018)	0.9698 (0.0020)	0.6862 (0.1020)
Fashion	0.8427 (0.0025)	0.8113 (0.0060)	0.8921 (0.0042)	0.5384 (0.0969)

a label-efficiency interpretation. We therefore treat predicate adequacy as a moderating hypothesis, not a scaling law or prospective validation.

A.6 Backbone Capacity Does Not Explain the Low-Shot Baseline

We separately benchmark the compact four-block CNN against a ResNet-18 (He et al., 2016) trained from scratch under the same data and update schedule. Table 7 shows that the larger backbone is not a stronger low-label comparator: the compact CNN is better in the one-shot MNIST experiment and in matched 1,000-label runs on MNIST and Fashion-MNIST. The corresponding plain classification paths contain 242,250 and 11,172,810 parameters, respectively. This is a backbone-selection result, not evidence that rule compilation reduces model size. The matched KBANN adds 0.61% to the compact CNN, and the causal comparison for rules remains hybrid versus compact CNN within the same backbone.

A.7 Detached Readout and Extraction

If stamped rules disturb a working CNN through gradients, a detached branch can read its representation without changing CNN training. CNN accuracy is then identical across rule arms. Table 8 shows that the label-trained correct branch has high predictive agreement with the CNN on both datasets, while scrambled agreement is much lower. The branch itself remains less accurate than the CNN.

This branch is trained against class labels rather than CNN decisions, so agreement does not show that it recovered the CNN's causal reasoning. It is a content-dependent auxiliary readout, not a faithful explanation. Separately, we port the M-of-N extraction procedure of Towell & Shavlik (1993) to the learned rule layers. In the three-seed, ten-class MNIST setting without pruning, first-layer stamped-edge retention is 1.0000 and the clustered network agrees with the trained network on 0.9814–0.9903 of test examples. The evaluator operates on learned soft predicates; formerly zero weights may enter the extracted groups, and no new antecedents are recovered. The round trip therefore demonstrates structural retention, not symbolic discovery or a separately verified Boolean program over pixels.

Imbalance is not an advantage. Under 50:1 class skew and class-weighted training, KBANN does not outperform the standard remedy. Weighted KBANN minus weighted CNN is $+0.0102$ on MNIST (not significant), -0.0433 on Fashion-MNIST, and -0.0349 on CIFAR-10. Parameter counts are effectively identical, and recorded KBANN training time is higher on all three datasets. We therefore make no claim of imbalance robustness, lower cost, or faster deployment.

A.8 Supervision Accounting

Table 9: Supervision accounting for one-shot MNIST. The neural subset is contained in the rule-fit pool where both are used, so the union, rather than their sum, gives the distinct labeled-image count. Calibration images fit predicate thresholds without using their class labels.

Rule source	Neural	Rule fit	Labeled union	Calibration
Supplied	10	0	10	20,000
Same-budget induced	10	same 10	10	20,000
Pool-induced (exploratory)	10	20,000	20,000	20,000

A.9 Prior Scale and the Signed Projection

The signed projection has no intrinsic magnitude, so its class columns are normalized to a fixed L_2 norm and the comparison in Section 4.2 is run at three settings. The compiled branch's own output has standard deviation 1.374 on MNIST, which the $L_2 = 2.3$ setting matches.

Table 10: Signed projection minus compiled stack on MNIST with the supplied theory, at three prior scales. Positive means compiling costs accuracy. Ten seeds, paired.

Column L_2	Branch logit sd	1 img/cls	2	3	5
1.5	0.895	$+0.0388$	$+0.0367$	$+0.0409$	$+0.0331$
2.3	1.373	$+0.0664$	$+0.0469$	$+0.0442$	$+0.0313$
4.0	2.387	$+0.0594$	$+0.0294$	$+0.0164$	$+0.0077$

The signed projection leads in eleven of these twelve cells, across a $2.7\times$ range of scale spanning the compiled branch's own, and by the largest margin when the two are matched. The single exception is the loudest prior at five images per class, where an over-scaled branch expires sooner; the interpolation coefficient λ is not annealed in any arm, which is the same limitation noted in Section 4.3 for the residual case.

A.10 Extraction Measurement

M-of-N extraction clusters the incoming weights of each learned rule unit and replaces each cluster by its mean, following Towell & Shavlik (1993). Reported fidelity is test-set prediction agreement between this clustered network and the corresponding trained network. Retention is evaluated on the first compiled rule layer: an originally stamped edge is retained when its sign is unchanged and its magnitude exceeds half the largest absolute incoming weight for that output unit. The ten-class result uses seeds 42–44, zero pruning, Towell bias initialization, and max disjunction. It must not be pooled with the separate four-class, one-seed pruning experiment.