WRONG RULES, RIGHT PREDICTIONS: CONCEPT REMAPPING AND REPAIR IN NEURO-SYMBOLIC MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neuro-symbolic models combine neural perception with symbolic knowledge, but what happens when the symbolic knowledge itself is wrong? A model can compensate for incorrect rules by changing the meaning of its learned concepts while preserving task performance. On MNIST addition, permuted rules retain 97.5% label accuracy even as joint concept accuracy falls to 35.8%. We study what can be certified about such misspecified knowledge before training and whether these guarantees translate into recovery during learning. Given a reference for the intended semantics, we certify whether concept remapping can compensate for an incorrect program and whether the intended program is its unique minimum-edit repair. For full-support addition with n concepts, we prove that the nearest competing zero-loss program differs in exactly $2(n-1)$ table entries, yielding a tight unique-repair guarantee for up to $n-2$ corruptions. Yet identifiability does not imply recoverability. On ten previously unseen identifiable twenty-edit tables, a residual encoder recovers the intended program in 48/50 MNIST runs but only 10/50 SVHN runs, while a smaller CNN recovers 46/50 on both. Failures often coincide with the collapse of a single-digit concept. These results separate structural compatibility, repair identifiability, and neural recoverability under misspecified symbolic knowledge.

1 INTRODUCTION

Neuro-symbolic models combine neural perception with symbolic knowledge. A neural component maps raw inputs to concepts, while a symbolic component reasons over those concepts to produce predictions. This separation is attractive because the intermediate concepts are expected to carry meaningful semantics. But what happens when the symbolic knowledge supplied to the model is itself misspecified?

A model can hide such an error by changing what its learned concepts mean. Figure 1(a) illustrates this for digit addition. If the supplied table swaps $3 \leftrightarrow 5$ and $7 \leftrightarrow 9$, the encoders can learn the same swap and still produce the correct sums. On MNIST, label accuracy remains 0.975 while joint concept accuracy falls to 0.358. The model, therefore, appears correct at the task level even though the semantics of its learned concepts have changed. Task accuracy alone cannot reveal this failure.

Reasoning-shortcut work has shown that unintended concepts can arise even when the supplied symbolic knowledge is correct (Marconato et al., 2023b; He & Li, 2025). Existing identifiability analyses largely study this setting, where the intended concept mapping is assumed to be compatible with the supplied knowledge. For example, Takemura et al. (2026) explicitly assumes that the intended mapping satisfies the supplied constraints. We study the complementary case in which the supplied symbolic program K may itself disagree with the intended program $K^\star$.

This setting raises three distinct questions. *Compatibility*: can a remapping of the learned concepts reconcile K with $K^\star$? *Repair identifiability*: among zero-loss explanations, is $K^\star$ the unique minimum-edit repair of K? *Neural recovery*: if the intended repair is identifiable, does neural training actually recover it? These questions separate what the symbolic structure permits, what is uniquely determined, and what learning ultimately recovers.

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.

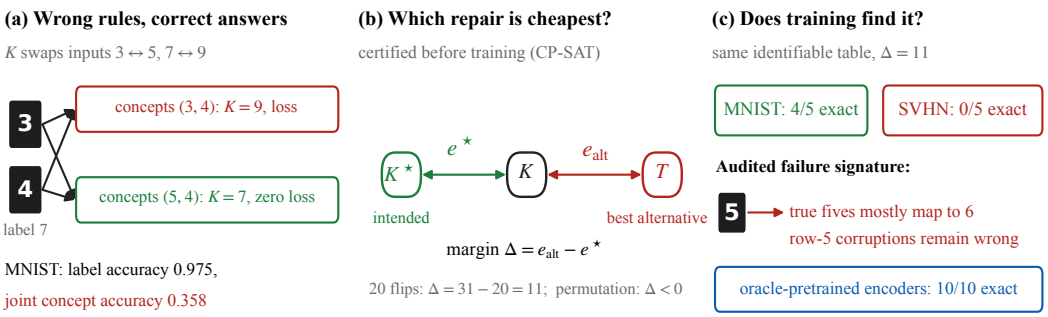


Figure 1: (a) Misspecified symbolic knowledge can be compensated for by changing concept semantics while preserving task predictions. (b) The certified margin $\Delta = e_{\text{alt}} - e^{\star}$ compares the edit cost $e^{\star}$ of the intended repair with the cost e_{alt} of the cheapest competing zero-loss explanation. (c) An identifiable repair need not be recovered by neural training.

We develop a reference-based framework for answering the first two questions. It tests whether concept remapping can compensate for the supplied program and compares the intended repair with alternative zero-loss explanations. Let $e^{\star}$ denote the number of edits needed to transform K into $K^{\star}$, and let e_{alt} denote the cost of the cheapest competing zero-loss explanation. We define the certified edit margin.

$$\Delta = e_{\text{alt}} - e^{\star}.$$

A positive Δ certifies that the intended program is the unique minimum-edit zero-loss explanation. Importantly, this certificate depends only on the symbolic programs and can therefore be computed before neural training.

For full-support addition with n concepts, we obtain an exact structural result. We prove that the nearest competing zero-loss program differs from the intended addition program in exactly $2(n-1)$ table entries. Consequently, every corruption of at most $n-2$ cells has the intended program as its unique minimum-edit repair, while a tie can occur at $n-1$ corruptions. This radius is tight, although instance-specific certificates can extend beyond it.

These structural guarantees do not determine what neural training recovers. At a fixed number of corruptions, two pre-registered replications find only weak associations between the certified margin and recovery, with neither remaining significant after correction. We then evaluate ten previously unused identifiable twenty-edit tables. A residual encoder recovers the intended program in 48/50 MNIST runs but only 10/50 SVHN runs, while a smaller CNN recovers 46/50 runs on both datasets. Failures often coincide with the collapse of a single-digit concept. On one difficult table, correcting the affected digit representation before repair increases exact recovery from 0/10 to 5/10, while encoders pretrained on the intended concepts recover in all ten runs. These results show that a repair can be structurally identifiable and reachable without being recovered by end-to-end training. Concept collapse provides a supported, though incomplete, explanation for this gap.

Our contributions are:

1. **A reference-based analysis of misspecified symbolic knowledge.** We separate compatibility, repair identifiability, and neural recovery, and provide certificates for concept compensation, concept preservation, and minimum-edit repair.
2. **A tight repair theorem for full-support addition.** We prove that the nearest competing zero-loss program is exactly $2(n-1)$ table entries away from the intended program, yielding a tight worst-case unique-repair radius of $n-2$ corruptions.
3. **An empirical separation between identifiability and recovery.** Pre-registered experiments show that structural certificates do not ensure neural recovery. At fixed edit count, margin-recovery associations are weak and not significant after correction. On previously unused identifiable tables, recovery varies with both the encoder and visual domain, and targeted interventions connect residual failures to the collapse of individual concepts.

2 Related Work

Reasoning shortcuts arise when a model performs well on the observed task without capturing the intended semantics. More broadly, shortcut learning describes predictors that exploit patterns that succeed on a benchmark but fail to transfer (Geirhos et al., 2020). In neuro-symbolic models, this can appear as a concept mapping that preserves task labels while changing the meaning of learned concepts (Marconato et al., 2023b), and similar failures can affect knowledge-based continual learning (Marconato et al., 2023a). Recent work has studied joint identifiability of concepts and inference layers (Bortolotti et al., 2025), connected learnability to uniqueness in an induced constraint satisfaction problem (He & Li, 2025), and examined whether a knowledge base provides enough supervision to identify intended concepts (Tao et al., 2024). The rsbench benchmark separates concept quality from task performance (Bortolotti et al., 2024), while a recent survey summarizes reasoning shortcuts more broadly (Marconato et al., 2026). Across much of this work, the supplied knowledge acts as the semantic reference. We instead allow the supplied program K to disagree with the intended program $K^\star$.

This distinction is related to recent work on neuro-symbolic identifiability. Takemura et al. (2026) verify shortcut uniqueness, study minimal repairs obtained by fixing concept positions, and bound the label queries required for oracle disambiguation. Their analysis assumes that the intended mapping satisfies the supplied constraints. We study the complementary case $K \neq K^\star$, where the symbolic program itself may require repair. We certify whether $K^\star$ is the unique minimum-edit zero-loss repair and ask whether neural training recovers it.

Related work also separates structural properties from optimization behavior. Xu (2026) distinguish what value symmetries permit, what architectures can realize, and what optimization selects. Independence factorization can induce overconfidence (van Krieken et al., 2024) and obscure shortcut combinations (van Krieken et al., 2025), while BEARS addresses uncertainty using ensembles (Marconato et al., 2024). Our framework instead connects neural concept correctness to discrete symbolic repair.

Several approaches mitigate shortcuts while keeping symbolic knowledge fixed. These include semantic loss (Xu et al., 2018), constraint-tracking methods (Li et al., 2023a), softened grounding (Li et al., 2023b), prototype-based concept anchoring (Andolfi & Giunchiglia, 2025), curricula for abductive learning (Wei et al., 2025), one-to-one grounding in differentiable logic programs (Takemura & Inoue, 2026), and limited concept supervision (Li et al., 2026). Abductive methods similarly revise perceived facts (Dai et al., 2019), handle ambiguous explanations (He et al., 2024), rectify inconsistent predictions (Hu et al., 2025), or use rule-based verification (Jia et al., 2025). In these settings, the symbolic rules generally remain the reference.

Some work explicitly considers imperfect symbolic knowledge. Safe abductive learning refines inaccurate rules (Yang et al., 2024), knowledge refinement updates conflicting knowledge bases (Huang et al., 2023), and inductive neuro-symbolic learners jointly learn programs and neural components (Cunnington et al., 2023). Our learned table serves as an experimental instrument rather than a general-purpose refinement method. The probabilistic head follows DeepProbLog-style marginalization (Manhaeve et al., 2021). Concept bottleneck models expose related gaps between correct predictions and corrupted concepts (Koh et al., 2020; Havasi et al., 2022), and symbol correctness matters for explanation (Bembenek & Murray, 2024). We focus on cases where the intended symbolic repair is uniquely identifiable, yet neural training still fails to recover it.

3 Problem formulation

Let $\mathcal{V} = \{0, \ldots, n-1\}$ be the concept vocabulary and $\mathcal{Y}$ the label set. True concepts $c = (c_1, c_2)$ follow a distribution π with finite support $\mathcal{S} \subseteq \mathcal{V}^2$ and positive mass on every supported pair. The true program $K^\star : \mathcal{V}^2 \to \mathcal{Y}$ generates $y = K^\star(c)$. Conditional on c, images are sampled independently, $x_i \sim P_i(\cdot \mid c_i)$: each slot image depends only on its own concept. Our sampler guarantees this for SVHN too; its crops, resizing, and clutter only make classes overlap more (Appendix C.6). A supplied table K may differ from $K^\star$. Except for the stated noise experiment, labels remain correct; we corrupt the knowledge, not the targets.

Table 1: Levels of correctness in neuro-symbolic evaluation.

Level	Quantity	Correct when
Task	label accuracy of $p_{\theta,K}(y \mid x)$	predicted label equals y
Concept (hard / soft)	A_{hard} / A_{soft}, Equation 2	argmax pair is c / mass on c
Symbolic consistency	$\delta_{\mathrm{det}} = \min_{\alpha \in \mathcal{D}} v_K(\alpha)$	$\delta_{\mathrm{det}} = 0$, i.e. $\mathcal{A} \neq \emptyset$
Program recovery	learned table T	$T(a,b) = K^\star(a,b)$ for all $(a,b) \in \mathcal{V}^2$

Local encoders $q_{\theta_i}(a \mid x_i)$ each see only their own image and output categorical distributions. Write $x = (x_1, x_2)$ and $\theta = (\theta_1, \theta_2)$. The fixed-table model marginalizes their product:

$$p_{\theta,K}(y \mid x) = \sum_{a,b \in \mathcal{V}} \mathbf{1}\{K(a,b) = y\} q_{\theta_1}(a \mid x_1) q_{\theta_2}(b \mid x_2). \tag{1}$$

Training minimizes empirical negative log-likelihood (NLL). For evaluation we use the bounded population risk $R_B = \mathbb{E}[\min\{B, -\log p_{\theta,K}(y \mid x)\}]$, with $B \geq 1$. This distinction matters because clipping invalidates a naive Jensen argument for log-loss. All expectations here refer to the specified data distribution.

For a fixed tie-breaking rule, joint hard concept accuracy and joint soft concept mass are

$$A_{\mathrm{hard}} = \mathbb{E}\left[\prod_{i=1}^{2} \mathbf{1}\{\arg\max_a q_{\theta_i}(a \mid x_i) = c_i\}\right], \qquad A_{\mathrm{soft}} = \mathbb{E}\left[\prod_{i=1}^{2} q_{\theta_i}(c_i \mid x_i)\right]. \tag{2}$$

We report both. Label accuracy evaluates the argmax of Equation 1; it is a third quantity. In particular, averaging a concept distribution over images before taking its argmax does not reproduce A_{hard}.

Our discrete analysis ranges over $\mathcal{D}$, all pairs of deterministic maps $\alpha_i : \mathcal{V} \to \mathcal{V}$. Write $\alpha(c) = (\alpha_1(c_1), \alpha_2(c_2))$ and define

$$v_K(\alpha) = \Pr_{c \sim \pi}[K(\alpha(c)) \neq K^\star(c)], \quad f(\alpha) = \Pr_{c \sim \pi}[\alpha(c) = c], \quad \mathcal{A} = \{\alpha \in \mathcal{D} : v_K(\alpha) = 0\}. \tag{3}$$

We call an element of $\mathcal{A}$ a *cross-shortcut*, allowing the identity when K is correct. A nonidentity map represents compensation across the two programs. These maps need not be permutations in general. We assume full support and uniform π for the main digit experiments; restricted support is treated explicitly in the appendix. Table 1 lists four levels of correctness (δ_{det}: Section 4); each can hold while the next fails.

4 Reference-based certification and repair

The framework first analyzes the two symbolic tables, then compares its certificates with separately trained neural models. It does not insert a solver into the training loop. Access to $K^\star$ is required for the certification stage and for evaluating concept correctness and exact repair. Ordinary weakly supervised training uses only images, task labels, and K.

Compatibility and concept ceilings. We compute the minimum deterministic violation $\delta_{\mathrm{det}} = \min_{\alpha \in \mathcal{D}} v_K(\alpha)$. A positive value certifies that no slotwise reassignment can exactly reconcile K with $K^\star$. To study concept preservation at a violation tolerance $\varepsilon \geq 0$, define

$$\Phi(\varepsilon) = \max_{\alpha \in \mathcal{D} :\, v_K(\alpha) \leq \varepsilon} f(\alpha), \qquad \max \emptyset := 0. \tag{4}$$

When $\mathcal{A} \neq \emptyset$, $\Phi(0)$ is the best joint fixed-point mass among exact compensations. It is an ideal zero-loss quantity, not the measured accuracy of a trained checkpoint.

Minimum-edit alternatives. Let $T : \mathcal{V}^2 \to \mathcal{Y}$ be a candidate repaired table and $\mathrm{ed}(K,T) = \sum_{a,b} \mathbf{1}\{K(a,b) \neq T(a,b)\}$ count changed cells. Edit count is our stated preference among explanations, not a preference inferred from labels. It is also the maximum-a-posteriori explanation when

cells are corrupted independently and uniformly (Appendix A.5). Under full support, the identity explanation costs $e^\star = \mathrm{ed}(K, K^\star)$. Its cheapest nonidentity competitor costs

$$e_{\mathrm{alt}} = \min_{\alpha \in \mathcal{D},\, \alpha \neq \mathrm{id},\, T} \left\{ \mathrm{ed}(K, T) : T(\alpha(c)) = K^\star(c)\ \forall c \in \mathcal{V}^2 \right\}, \qquad \Delta = e_{\mathrm{alt}} - e^\star. \tag{5}$$

Here id is the identity map pair and $\min \emptyset := +\infty$. A positive Δ makes $(\mathrm{id}, K^\star)$ the unique minimum-edit zero-loss explanation. At zero it ties with an alternative; below zero it is more expensive. This criterion follows from comparing feasible explanations. It says nothing about minimizing a continuous penalized objective in a finite number of gradient steps.

Computing a certificate. We encode $2n$ integer map variables with domain $\mathcal{V}$. For each supported pair (u, v), an element constraint reads cell $n\alpha_1(u) + \alpha_2(v)$ of the flattened table. Reified equalities to $K^\star(u, v)$ give violations; fixed-point indicators give f. Repair adds n^2 label variables for T, edit indicators against K, and a constraint that at least one map value moves. CP-SAT minimizes violations or edits, or maximizes fixed mass via the equivalent minimization of moved mass. For uniform support, objectives are integer counts divided by $|\mathcal{S}|$. We validate returned witnesses and retain solver status, incumbent, and bound. Only closed bounds are called exact. A minimization lower bound and a maximization upper bound remain safe when a solve is unfinished. Enumerating maps directly would consider n^{2n} possibilities; we make no polynomial-time scaling claim for the constraint solver.

Neural realization. As a controlled realization of this analysis, rather than a claim of a new competitive refinement system, replace the indicator table in Equation 1 by $Q_\psi(y \mid a, b) = \mathrm{softmax}(z_{ab})_y$ with $z_{ab} \in \mathbb{R}^{|\mathcal{Y}|}$ and $\psi = (z_{ab})_{a,b}$. For N_{train} training pairs and penalty weight $\lambda \geq 0$, minimize

$$\mathcal{L}(\theta, \psi) = \frac{1}{N_{\mathrm{train}}} \sum_{j=1}^{N_{\mathrm{train}}} -\log p_{\theta, Q_\psi}(y_j \mid x_j) + \lambda \sum_{a,b} \big[1 - Q_\psi(K(a, b) \mid a, b)\big]. \tag{6}$$

The penalty approaches edit count for deterministic tables. It is a surrogate, not an exact solution of Equation 5. Under Adam the soft table first moves toward K; repair starts only once perception has formed (Appendix C.7). We initialize the supplied label's logit at 6 and the others at zero, and evaluate exact recovery by $\arg\max_y z_{ab,y} = K^\star(a, b)$ for every cell. Ablations remove the penalty, replace it by L1 logit deviation, or initialize without K.

Downstream diagnostic application: loss diagnostics and oracle auditing. We present these tools as a downstream application of the certificates, not as a theoretical contribution. A complementary score subtracts a seed-matched unconstrained neural baseline's loss from fixed-table loss. We calibrate a threshold for *positive-floor* tables; zero-floor wrong tables are not its positive class. Separately, an audit policy selects training examples for which an oracle reveals the true concept pair. The known task label then verifies that true table cell. Residual coverage ranks predicted cells by mean clipped NLL, selects high-residual representatives, and repairs the verified true cells before retraining. Masked coverage excludes verified cells between query batches. These policies are heuristics for finding wrong cells; their query count need not equal the number of distinct true cells inspected. They are not the discrete certificate or a blind repair guarantee.

5 What the certificates establish

The probabilistic statements concern fixed K, local encoders, and the conditional-independence model of Section 3. Define the class-averaged channel $\bar{\alpha}_i(a \mid v) = \mathbb{E}[q_{\theta_i}(a \mid x_i) \mid c_i = v]$, with arbitrary rows for zero-probability values. Drawing every map value $\alpha_i(v)$ independently from this channel defines a distribution μ on $\mathcal{D}$. Finite summation gives

$$\mathbb{E}_\mu[v_K(\alpha)] = \mathbb{E}[1 - p_{\theta, K}(y \mid x)] \leq R_B, \qquad \mathbb{E}_\mu[f(\alpha)] = A_{\mathrm{soft}}. \tag{7}$$

The inequality uses $1 - p \leq \min\{B, -\log p\}$ for $B \geq 1$ directly, without assuming clipped log-loss is convex.

Proposition 1 (Compatibility and finite-loss concept mass)**.** *Every such predictor satisfies* $R_B \geq \delta_{\mathrm{det}}$*. An ideal row-stochastic concept channel with zero loss exists if and only if* $\mathcal{A} \neq \emptyset$*. For a zero-risk predictor,* $A_{\mathrm{soft}} \leq \Phi(0)$*; for every* $\varepsilon > 0$*,*

$$A_{\mathrm{soft}} \leq \min\{1, \Phi(\varepsilon) + R_B/\varepsilon\}. \tag{8}$$

The floor bounds clipped risk, not argmax task-error probability. The ideal-channel existence statement does not imply that a neural encoder can realize it from ambiguous images. Equation 7 proves the floor. At zero risk, μ is supported on cross-shortcuts. For finite risk, Markov's inequality bounds the mass of maps with violation above ε. These are consequences of the finite-map analysis, not claims of a new general concentration principle.

Proposition 2 (Hard–soft confidence bridge)**.** *Let* $C(x) = \prod_i \max_a q_{\theta_i}(a \mid x_i)$*. With hard and soft metrics as in Equation 2,*

$$|A_{\mathrm{hard}} - A_{\mathrm{soft}}| \leq \mathbb{E}[1 - C(x)]. \tag{9}$$

Pointwise, either both argmax concepts are correct and the true-pair mass is C, or the true and argmax pairs are distinct and their masses sum to at most one. The bridge requires the product concept distribution, but no independence assumption on the data: it holds pointwise. The conditional-independence assumption is needed for Equation 7. The bridge also holds empirically and for encoders attached to a learned head. Proposition 1, however, requires the specified fixed head. Finite-sample population claims additionally need risk and confidence estimation corrections; Appendix A.2 states these and explains why the present finite-loss bounds are usually loose. At $B = 10$ its Hoeffding corrections (0.12 nats) are descriptive.

The repair criterion is general, but its distance can depend strongly on the program. Addition permits the following exact characterization.

Theorem 3 (Tight addition repair radius)**.** *For* $K^\star(u, v) = u + v$ *on* $\{0, \ldots, n-1\}^2$*,* $n \geq 2$*, full support, separate slot maps, and unweighted table edits, every zero-loss table is* $T_{p,q}(a, b) = p(a) + q(b)$ *for permutations* p, q*. Moreover,*

$$\min_{T_{p,q} \neq K^\star} \mathrm{ed}(K^\star, T_{p,q}) = 2(n-1). \tag{10}$$

Consequently, $t = \mathrm{ed}(K, K^\star) \leq n - 2$ *guarantees a unique minimum-edit identity explanation. This uniform radius is tight: a tie exists at* $t = n - 1$*.*

To see the structure, if two true values shared a mapped value, fixing the other slot would force two different sums to coincide. Both maps are therefore permutations. Agreement with $K^\star$ is the inner product of their reflected displacement histograms. Cauchy–Schwarz bounds it by $(n-1)^2 + 1$ for two nonidentity permutations; opposite cyclic shifts attain the bound. The triangle inequality then gives $e_{\mathrm{alt}} \geq 2(n-1) - t$. Complete proofs are in Appendix A. For ten digits, eight arbitrary wrong cells are within the guaranteed radius and a nine-cell tie is possible. This is addition-specific: for clean comparison $e_{\mathrm{alt}} = 2$ for every n (Proposition 6), so no corruption is uniformly certified. Appendix Table 4 separates general, family-specific, and purely empirical results.

6 Experimental setup

We use MNIST handwritten digits (LeCun et al., 1998) and SVHN photographic house-number crops (Netzer et al., 2011). Each example contains two independently sampled images conditional on a uniformly sampled supported digit pair. Addition has 19 labels; comparison has three. Main runs use 30,000 training pairs, 5,000 validation pairs, and 10,000 test pairs. Image pools for training and validation are disjoint. Test sampling is fixed across seeds. SVHN is converted to grayscale and resized to 28×28; it extends the image domain while retaining digit concepts and the same symbolic task.

Each separate SlotCNN encoder has two 3×3 convolutional layers with 32 and 64 channels, ReLU and max-pooling, a 128-unit hidden layer, and a ten-way output. Adam trains for ten epochs with batch size 128 and encoder learning rate 10^{-3}. Learned tables use head learning rate 10^{-2} and, unless ablated, $\lambda = 0.01$. We select the checkpoint with lowest validation clipped NLL; evaluation clips at $B = 10$. The free baseline concatenates the two CNN feature vectors before a label classifier

Table 2: Fixed addition tables, five seeds per row; mean ± sample SD. Label and joint hard accuracy are primary metrics; joint soft mass is supporting. Permutation corruption preserves label accuracy while reducing concept correctness. Random corruption also separates hard correctness from soft mass.

Domain	Knowledge	Label	Joint hard	Joint soft
MNIST	`clean`	0.975 ± 0.003	0.975 ± 0.003	0.968 ± 0.006
MNIST	`perm_both`	0.975 ± 0.003	0.358 ± 0.002	0.357 ± 0.001
MNIST	`flip_20`	0.730 ± 0.008	0.890 ± 0.014	0.591 ± 0.026
SVHN	`clean`	0.712 ± 0.008	0.710 ± 0.008	0.626 ± 0.011
SVHN	`perm_both`	0.711 ± 0.007	0.276 ± 0.003	0.248 ± 0.003
SVHN	`flip_20`	0.490 ± 0.011	0.573 ± 0.017	0.334 ± 0.022

and trains for 40 epochs. Concept-pretrained initialization is an oracle control with extra supervision and compute.

The `perm_both` table permutes both inputs by $(3\ 5)(7\ 9)$; `perm_one` permutes only the first. Random `flip_`t changes t distinct cells to different labels. Complete configurations appear in Appendix B. Seed counts per study are listed in Appendix B; repeated seeds on one table do not create new tables. A second encoder, SmallResNet, uses four pre-activation residual blocks at widths 32 and 64 with batch normalization and global average pooling, at the same budget. Budget controls run SlotCNN for 40 and 100 epochs. An oracle control (E18) pretrains both SlotCNN encoders on true concepts, then repairs the same table.

Primary metrics are label accuracy, joint hard concept accuracy A_{hard}, and exact table recovery; clipped risk $\widehat{R}_B$ and soft mass A_{soft} are supporting metrics. Accuracies are fractions; concept metrics are joint unless stated otherwise. Seed summaries use sample standard deviation (SD). Intervals over independently sampled corruption tables use a Student t interval on the ten seed-averaged table values, conditional on the shared image protocol. Boundary results show individual table rates rather than binomial intervals over pooled seeds.

7 Results

7.1 Wrong knowledge can preserve predictions while corrupting concepts

Table 2 shows the central dissociation. On MNIST, `perm_both` preserves label accuracy (0.9753 versus 0.9750 for clean) while hard concept accuracy falls from 0.9750 to 0.3581 and soft concept mass to 0.3566. The certified unique exact compensation inverts the supplied permutations; each fixes six digits, so its ideal joint fixed mass is $0.6^2 = 0.36$, which the trained models approach at nonzero risk.

SVHN reproduces the dissociation (label 0.7124 versus 0.7110; hard concept accuracy 0.7102 versus 0.2762), as does a second encoder (Table 26). Under random corruption the two concept metrics diverge instead: on MNIST `flip_20`, hard correctness is 0.8897 but soft mass only 0.5909. All 1,150 checkpoints with saved soft metrics satisfy Proposition 2 pointwise and in aggregate.

A positive δ_{det} rules out exact compensation. In the original addition sweep, all eight positive-floor conditions have mean training clipped loss within 0.0241 nats of a feasible numerical tabular reference (Appendix C.1). Across thirty independently sampled corruption tables, ten each at 5, 10, and 20 edits, none of the 90 test losses falls below its certified floor. The floor, $t/100$, cannot rank tables within an edit count, and training/reference gaps stay below 0.035 nats on 28 of 30 table means but reach 0.576 and 0.227 on two (Table 10). Incompatibility explains unavoidable error without predicting achieved loss.

7.2 The certified margin distinguishes identifiable from ambiguous tables

The sign of Δ is a property of the tables alone. For ten-digit addition, Theorem 3 certifies every corruption of at most eight cells. Beyond that radius, instance certificates can still be positive: the original twenty-edit table has $e^\star = 20$, $e_{\mathrm{alt}} = 31$, and $\Delta = 11$. Permutation corruption has a

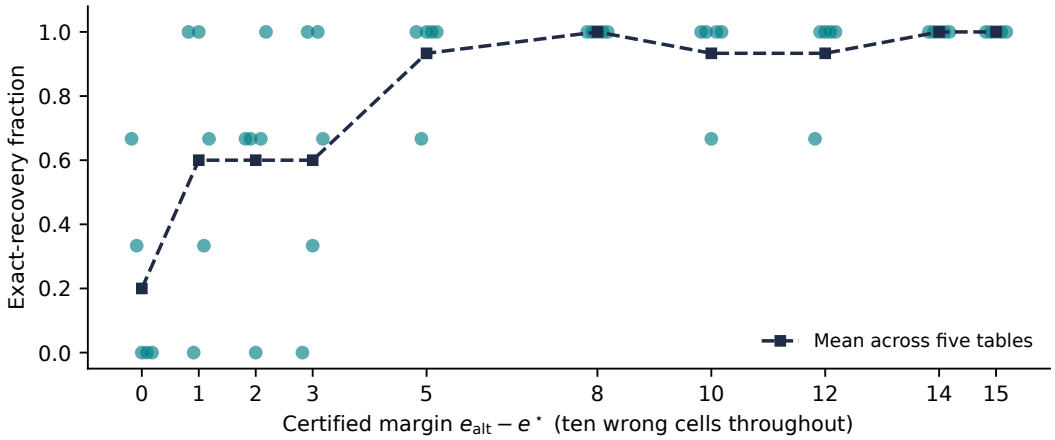


Figure 2: Exact repair at ten fixed edits ($e^\star = 10$) against the certified margin $\Delta = e_{\text{alt}} - e^\star$. Circles are individual tables' three-seed recovery fractions; squares average five tables per margin. Positive margins can fail and ties can recover.

zero-edit compensating explanation, and the structured ten-edit corruption ties at $e^\star = e_{\text{alt}} = 10$. The neural repair sweep recovers $K^\star$ in 29/30 runs over six identifiable corrupted tables and in 0/20 runs over four non-identifiable ones. These groups also differ in corruption family and edit count, which motivates the controlled study below.

7.3 At fixed edit count, the margin–recovery associations are weak and not significant after correction

From 114 constructed tables with exactly ten wrong cells, we train the first five per certified-margin bin by construction index: 50 distinct tables, three seeds each (Appendix B.1). Recovery averages 0.20 at a tie, 0.60 at margins one to three, and 0.97 at margins of at least five (Figure 2). The table-level Spearman correlation is 0.666 (no one of 20,000 permutations reaches it), but here the margin is confounded with edit composition.

The margin covaries with s_1, the number of edits agreeing with the constructed alternative (Spearman -0.90), since $\Delta \leq |D| - 2s_1 - s_2$ for the alternative's disagreement set D and third-label edits s_2 (Appendix B.1). An exploratory study at fixed $s_1 = 8$ found recovery rising from 0.07 to 0.87 (Appendix C.10). Two pre-registered replications followed, each with 50 new tables and three seeds per table. With $t = 10$ and $s_1 = 8$ fixed, recovery across margin bins $0, 2, 4, 8, \geq 9$ is 0.57, 0.50, 0.97, 0.83, and 0.73 (one-sided table-level Spearman $\rho = 0.26$, $p = 0.038$, Holm-adjusted 0.11; Appendix D.3). Because $|D|$ and s_2 still changed with the bin, the second fixes the whole composition ($|D| = 28$, $s_1 = 8$, $s_2 = 0$), so tables differ only in which cells are corrupted and in the alternative's permutation pair; in the configurations we explored this is possible only for margins 7–12, since at smaller $|D|$ a fixed composition almost always pinned the margin. Recovery is 0.57 at margins 7–8 and 0.77–0.97 above ($\rho = 0.22$, $p = 0.064$, Holm-adjusted 0.19; Appendix E.2). Neither primary test is significant after correction, so these experiments show weak, non-significant associations.

7.4 An identifiable repair can fail, and failure accompanies a concept collapse

On the original twenty-edit table ($\Delta = 11$), SlotCNN repairs 4/5 MNIST and 0/5 SVHN seeds; neither 100-epoch training nor twelve optimizer settings (0/36 SVHN runs) change this (Appendices C.8, D.5). To test whether the failure belongs to the domain, we trained both encoders on both domains on ten identifiable twenty-edit tables ($\Delta = 11$–12, certified before training) that no earlier experiment used (Table 3; four further tables, shared with an earlier study, are reported in Appendix D.1). SlotCNN recovers 46/50 on SVHN and 46/50 on MNIST, so its SVHN failure is specific to the original table. SmallResNet recovers 48/50 on MNIST but only 10/50 on SVHN,

Table 3: Repair on ten identifiable twenty-edit tables that no earlier experiment used ($\Delta = 11$–12), five seeds per cell. Hard is the repaired models' joint hard concept accuracy. Per-table counts are in Appendix E.1.

Domain	Encoder	Exact	Tables ≥ 1	Hard
MNIST	SlotCNN	46/50	10/10	0.948
MNIST	SmallResNet	48/50	10/10	0.977
SVHN	SlotCNN	46/50	10/10	0.704
SVHN	SmallResNet	10/50	4/10	0.721

lower on every table (table-level paired Wilcoxon $p = 0.002$), although its repaired SVHN models reach similar concept accuracy (0.721 against 0.704). Clean perception accuracy does not predict recovery either: MNIST with pixel noise at clean concept accuracy 0.454, below SVHN's 0.710, still recovers 5/5 (Appendix E.4).

Failures usually accompany a single-digit concept collapse: 35 of 50 failing checkpoints on these tables recognize some digit in some slot with accuracy below 0.5, against none of 150 successes; the other 15 fall short of that threshold, and the collapsed digit varies with the table. On the original table, failing checkpoints map 0.873–0.997 of slot-1 fives to six, and there we intervened (Appendices D.2, E.3). A read-out correction on fives and sixes alone is undone by the unchanged repair (0/10; pre-registered $p = 1$). Retraining the read-out on all ten digits and then freezing slot 1 recovers 5/10 with 1.2 wrong cells on average; the identical intervention without digit 5 recovers 0/10 with 6.6. The pre-registered test on exact recovery is not significant ($p = 0.06$); the wrong-cell difference is a secondary result (paired Wilcoxon $p = 0.002$), and recovery tracks how far digit 5 is restored. An oracle start pretrained on true concepts recovers 5/5 per domain (Appendix C.9). Including digit 5 thus reduces residual errors on this one table; as the whole read-out is retrained and slot 1 frozen, this supports but does not isolate a causal role of the digit-5 representation. Collapse is a supported but incomplete explanation of failed recovery. Safe-ABL (Yang et al., 2024), adapted to a supplied table, behaves the same way: from pretrained encoders it flags wrong entries (AUROC 0.95–0.97) and, given replacement candidates, repairs all 18 runs, but from scratch it neither detects nor repairs (Appendix E.5).

E8, the matched-composition study, and the frozen-encoder and oracle controls are exploratory; the other studies were pre-registered. All share one fixed test-image protocol.

8 Limitations and Conclusion

Our analysis requires access to the semantic reference $K^\star$, while concept evaluation and oracle queries use concept labels. The theoretical bounds assume finite categorical concepts, local encoders, and the stated factorization, and the tight repair radius is specific to full-support addition with unweighted edits. Our experiments cover two digit-image domains, two symbolic families, and two slotwise encoders; with a joint encoder, concept identifiability can fail even when the symbolic knowledge is correct. The single-digit collapse is established by intervention on one table and only by association on others, and the finite-risk concept bounds can be loose.

We separate what can be certified about misspecified symbolic knowledge from what neural training recovers. Incorrect knowledge can be hidden by compensating changes in concept semantics. Our certificates determine whether such compensation is possible and whether the intended program is the unique minimum-edit repair, which, for full-support addition, is guaranteed up to $n - 2$ corruptions. These guarantees do not ensure recovery. Margin-recovery associations are weak and not significant after correction, and on previously unused identifiable tables, a residual encoder recovers 48/50 on MNIST but 10/50 on SVHN. A uniquely identifiable repair can therefore remain unrecovered by end-to-end training.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work primarily to support manuscript review and editing, as well as debugging, consistency checks, and verification of saved experimental artifacts. All experiments, scientific claims, numerical results, proofs, and findings were independently conducted and verified by the authors. The final manuscript was reviewed and approved by the authors, who take full responsibility for the content of this work.

ETHICS STATEMENT

This work investigates concept learning and symbolic knowledge in neuro-symbolic models through theoretical analysis and experiments on the publicly available MNIST and SVHN benchmarks. The study involves no recruitment of human participants or collection of new personal data. We do not identify specific ethical concerns arising from the benchmark experiments reported in this paper.

REPRODUCIBILITY STATEMENT

Sections 3–6 describe the model, objectives, certification problems, and experimental protocol. Complete proofs are provided in Appendix A, with implementation and construction details in Appendix B. The anonymous supplementary archive contains the code, tests, 2,730 saved training records, 1,150 soft-metric records, construction tables, solver certificates, pre-registration files with SHA-256 hashes, per-epoch traces, per-digit diagnostics, and the experiment ledger. It also includes scripts that regenerate all 44 manuscript tables from the saved records, compare them with the reported values, and independently verify the results. A manifest provides checksums for all files. Model checkpoints and per-example loss arrays (5.2 GB) are excluded from the archive because of their size and are hosted anonymously at `https://huggingface.co/datasets/event-horizon-data/wrongk-checkpoints`; the provided scripts download and verify them against published checksums. Each run is keyed by a hash of its configuration and can be reproduced using the provided training scripts. MNIST and SVHN are downloaded automatically by the data loader. Code for all experiments is also available at `https://anonymous.4open.science/r/wrongk-iclr27`.

REFERENCES

Luca Andolfi and Eleonora Giunchiglia. Right for the right reasons: Avoiding reasoning shortcuts via prototypical neurosymbolic AI. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL `http://papers.nips.cc/paper_files/paper/2025/hash/e6bb63d00f597c380c012bd99c68b0c1-Abstract-Conference.html`.

Aaron Bembenek and Toby Murray. Symbol correctness in deep neural networks containing symbolic layers, 2024. URL `https://doi.org/10.48550/arXiv.2402.03663`.

Samuele Bortolotti, Emanuele Marconato, Tommaso Carraro, Paolo Morettin, Emile van Krieken, Antonio Vergari, Stefano Teso, and Andrea Passerini. A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL `http://papers.nips.cc/paper_files/paper/2024/hash/d1d11bf8299334d354949ba8738e8301-Abstract-Datasets_and_Benchmarks_Track.html`.

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

In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/fcdccd419c4dc471fa3b73ec97b53789-Abstract-Conference.html.

Daniel Cunnington, Mark Law, Jorge Lobo, and Alessandra Russo. Neuro-symbolic learning of answer set programs from raw data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3586–3596. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/399. URL https://doi.org/10.24963/ijcai.2023/399.

Wang-Zhou Dai, Qiu-Ling Xu, Yang Yu, and Zhi-Hua Zhou. Bridging machine learning and logical reasoning by abductive learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2811–2822, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9c19a2aa1d84e04b0bd4bc888792bd1e-Abstract.html.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020. doi: 10.1038/S42256-020-00257-Z. URL https://doi.org/10.1038/s42256-020-00257-z.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/944ecf65a46feb578a43abfd5cddd960-Abstract-Conference.html.

Hao-Yuan He and Ming Li. A learnability analysis on neuro-symbolic learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/7aa320d2b4b8f6400b18f6f77b6c1535-Abstract-Conference.html.

Hao-Yuan He, Hui Sun, Zheng Xie, and Ming Li. Ambiguity-aware abductive learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18019–18042. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/he24j.html.

Wen-Chao Hu, Wang-Zhou Dai, Yuan Jiang, and Zhi-Hua Zhou. Efficient rectification of neuro-symbolic reasoning inconsistencies by abductive reflection. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 17333–17341. AAAI Press, 2025. doi: 10.1609/AAAI.V39I16.33905. URL https://doi.org/10.1609/aaai.v39i16.33905.

Yu-Xuan Huang, Wang-Zhou Dai, Yuan Jiang, and Zhi-Hua Zhou. Enabling knowledge refinement upon new concepts in abductive learning. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium*

on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pp. 7928–7935. AAAI Press, 2023. doi: 10.1609/AAAI.V37I7.25959. URL https://doi.org/10.1609/aaai.v37i7.25959.

Lin-Han Jia, Wen-Chao Hu, Jie-Jing Shao, Lan-Zhe Guo, and Yu-Feng Li. Verification learning: Make unsupervised neuro-symbolic system feasible. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/jia25c.html.

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.

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

Yu-Feng Li, Xiao-Wen Yang, Wen-Da Wei, Jie-Jing Shao, and Lan-Zhe Guo. Mitigating neuro-symbolic reasoning shortcuts with data-driven knowledge augmentation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2861–2872, 2026. doi: 10.1145/3770855.3817988.

Zenan Li, Zehua Liu, Yuan Yao, Jingwei Xu, Taolue Chen, Xiaoxing Ma, and Jian Lü. Learning with logical constraints but without shortcut satisfaction. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/forum?id=M2unceRvqhh.

Zenan Li, Yuan Yao, Taolue Chen, Jingwei Xu, Chun Cao, Xiaoxing Ma, and Jian Lü. Softened symbol grounding for neuro-symbolic systems. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/forum?id=HTJE5Krui0g.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Neural probabilistic logic programming in deepproblog. *Artif. Intell.*, 298:103504, 2021. doi: 10.1016/J.ARTINT.2021.103504. URL https://doi.org/10.1016/j.artint.2021.103504.

Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro-symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23915–23936. PMLR, 2023a. URL https://proceedings.mlr.press/v202/marconato23a.html.

Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/e560202b6e779a82478edb46c6f8f4dd-Abstract-Conference.html.

Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Antonio Vergari, Andrea Passerini, and Stefano Teso. BEARS make neuro-symbolic models aware of their reasoning shortcuts. In Negar Kiyavash and Joris M. Mooij (eds.), *Uncertainty in Artificial Intelligence, 15-19 July 2024, Universitat Pompeu Fabra, Barcelona, Spain*, volume 244 of *Proceedings of Machine Learning Research*, pp. 2399–2433. PMLR, 2024. URL https://proceedings.mlr.press/v244/marconato24a.html.

Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Paolo Morettin, Elena Umili, Antonio Vergari, Efthymia Tsamoura, Andrea Passerini, and Stefano Teso. Symbol grounding in neuro-symbolic AI: A gentle introduction to reasoning shortcuts. *J. Artif. Intell. Res.*, 86, 2026. doi: 10.1613/JAIR.1.22389. URL https://doi.org/10.1613/jair.1.22389.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Laurent Perron, Frédéric Didier, and Steven Gay. The CP-SAT-LP solver (invited talk). In Roland H. C. Yap (ed.), *29th International Conference on Principles and Practice of Constraint Programming, CP 2023, Toronto, Canada, August 27-31, 2023*, volume 280 of *LIPIcs*, pp. 3:1–3:2. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPICS.CP.2023.3. URL https://doi.org/10.4230/LIPIcs.CP.2023.3.

Akihiro Takemura and Katsumi Inoue. Differentiable logic programming to mitigate reasoning shortcuts in neurosymbolic systems. *CoRR*, abs/2607.21185, 2026. doi: 10.48550/ARXIV.2607.21185. URL https://doi.org/10.48550/arXiv.2607.21185.

Akihiro Takemura, Katsumi Inoue, and Masaaki Nishino. Constraint-based analysis of reasoning shortcuts in neurosymbolic learning, 2026. URL https://doi.org/10.24963/kr.2026/106.

Lue Tao, Yu-Xuan Huang, Wang-Zhou Dai, and Yuan Jiang. Deciphering raw data in neuro-symbolic learning with provable guarantees. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 15310–15318. AAAI Press, 2024. doi: 10.1609/AAAI.V38I14.29455. URL https://doi.org/10.1609/aaai.v38i14.29455.

Emile van Krieken, Pasquale Minervini, Edoardo M. Ponti, and Antonio Vergari. On the independence assumption in neurosymbolic learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 49078–49097. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/van-krieken24a.html.

Emile van Krieken, Pasquale Minervini, Edoardo M. Ponti, and Antonio Vergari. Neurosymbolic reasoning shortcuts under the independence assumption. In Leilani H. Gilpin, Eleonora Giunchiglia, Pascal Hitzler, and Emile van Krieken (eds.), *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning (NeSy 2025), 8-10 September 2025, Santa Cruz, CA, USA*, volume 284 of *Proceedings of Machine Learning Research*, pp. 285–302. PMLR, 2025. URL https://proceedings.mlr.press/v284/krieken25a.html.

Wen-Da Wei, Xiao-Wen Yang, Jie-Jing Shao, and Lan-Zhe Guo. Curriculum abductive learning for mitigating reasoning shortcuts. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 6533–6541. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/727. URL https://doi.org/10.24963/ijcai.2025/727.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5498–5507. PMLR, 2018. URL http://proceedings.mlr.press/v80/xu18h.html.

Xin Xu. Reasoning shortcuts and value symmetries: What symmetry permits, architecture realizes, and optimization selects. *CoRR*, abs/2608.10420, 2026. doi: 10.48550/ARXIV.2608.10420. URL https://doi.org/10.48550/arXiv.2608.10420.

Xiaowen Yang, Jie-Jing Shao, Wei-Wei Tu, Yufeng Li, Wang-Zhou Dai, and Zhi-Hua Zhou. Safe abductive learning in the presence of inaccurate rules. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 16361–16369. AAAI Press, 2024. doi: 10.1609/AAAI.V38I15.29572. URL https://doi.org/10.1609/aaai.v38i15.29572.

A PROOFS AND SUPPLEMENTARY ANALYSIS

Table 4: Theoretical scope: what is assumed, what is proved in general, what is proved for one family, and what is only measured.

Layer	Content	Status
General framework	finite categorical concepts, local encoders, $x_i \sim P_i(\cdot \mid c_i)$, fixed table head, full or restricted support	assumptions (Section 3)
Certification	$\delta_{\mathrm{det}}, \Phi(\varepsilon), e^\star, e_{\mathrm{alt}}, \Delta$; floor and concept-mass bounds (Prop. 1); hard–soft bridge (Prop. 2)	proved for any table; values certified per instance by CP-SAT
Family-specific	addition: nearest competitor at $2(n-1)$, radius $n-2$ (Thm. 3); comparison: $e_{\mathrm{alt}} = 2$, radius 0 (Prop. 6)	proved; full support, separate maps, unweighted edits
Empirical only	silent corruption on MNIST/SVHN; margin–recovery association; SVHN repair failure; slot-1 digit collapse; optimization and encoder controls	measured; no theorem covers neural recovery

A.1 MIXTURE REPRESENTATION AND COMPATIBILITY

An ideal channel is a pair of row-stochastic matrices $\bar{\alpha}_i$. Its predictor is

$$p_{\bar{\alpha},K}(y \mid c) = \sum_{a,b} \mathbf{1}\{K(a,b) = y\} \bar{\alpha}_1(a \mid c_1) \bar{\alpha}_2(b \mid c_2).$$

Its loss is the expected clipped NLL on $c \sim \pi$ with label $K^\star(c)$. For $B \geq 1$, zero loss means correct-label probability one on every supported pair. For each slot i and true value v, independently sample $\alpha_i(v) \sim \bar{\alpha}_i(\cdot \mid v)$. At any fixed pair c, the random vector $\alpha(c)$ has the product-channel law. Therefore

$$\Pr_{\mu}[K(\alpha(c)) = K^\star(c)] = \sum_{a,b} \mathbf{1}\{K(a,b) = K^\star(c)\} \bar{\alpha}_1(a \mid c_1) \bar{\alpha}_2(b \mid c_2), \tag{11}$$

$$\Pr_{\mu}[\alpha(c) = c] = \bar{\alpha}_1(c_1 \mid c_1) \bar{\alpha}_2(c_2 \mid c_2). \tag{12}$$

Conditional independence of the two images identifies the first expression with $\mathbb{E}[p_{\theta,K}(y \mid x) \mid c]$ and the second with the conditional expected true-concept mass. Averaging over π proves Equation 7. This construction does not require that one sampled map explain every image; it is a distribution over maps representing the class-averaged channel.

Proof of Proposition 1. By definition $v_K(\alpha) \geq \delta_{\mathrm{det}}$ for every deterministic map, so Equation 7 gives $R_B \geq \delta_{\mathrm{det}}$. If a cross-shortcut exists, its one-hot channel has zero ideal loss. Conversely, zero ideal loss makes the nonnegative expected violation zero, so every map with positive μ mass has zero violation. The finite probability space has nonempty support, which supplies a cross-shortcut. The same support argument at zero neural risk gives $A_{\mathrm{soft}} \leq \Phi(0)$.

For $\varepsilon > 0$, Markov's inequality gives $\Pr_{\mu}[v_K(\alpha) > \varepsilon] \leq R_B/\varepsilon$. Splitting the expected fixed mass over this event and its complement yields

$$A_{\mathrm{soft}} = \mathbb{E}_{\mu}[f(\alpha)] \leq \Phi(\varepsilon) \Pr_{\mu}[v_K \leq \varepsilon] + \Pr_{\mu}[v_K > \varepsilon] \leq \Phi(\varepsilon) + R_B/\varepsilon.$$

Together with $A_{\mathrm{soft}} \leq 1$, this proves the claim, including an empty low-violation feasible set. The pointwise bound $1 - p \leq \min\{B, -\log p\}$ also holds at $p = 0$, with clipped loss defined as B. Thus no convexity of clipped log-loss is used. □

An ideal channel is an analysis relaxation. A deterministic mapping of true concepts may be unrealizable by a given CNN or by any image-only predictor if class-conditional image distributions overlap. The existence half of the proposition therefore concerns the channel problem, whereas the lower bound applies to every predictor satisfying the data and factorization assumptions.

A.2 CONFIDENCE AND FINITE SAMPLES

Proof of Proposition 2. Write $H(x, c)$ for joint hard correctness and $S(x, c)$ for true-pair mass. If $H = 1$, each true concept is the selected argmax, so $S = C$ and $|H - S| = 1 - C$. Otherwise $H = 0$ and the true pair is distinct from the selected argmax pair. Their probabilities under the product distribution are S and C, so $S + C \leq 1$. Thus $|H - S| = S \leq 1 - C$. Taking expectations and applying $|\mathbb{E}[H - S]| \leq \mathbb{E}|H - S|$ proves the claim. The identical argument holds for finite sample averages. Ties are harmless provided the argmax rule is fixed. □

Write $\widehat{R}_B$ and $\widehat{U}$ for sample means of clipped loss and $1-C$. For one fixed model and an independent iid test sample of size N_{test}, with $0 < \eta < 1$, one-sided Hoeffding gives, with probability at least $1 - \eta$,

$$R_B \leq \widehat{R}_B + B\sqrt{\frac{\log(1/\eta)}{2N_{\text{test}}}}. \tag{13}$$

Substituting this upper bound in Equation 8 gives a finite-sample soft-accuracy upper bound. For a hard-accuracy statement, also bound $U = \mathbb{E}[1 - C]$. For error allocations $\eta_R, \eta_U > 0$ with $\eta_R + \eta_U = \eta$, a union bound yields

$$A_{\text{hard}} \leq \min\left\{1, \Phi(\varepsilon) + \frac{\widehat{R}_B + B\sqrt{\log(1/\eta_R)/(2N_{\text{test}})}}{\varepsilon} + \widehat{U} + \sqrt{\frac{\log(1/\eta_U)}{2N_{\text{test}}}}\right\}. \tag{14}$$

The added confidence correction uses range one. For simultaneous claims over M independently specified models, allocate failure probability across models as well. Independence between their losses is not needed for the union bound. Adaptive use of the test outcomes to specify models or thresholds, however, is not justified by this argument. The population underlying our sampling is the fixed test-image pool; it is not a guarantee about unseen image sources.

At $B = 10$, $N_{\text{test}} = 10{,}000$, and $\eta = 0.05$, the single-model risk correction is 0.1224 nats before division by ε. This can make a seemingly informative empirical substitution vacuous. The saved grid in `finite_loss_bound_grid.csv` evaluates the population expression at empirical risks. For example, at $\varepsilon = 0.3$ the recorded `perm_both` value is 0.844 for soft accuracy before finite-sample correction, far above measured mass 0.357. The corresponding comparison `reverse` value at $\varepsilon = 0.4$ is 0.737, versus measured mass 0.005. These are descriptive substitutions, not high-probability certificates. The grid-based tolerance is selected using validation risk. We do not claim simultaneous non-vacuity or that these bounds accurately predict checkpoints. The ceiling $\Phi(\varepsilon)$ equals one under clean knowledge because the identity is feasible; it cannot explain low clean concept recovery.

A lower-side margin. To analyze that complementary direction, we define

$$\gamma(e) = \min_{\alpha \in \mathcal{D}:\, 1 - f(\alpha) \geq e} v_K(\alpha), \qquad 0 < e \leq 1.$$

For a nonempty constraint set with $\gamma(e) > 0$, let $Z = 1 - f(\alpha)$ under μ. Then $\Pr[Z \geq e] \leq R_B/\gamma(e)$, since every map in that event has violation at least $\gamma(e)$. Splitting the expectation gives

$$1 - A_{\text{soft}} = \mathbb{E}Z \leq e + (1 - e)\frac{R_B}{\gamma(e)}.$$

This routine bound uses the same mixture and requires its own safe solver lower bound. It is a structural diagnostic, not an optimizer convergence result. A minimum *number* of moved slot values is a different constraint from minimum joint moved probability mass; the two should not be interchanged.

A.3 ADDITION DISTANCE

Lemma 4. *If $T(\alpha_1(u), \alpha_2(v)) = u + v$ on the entire $n \times n$ grid, both maps are permutations. Writing $p = \alpha_1^{-1}$ and $q = \alpha_2^{-1}$, $T(a, b) = p(a) + q(b)$. This table equals $K^\star$ only if $p = q = \mathrm{id}$.*

Proof. If $\alpha_1(u) = \alpha_1(u')$, choose any v and compare the two labels. Equality of the table cells gives $u + v = u' + v$, hence $u = u'$. Injectivity on a finite domain implies a permutation; the same holds for the second slot. Substituting $u = p(a)$ and $v = q(b)$ proves the table representation. If $p(a) + q(b) = a + b$ for every pair, then $p(a) - a$ is a constant and $q(b) - b$ is its negative. A permutation of $\{0, \dots, n-1\}$ cannot translate every value by a nonzero constant, so both are identity. The converse construction uses (p^{-1}, q^{-1}) and is immediate. □

Lemma 5 (Displacement histogram)**.** *For a nonidentity permutation p, put $N_p(k) = |\{a : p(a) - a = k\}|$. Then $\sum_k N_p(k)^2 \le (n-1)^2 + 1$.*

Proof. Let $M = \max_k N_p(k)$. If $M = n$, the displacement is constant, forcing the identity. Thus $1 \le M \le n-1$. The counts other than a maximizing count sum to $n - M$, and their sum of squares is at most $(n - M)^2$. Consequently $\sum_k N_p(k)^2 \le M^2 + (n - M)^2$. This convex quadratic is maximized on $[1, n-1]$ at an endpoint, where it equals $(n-1)^2 + 1$. □

Proof of Theorem 3. Lemma 4 gives the table family. An entry agrees with $K^\star$ exactly when $(p(a) - a) + (q(b) - b) = 0$, so the number of agreements is $\sum_k N_p(k) N_q(-k)$. If exactly one permutation is nonidentity, it fixes at most $n-2$ points and there are at most $n(n-2)$ agreeing entries. The edit distance is then at least $2n$. If both are nonidentity, Cauchy–Schwarz and Lemma 5 give at most $(n-1)^2 + 1$ agreements, hence distance at least $2(n-1)$.

Take $p(a) = a + 1 \bmod n$ and $q(b) = b - 1 \bmod n$. Their displacement counts are $(n-1, 1)$ with opposite signs, so their number of agreements is exactly $(n-1)^2 + 1$. This proves attainment. For any supplied table with $t = \mathrm{ed}(K, K^\star)$, the triangle inequality gives $\mathrm{ed}(K, T) \ge 2(n-1) - t$ for every alternative zero-loss table. When $t \le n-2$, this exceeds t, so the identity explanation is uniquely minimum-edit.

For tightness, choose the attaining alternative table and change exactly $n-1$ of its $2(n-1)$ disagreements in $K^\star$ to its labels. The resulting K is distance $n-1$ from both tables. The same lower bound excludes any closer alternative, so $e^\star = e_{\mathrm{alt}} = n - 1$. □

The proof uses the ordered integer addition table, not modular addition. It also uses full support, separate maps, and edits counted over all cells. The comparison table is not cancellative, so the addition injection argument cannot simply be reused for it; there, the diagonal gives injectivity instead.

A.4 Comparison distance

Proposition 6 (Comparison repair distance)**.** *Let $K^\star(u, v) = \mathrm{sign}(u - v)$ (coded $<, =, >$) on $\{0, \dots, n-1\}^2$, $n \ge 2$, with full support, separate slot maps, and unweighted edits. Every zero-loss explanation uses two permutations g, h with $T(a, b) = K^\star(g^{-1}(a), h^{-1}(b))$. Its edit distance from $K^\star$ is $2\,\mathrm{inv}(g)$ when $g = h$, where* inv *counts inversions, and at least $2|\{u : h^{-1}g(u) \ne u\}| \ge 4$ otherwise. Hence $e_{\mathrm{alt}} = 2$ for clean comparison, attained exactly when $g = h$ is an adjacent transposition, and a tie $e^\star = e_{\mathrm{alt}} = 1$ exists at one corruption.*

Proof. Write $g = \alpha_1$ and $h = \alpha_2$. If $g(u) = g(u')$ with $u < u'$, the pairs (u, u) and (u', u) read the same cell $(g(u), h(u))$ but require labels $=$ and $>$. Hence g is injective, so a permutation, and likewise h. Every cell is then hit exactly once, and $T(g(u), h(v)) = K^\star(u, v)$ determines T. The edits are the pairs (u, v) with $K^\star(u, v) \ne K^\star(g(u), h(v))$. Put $\sigma = h^{-1}g$. A pair $(u, \sigma(u))$ with $\sigma(u) \ne u$ lands on the diagonal, since $g(u) = h(\sigma(u))$, although $u \ne \sigma(u)$; a pair (u, u) with $\sigma(u) \ne u$ has label $=$ but lands off the diagonal. The two sets are disjoint, so $\sigma \ne \mathrm{id}$ gives at least $2|\{u : \sigma(u) \ne u\}| \ge 4$ edits. If $\sigma = \mathrm{id}$, then $g = h \ne \mathrm{id}$, and an ordered pair is an edit exactly when g reverses its order. This gives $2\,\mathrm{inv}(g) \ge 2$ edits, with equality exactly for adjacent transpositions, which change only the cells $(i, i+1)$ and $(i+1, i)$. Copying one of these two labels into $K^\star$ gives a table at distance one from both explanations. □

Exhaustive enumeration of all n^{2n} map pairs for $n = 2, \dots, 5$ (up to 9.8×10^6 pairs; `scripts/rev_b4_compare_check.py`) confirms every statement, including that no pair containing a non-permutation has zero loss; the ten-value CP-SAT certificate gives $e_{\mathrm{alt}} = 2$ as

well. Unlike addition, comparison has uniform unique-repair radius zero: a single corrupted cell can tie with reordering two adjacent values.

A.5 Minimum edit as a maximum-a-posteriori explanation

Let $|\mathcal{Y}| \geq 2$ and suppose each cell of the intended table T is corrupted independently with probability ρ to a label drawn uniformly from the $|\mathcal{Y}| - 1$ wrong ones. Then $P(K \mid T) = (\rho/(|\mathcal{Y}| - 1))^{\mathrm{ed}(K,T)}(1-\rho)^{n^2 - \mathrm{ed}(K,T)}$, which for $0 < \rho < (|\mathcal{Y}| - 1)/|\mathcal{Y}|$ is strictly decreasing in $\mathrm{ed}(K, T)$. The two boundary cases are degenerate rather than informative: at $\rho = 0$ only explanations with $T = K$ have nonzero likelihood, so if no such zero-loss explanation exists the posterior is undefined, and at $\rho = (|\mathcal{Y}| - 1)/|\mathcal{Y}|$ the likelihood does not depend on the edit distance at all. With a uniform prior over zero-loss explanations (α, T) and ρ in the open interval, the posterior mode is therefore the minimum-edit explanation, and $\Delta > 0$ states that the intended program is the unique mode. Two departures change the metric but not the method. Cell-specific rates $\rho_{ab} \in (0, (|\mathcal{Y}| - 1)/|\mathcal{Y}|)$ give weights $w_{ab} = \log[(1 - \rho_{ab})(|\mathcal{Y}| - 1)/\rho_{ab}]$ and a weighted edit distance; the CP-SAT objective is linear in the edit indicators, so weighted certificates need no new machinery, but Theorem 3 and Proposition 6 are proved for unit weights only. Structured corruption, such as shifting a whole row, is not independent across cells; the uniform model can then rank explanations differently from a structured prior, which is one reading of the tied structured ten-edit table in Section 7.2. A non-uniform prior over maps would likewise add a map cost to the objective. We report unweighted certificates throughout and make no robustness claim for these alternatives.

B Implementation and experimental protocol

Seeds per study. Default tables use five training seeds. The independent-table study uses ten tables per edit count and three seeds per table; the boundary study uses five constructed tables per margin and three seeds each. The initial repair ablation uses three seeds per condition, and its final twenty-edit comparison uses ten. Budget and second-encoder controls use five seeds, are separate controls rather than an architecture-by-budget grid, and keep the ten-epoch protocol unless stated. The oracle control (E18) pretrains both SlotCNN encoders on true concepts for ten epochs, five seeds per domain. Seed counts for the additional experiments are given with each study in Appendix D.

Datasets and splits. MNIST uses its official training and test pools. A permutation with seed 777 splits the training pool into 50,000 training and 10,000 validation images. SVHN uses the official train and test splits, with 61,047 and 12,210 images in the derived training and validation pools; the extra split is unused. For each pair, a supported concept pair is drawn uniformly, then an image is drawn independently from each requested digit class. Pair sampling is with replacement. The test-pair seed is 12345. These choices are implemented in `wrongk/data.py`. MNIST normalization uses mean 0.1307 and standard deviation 0.3081. SVHN uses grayscale conversion, bilinear resizing to 28×28, and normalization by mean 0.4455 and standard deviation 0.1943, computed from the transformed official training split. The same encoder is used across domains within each experiment. E17 replaces SlotCNN with SmallResNet in both domains. Its stem has 32 channels, followed by two residual blocks, a stride-two convolution to 64 channels, two further blocks, batch normalization, global average pooling, and a ten-way linear output. Each block uses two pre-activation convolutions.

Training and initialization. The SlotCNN convolutions use unit stride and no padding. After the second pooling, the $64 \times 5 \times 5$ representation feeds the 128-unit layer. There is no shared encoder in the reported main experiments. Each seed initializes both PyTorch and NumPy; optimizers use the code's default Adam settings apart from the reported learning rates. Learned table logits have shape $10 \times 10 \times 19$ for addition. Random-logit initialization samples each logit from $\mathcal{N}(0, 1)$. Random-table initialization samples a label uniformly per cell and assigns it logit six, leaving other logits zero. With zero penalty, these controls do not use K in their predictions or objective. Replicating them under different corruption identifiers does not create independent knowledge conditions.

Oracle encoder pretraining minimizes cross-entropy against true concepts for ten epochs; witness pretraining uses a certified remapping of those labels. Its checkpoint selection maximizes validation accuracy against those concept targets. The subsequent task-only training uses the same ten-epoch

protocol as scratch training. Oracle and witness runs therefore have more supervision and compute. The free baseline predicts labels from concatenated CNN features and has no concept-accuracy interpretation.

Corruption definitions. The default permutation swaps 3 with 5 and 7 with 9. For `flip_`t, NumPy selects t cells without replacement; each replacement is sampled uniformly from the labels other than the correct label. A suffix `@c`s specifies corruption seed s; no suffix uses seed zero. The `flip_struct` table changes every cell in row three by adding two modulo the number of labels. `shift` adds one everywhere modulo that number. For comparison, `reverse` exchanges less-than and greater-than labels while preserving equality. Random permutation negatives in the detection study use independent slot permutations as specified in `wrongk/knowledge.py`.

Numerical reference and solver interpretation. The tabular reference optimizes two 10×10 row-stochastic channels with Adam. Its objective is the clipped expected log-loss on uniform supported concept pairs, using exact label marginalization. The routine compares soft channels and their near-deterministic argmax versions, retaining the best feasible value across restarts. Because the problem is nonconvex, this gives an upper bound on the minimum only. It is not a lower bound on neural loss: clipping precludes invoking Jensen to identify the two objectives.

We use the CP-SAT solver in Google OR-Tools (Perron et al., 2023), combining constraint programming with satisfiability-based search. The symbolic solver accepts weighted supports and exposes rounding corrections, but the principal tables use exact uniform integer counts. Enumeration is complete only when the saved status says so. For an incomplete minimization, the incumbent gives an upper bound and the solver objective bound gives a lower bound. We use the corresponding safe direction for maximization. The comparison `flip_20` and `shift` violation intervals are $[0.08, 0.09]$ and $[0.14, 0.22]$ in the saved artifact; these are not exact point values. A sparse-repair witness is checked against all supported pairs and its actual table edit count.

B.1 Boundary construction and selection

Let $D = \{(a, b) : T_{p,q}(a, b) \neq K^\star(a, b)\}$. Select s_1 cells in D and assign the alternative label, s_2 other cells in D and assign a third label, and s_3 cells outside D and assign any wrong label, with $s_1 + s_2 + s_3 = 10$. Then

$$\mathrm{ed}(K, T_{p,q}) = |D| - s_1 + s_3, \qquad \Delta \leq |D| - 2s_1 - s_2.$$

This is only an upper bound on the actual margin because another alternative may be closer. CP-SAT recertifies e_{alt} for each construction. The permutation catalogue uses cyclic shifts and shifts composed with a transposition; construction randomness is seeded from a stable hash of its key. The saved manifest contains 114 candidates. Actual certified margins, not requested target margins, determine the bins; the first five candidates by construction index within each bin are used; the other 64 candidates were not trained. The 50 trained table arrays are distinct and every one has exactly ten wrong cells. The saved certificate records report closed optimal bounds and validated witnesses for all 50. Three additional spot solves at margins zero, one, and fifteen also returned optimal status. This does not claim a fresh solve of all fifty instances during manuscript preparation.

Uncertainty and reuse. Five seeds at one table estimate training variability for that table; ten corruption seeds estimate a different source of variation. The E9 intervals average the three training seeds first and apply a Student t interval to ten table means, with nine degrees of freedom. The boundary experiment is a structured collection and we show its table rates descriptively. Its pooled seed correlation and pooled Fisher test are not treated as inferential evidence. The original prediction-agreement bootstrap resamples common test examples while keeping five trained model pairs fixed. All intervals are conditional on the image protocol and are not corrected into claims over unseen tasks.

C Extended experimental results

C.1 Addition: fixed tables, numerical reference, and repair

Tables 5–7 give the complete original addition conditions. Three five-edit tables use distinct corruption seeds. Random flip counts have positive deterministic floors here, while both permutation conditions admit exact compensation. The structured corruption has a minimum-edit tie and was trained; it should not be described as an untested tie fixture. The five clean controls in the repair table are not nontrivial repairs. The constructed nine-edit fixture establishes the theoretical tight radius; the neural boundary study instead holds the edit count at ten.

Table 5: Full fixed-addition sweep, five seeds per row. The signed hard-minus-soft column separates argmax correctness from probability mass.

Knowledge	Label	Joint hard	Joint soft	Hard − soft
clean	0.975 ± 0.003	0.975 ± 0.003	0.968 ± 0.006	0.007
perm_both	0.975 ± 0.003	0.358 ± 0.002	0.357 ± 0.001	0.001
perm_one	0.976 ± 0.002	0.583 ± 0.001	0.580 ± 0.000	0.003
flip_2	0.954 ± 0.002	0.973 ± 0.002	0.927 ± 0.006	0.045
flip_5	0.927 ± 0.001	0.974 ± 0.001	0.870 ± 0.017	0.104
flip_5@c1	0.921 ± 0.003	0.969 ± 0.004	0.867 ± 0.014	0.102
flip_5@c2	0.922 ± 0.003	0.969 ± 0.002	0.859 ± 0.016	0.110
flip_10	0.869 ± 0.003	0.963 ± 0.004	0.768 ± 0.022	0.195
flip_20	0.730 ± 0.008	0.890 ± 0.014	0.591 ± 0.026	0.299
flip_struct	0.873 ± 0.001	0.832 ± 0.056	0.748 ± 0.039	0.083
shift	0.874 ± 0.005	0.099 ± 0.002	0.081 ± 0.003	0.017

Table 6: Addition: certified deterministic violation floor, feasible numerical reference, and measured clipped losses in nats. The numerical column is not a certified optimum. Measured columns give mean $\pm$ sample standard deviation.

Knowledge	δ_{det}	Numerical ref.	Train $\widehat{R}_B$	Test $\widehat{R}_B$
clean	0.000	0.000	0.018 ± 0.012	0.081 ± 0.006
perm_both	0.000	0.000	0.011 ± 0.004	0.085 ± 0.009
perm_one	0.000	0.000	0.014 ± 0.002	0.082 ± 0.010
flip_2	0.020	0.130	0.134 ± 0.011	0.216 ± 0.008
flip_5	0.050	0.295	0.301 ± 0.015	0.381 ± 0.006
flip_5@c1	0.050	0.308	0.299 ± 0.016	0.407 ± 0.013
flip_5@c2	0.050	0.295	0.302 ± 0.015	0.387 ± 0.005
flip_10	0.100	0.508	0.524 ± 0.012	0.629 ± 0.016
flip_20	0.200	0.957	0.981 ± 0.016	1.161 ± 0.010
flip_struct	0.100	0.358	0.379 ± 0.027	0.488 ± 0.013
shift	0.100	0.358	0.377 ± 0.015	0.464 ± 0.010

Table 7: Discrete edit costs and exact neural table recovery, five seeds per row. A positive gap identifies the minimum-edit explanation, not the optimizer's outcome.

Knowledge	$e^\star$	e_{alt}	Identifiable	Exact table
`clean`	0	18	Yes	5/5
`perm_both`	56	0	No	0/5
`perm_one`	40	0	No	0/5
`flip_2`	2	20	Yes	5/5
`flip_5`	5	21	Yes	5/5
`flip_5@c1`	5	21	Yes	5/5
`flip_5@c2`	5	21	Yes	5/5
`flip_10`	10	23	Yes	5/5
`flip_20`	20	31	Yes	4/5
`flip_struct`	10	10	No	0/5
`shift`	100	10	No	0/5

Table 8: Seed-level addition under `shift`. Opposite slot-specific modes make a pooled slot-1 mean misleading.

Seed	Slot 1 hard	Slot 2 hard	Joint hard
0	0.9842	0.1026	0.1011
1	0.9836	0.1018	0.1000
2	0.9826	0.1020	0.1000
3	0.0976	0.9849	0.0960
4	0.0978	0.9856	0.0958

The first three shift seeds preserve high slot-1 correctness and low slot-2 correctness; the remaining two reverse the pattern. Joint accuracy stays low in both modes. Aggregating each slot alone would conceal this mechanism.

Prediction agreement under permutation. Paired clean and permuted MNIST models agree on 0.9761 of label predictions but on only 0.3570 of joint concept predictions; the 95% bootstrap interval for their label-accuracy difference, resampling test examples, is $[-0.00154, 0.00090]$. SVHN reproduces the dissociation (label 0.7124 versus 0.7110; hard concept accuracy 0.7102 versus 0.2762), as does the second encoder reported in Appendix C.8. Under random corruption the two concept metrics diverge instead: on MNIST `flip_20`, hard correctness is 0.8897 but soft mass only 0.5909. All 1,150 checkpoints with saved soft metrics satisfy Proposition 2 pointwise and in aggregate.

C.2 Boundary and independent-table variation

Table 9 complements Figure 2; table standard deviation refers to the five three-seed recovery fractions. The E9 experiment in Table 10 comprises thirty independently sampled wrong tables and ninety training outcomes. Fifteen outcomes reuse original runs, and seventy-five are additional training runs. The ten-edit outlier is `flip_10@c6`; its loss discrepancy is retained in the analysis.

Table 9: Ten wrong cells throughout. Exact recovery counts and sample SD of five per-table recovery fractions, each based on three seeds.

Δ	Recovered	Mean rate	Table SD	Five table recovery rates
0	3/15	0.200	0.298	0.00, 0.00, 0.00, 0.33, 0.67
1	9/15	0.600	0.435	0.00, 0.33, 0.67, 1.00, 1.00
2	9/15	0.600	0.365	0.00, 0.67, 0.67, 0.67, 1.00
3	9/15	0.600	0.435	0.00, 0.33, 0.67, 1.00, 1.00
5	14/15	0.933	0.149	0.67, 1.00, 1.00, 1.00, 1.00
8	15/15	1.000	0.000	1.00, 1.00, 1.00, 1.00, 1.00
10	14/15	0.933	0.149	0.67, 1.00, 1.00, 1.00, 1.00
12	14/15	0.933	0.149	0.67, 1.00, 1.00, 1.00, 1.00
14	15/15	1.000	0.000	1.00, 1.00, 1.00, 1.00, 1.00
15	15/15	1.000	0.000	1.00, 1.00, 1.00, 1.00, 1.00

Table 10: Independent-table variation: ten corruption tables per edit count, three seeds each. Intervals are 95% Student t intervals over ten table means, with nine degrees of freedom. The floor is constant within edit count.

t	Label	Joint hard	Joint soft	Test $\widehat{R}_B$
5	0.923 [0.922, 0.925]	0.970 [0.969, 0.972]	0.870 [0.863, 0.878]	0.392 [0.377, 0.407]
10	0.863 [0.854, 0.871]	0.960 [0.956, 0.964]	0.746 [0.699, 0.793]	0.728 [0.596, 0.861]
20	0.738 [0.724, 0.752]	0.907 [0.868, 0.946]	0.588 [0.564, 0.612]	1.125 [1.048, 1.203]

C.3 Initialization and penalty ablations

The full metrics below report the initial three-seed ablations. The final ten-seed twenty-edit comparison is reported separately below. Labels, hard accuracy, and soft mass answer different questions for a table initialized without knowledge. In particular, failure to recover the exact table would not alone rule out learning a useful table under a different concept permutation; the low label accuracies delimit that possibility in these runs. The random initializations improve label accuracy over the uniform start but still do not approach the supplied-table initialization in this budget.

Table 11: Full repair-arm metrics for `flip_5`, three seeds per arm. Wrong cells is the mean number differing from $K^\star$.

Arm	Exact	Label	Hard	Soft	Wrong cells
A: fixed K	0/3	0.926	0.973	0.869	5.00
B: learnable at K, no penalty	3/3	0.976	0.976	0.969	0.00
C: uniform start	0/3	0.124	0.008	0.007	90.67
C′: random logits	0/3	0.330	0.013	0.013	92.67
C″: random table	0/3	0.412	0.016	0.020	94.67
D: L1, $\lambda = 10^{-3}$	3/3	0.974	0.974	0.968	0.00
D: L1, $\lambda = 10^{-2}$	0/3	0.928	0.975	0.970	5.00
D: L1, $\lambda = 10^{-1}$	0/3	0.927	0.974	0.966	5.00
E: edit penalty	3/3	0.975	0.975	0.969	0.00
F: oracle $K^\star$	3/3	0.974	0.974	0.965	0.00

Table 12: Full repair-arm metrics for `flip_10`, three seeds per arm. Wrong cells is the mean number differing from $K^\star$.

Arm	Exact	Label	Hard	Soft	Wrong cells
A: fixed K	0/3	0.869	0.963	0.754	10.00
B: learnable at K, no penalty	3/3	0.972	0.972	0.968	0.00
C: uniform start	0/3	0.124	0.008	0.007	90.67
C$'$: random logits	0/3	0.330	0.013	0.013	92.67
C$''$: random table	0/3	0.412	0.016	0.020	94.67
D: L1, $\lambda = 10^{-3}$	3/3	0.974	0.974	0.969	0.00
D: L1, $\lambda = 10^{-2}$	0/3	0.873	0.969	0.860	10.00
D: L1, $\lambda = 10^{-1}$	0/3	0.873	0.968	0.859	10.00
E: edit penalty	3/3	0.973	0.973	0.969	0.00
F: oracle $K^\star$	3/3	0.974	0.974	0.965	0.00

Table 13: Full repair-arm metrics for `flip_20`, three seeds per arm. Wrong cells is the mean number differing from $K^\star$.

Arm	Exact	Label	Hard	Soft	Wrong cells
A: fixed K	0/3	0.735	0.896	0.594	20.00
B: learnable at K, no penalty	1/3	0.911	0.907	0.901	8.00
C: uniform start	0/3	0.124	0.008	0.007	90.67
C$'$: random logits	0/3	0.330	0.013	0.013	92.67
C$''$: random table	0/3	0.412	0.016	0.020	94.67
D: L1, $\lambda = 10^{-3}$	1/3	0.897	0.917	0.870	4.67
D: L1, $\lambda = 10^{-2}$	0/3	0.740	0.919	0.686	20.00
D: L1, $\lambda = 10^{-1}$	0/3	0.723	0.886	0.686	20.00
E: edit penalty	2/3	0.942	0.941	0.938	2.33
F: oracle $K^\star$	3/3	0.974	0.974	0.965	0.00

Table 14: Full repair-arm metrics for `perm_both`, three seeds per arm. Wrong cells is the mean number differing from $K^\star$.

Arm	Exact	Label	Hard	Soft	Wrong cells
A: fixed K	0/3	0.976	0.359	0.357	56.00
B: learnable at K, no penalty	0/3	0.975	0.357	0.356	56.00
C: uniform start	0/3	0.124	0.008	0.007	90.67
C$'$: random logits	0/3	0.330	0.013	0.013	92.67
C$''$: random table	0/3	0.412	0.016	0.020	94.67
D: L1, $\lambda = 10^{-3}$	0/3	0.977	0.358	0.357	56.00
D: L1, $\lambda = 10^{-2}$	0/3	0.977	0.359	0.357	56.00
D: L1, $\lambda = 10^{-1}$	0/3	0.975	0.357	0.356	56.00
E: edit penalty	0/3	0.974	0.357	0.356	56.00
F: oracle $K^\star$	3/3	0.974	0.974	0.965	0.00

Earlier head and reliability controls. The original E3 sweep used the encoder learning rate for the head and did not produce the intended argmax table edits. E3b changes the head learning rate and is the main repair experiment. This empirical change does not establish a general bound on Adam's maximum step or a convergence theorem. The earlier runs remain in the artifact inventory, not pooled with E3b.

The reliability baseline uses a scalar w_{ab} on the supplied label and uniform mass $(1 - w_{ab})/(|\mathcal{Y}| - 1)$ over the other labels. Its regularizer is $0.01 \sum_{a,b}(1 - w_{ab})$ in E3b; logits start at four. Even with correct concepts, a wrong supplied label limits the true label's probability to $1/(|\mathcal{Y}| - 1)$. This head can suppress a rule but cannot concentrate all mass on its correct replacement. It is not an implementation of Safe-ABL or ABL knowledge refinement, and its results cannot establish comparative performance against those systems.

Table 15: E3b uniform-fallback reliability baseline, five seeds per row. This head attenuates supplied labels rather than learning a replacement label.

Knowledge	Label	Joint hard	Joint soft	Test $\widehat{R}_B$
clean	0.975 ± 0.002	0.975 ± 0.002	0.966 ± 0.004	0.085 ± 0.007
flip_10	0.879 ± 0.001	0.975 ± 0.002	0.966 ± 0.005	0.396 ± 0.007
flip_2	0.957 ± 0.003	0.976 ± 0.003	0.970 ± 0.004	0.145 ± 0.007
flip_20	0.765 ± 0.002	0.966 ± 0.004	0.894 ± 0.019	0.830 ± 0.009
flip_5	0.929 ± 0.001	0.976 ± 0.001	0.970 ± 0.004	0.229 ± 0.006
flip_5@c1	0.925 ± 0.002	0.974 ± 0.002	0.969 ± 0.002	0.241 ± 0.004
flip_5@c2	0.928 ± 0.003	0.975 ± 0.004	0.971 ± 0.004	0.233 ± 0.013
flip_struct	0.872 ± 0.003	0.832 ± 0.054	0.752 ± 0.043	0.477 ± 0.022
perm_both	0.976 ± 0.002	0.357 ± 0.001	0.356 ± 0.001	0.085 ± 0.005
perm_one	0.974 ± 0.001	0.582 ± 0.002	0.579 ± 0.002	0.082 ± 0.003
shift	0.874 ± 0.004	0.099 ± 0.002	0.089 ± 0.002	0.446 ± 0.007

Table 16: Addition initialization bracket. Oracle pretraining uses true concepts; witness pretraining uses the corresponding certified remapping. Five seeds per row; both add supervision and compute before task-only training.

Knowledge	Initialization	Seeds	Label	Joint hard	Joint soft
flip_5	oracle	5	0.927 ± 0.002	0.973 ± 0.002	0.867 ± 0.007
perm_both	oracle	5	0.977 ± 0.003	0.359 ± 0.001	0.358 ± 0.001
perm_both	witness	5	0.976 ± 0.003	0.358 ± 0.002	0.357 ± 0.002
perm_one	oracle	5	0.975 ± 0.003	0.584 ± 0.001	0.582 ± 0.001
perm_one	witness	5	0.976 ± 0.004	0.583 ± 0.004	0.582 ± 0.003
shift	oracle	5	0.875 ± 0.003	0.099 ± 0.003	0.081 ± 0.002

Final ten-seed comparison (E16). The final comparison uses one fixed twenty-edit table, ten seeds per arm, and the same optimizer and budget (Table 17). The edit penalty recovers 8/10, no penalty 6/10, L1 at 10^{-3} 2/10, and the oracle 10/10. The saved exploratory Fisher and wrong-count Mann–Whitney comparisons give $p = 0.023$ and 0.026 for edit versus L1, but $p = 0.63$ and 0.21 for edit versus no penalty. These are nominal, unadjusted comparisons on one table; the arms share seed identifiers and the reported tests do not exploit pairing. They do not establish a general penalty advantage, necessity, or equivalence to no penalty. Repeated seeds measure optimization variability, not a population of wrong tables.

Table 17: Final twenty-edit ablation, ten seeds on one table. Wrong is mean number of table entries differing from $K^\star$. Oracle correctness is fixed by supplying $K^\star$, not learned table recovery.

Arm	Exact	Wrong	Label	Hard	Soft
B no penalty	6/10	4.8	0.9276	0.9248	0.9200
D L1 $\lambda = 10^{-3}$	2/10	4.4	0.9175	0.9519	0.9099
E edit penalty $\lambda = 10^{-2}$	8/10	1.3	0.9563	0.9560	0.9515
F oracle $K^\star$	10/10	0.0	0.9746	0.9746	0.9659

C.4 COMPARISON AND RESTRICTED SUPPORT

Table 18: Fixed comparison, five seeds per condition. Large seed variability in the clean condition is part of the finding.

Knowledge	Label	Joint hard	Joint soft
clean	0.945 ± 0.010	0.150 ± 0.211	0.153 ± 0.209
perm_both	0.943 ± 0.014	0.068 ± 0.057	0.066 ± 0.056
perm_one	0.935 ± 0.015	0.110 ± 0.144	0.108 ± 0.138
reverse	0.942 ± 0.001	0.005 ± 0.005	0.005 ± 0.005
flip_2	0.936 ± 0.014	0.244 ± 0.190	0.244 ± 0.176
flip_5	0.921 ± 0.007	0.038 ± 0.029	0.041 ± 0.028
flip_10	0.894 ± 0.006	0.105 ± 0.057	0.116 ± 0.075
flip_20	0.866 ± 0.007	0.059 ± 0.017	0.061 ± 0.014
flip_struct	0.939 ± 0.009	0.069 ± 0.049	0.071 ± 0.048
shift	0.780 ± 0.003	0.008 ± 0.004	0.008 ± 0.002

Table 19: Comparison initialization bracket. The ceiling $\Phi(0)$ applies at zero risk; range denotes the five hard concept accuracies. Initialized runs use additional concept supervision and compute.

Knowledge	$\Phi(0)$	Init.	Label	Joint hard	Range	Test $\widehat{R}_B$
clean	1.00	scratch	0.945 ± 0.010	0.150 ± 0.211	[0.02, 0.52]	0.124
clean	1.00	oracle init	0.989 ± 0.002	0.977 ± 0.004	[0.97, 0.98]	0.042
reverse	0.00	scratch	0.942 ± 0.001	0.005 ± 0.005	[0.00, 0.01]	0.127
reverse	0.00	oracle init	0.874 ± 0.008	0.021 ± 0.011	[0.01, 0.04]	0.267
perm_both	0.36	scratch	0.943 ± 0.014	0.068 ± 0.057	[0.01, 0.16]	0.124
perm_both	0.36	oracle init	0.935 ± 0.018	0.442 ± 0.066	[0.36, 0.51]	0.146

Under reversed knowledge, initialized encoders leave the identity but reach worse task loss than scratch training. Under permuted knowledge, they retain more correct concept mass at nonzero loss. Neither behavior can be assessed by substituting the zero-risk ceiling directly for a finite-risk bound.

Restricted support. E6 samples both even–even and odd–odd digit pairs, giving fifty supported pairs. Its complement contains mixed-parity pairs. This differs from rsbench's MNAdd-EvenOdd setting with single-parity digit domains (Bortolotti et al., 2024). All ten scratch empirical maps have violation fraction 0.100 on the supported grid. Fifteen initialization-bracket runs have zero map violation. All twenty forty- and one-hundred-epoch restricted-support controls (E6L) select weights exactly equal, tensor by tensor, to their corresponding ten-epoch checkpoints; test NLL, validation NLL, and label accuracy also agree to six decimals. Thus their selected solutions were already reached within ten epochs. No per-epoch traces were saved for these controls: they show no improvement under best-validation selection, but provide no evidence about later iterates or whether training left and returned to the same solution. These empirical maps are class-averaged argmax maps; their violation score is different from per-image hard accuracy.

Table 20: Same-parity addition: scratch and initialized runs. Concept metrics remain per-image joint quantities; map violation is a separate diagnostic.

Run group	Knowledge	Init.	Seeds	Label	Joint hard	Joint soft
E6	clean	scratch	5	0.875 ± 0.005	0.486 ± 0.002	0.480 ± 0.002
E6	perm_both	scratch	5	0.873 ± 0.006	0.299 ± 0.255	0.297 ± 0.252
E6i	clean	oracle	5	0.975 ± 0.003	0.975 ± 0.003	0.971 ± 0.006
E6i	perm_both	oracle	5	0.977 ± 0.002	0.503 ± 0.002	0.502 ± 0.002
E6i	perm_both	witness	5	0.974 ± 0.005	0.502 ± 0.003	0.501 ± 0.003

C.5 TRAINING SIZE AND LABEL NOISE

Table 21: Robustness to training-set size and training-label noise, five seeds per condition. Validation and test labels remain clean.

Knowledge	n_{train}	Noise	Label	Joint hard	Joint soft
`flip_5`	3,000	0.0	0.872 ± 0.004	0.915 ± 0.005	0.813 ± 0.010
`flip_5`	10,000	0.0	0.906 ± 0.007	0.952 ± 0.007	0.850 ± 0.013
`flip_5`	30,000	0.1	0.920 ± 0.004	0.965 ± 0.005	0.755 ± 0.009
`perm_both`	3,000	0.0	0.937 ± 0.003	0.349 ± 0.001	0.344 ± 0.002
`perm_both`	10,000	0.0	0.966 ± 0.002	0.356 ± 0.001	0.354 ± 0.002
`perm_both`	30,000	0.1	0.970 ± 0.002	0.356 ± 0.002	0.309 ± 0.009

For each seed, the noisy 30,000-pair setting uses the same pair-sampling seed as its clean E1 counterpart. Noise is applied after sampling from an independent random stream seeded at the run seed plus 10,000. Each corrupted training label is replaced by a different label. Thus this comparison does not also change the pair-sampling seed. The full thirty-checkpoint robustness set has separate soft measurements.

C.6 SVHN: DOMAIN REPLICATION AND REPAIR FAILURE

Preprocessing and the slotwise assumption. Each SVHN crop is converted to grayscale, resized bilinearly (with antialiasing) from 32×32 to 28×28, and normalized with the training-split mean 0.4455 and standard deviation 0.1943. The original bounding boxes are non-square, so the 32×32 crops already stretch the digit, and many crops contain parts of neighbouring digits from the same house number. Because our sampler draws each slot image independently from the images whose SVHN label is c_i, the factorization $x_i \sim P_i(\cdot \mid c_i)$ holds exactly for the constructed pairs, with P_i the preprocessed class-conditional image distribution. What the preprocessing and clutter change is the separability of these distributions: a class-c_i image can show other digits, and label noise in the source annotation becomes concept noise. This lowers attainable concept accuracy (Table 22) without breaking any assumption. The assumption would fail if both slots were cut from the same house-number image, so that one image reveals the other's digit; we never form pairs that way. Only the mixture identity Equation 7 and hence Proposition 1 use the factorization; the hard–soft bridge of Proposition 2 holds pointwise, and the certificates are properties of the tables alone.

Table 22: Fixed addition across MNIST (M) and SVHN (S), five seeds per domain. The certified ceiling and floor are symbolic and therefore identical across domains; measured outcomes need not be. Dashes denote no exact compensation; $\Phi(0) = 0$ there by the empty-set convention.

Knowledge	$\Phi(0)$	δ_{det}	Label M	Label S	Joint M	Joint S	$\widehat{R}_B$ M	$\widehat{R}_B$ S	Seeds S
`clean`	1.00	0.00	0.975	0.712	0.975	0.710	0.081	1.050	5
`perm_both`	0.36	0.00	0.975	0.711	0.358	0.276	0.085	1.049	5
`perm_one`	0.60	0.00	0.976	0.714	0.583	0.440	0.082	1.053	5
`shift`	–	0.10	0.874	0.621	0.099	0.084	0.464	1.351	5
`flip_5`	–	0.05	0.927	0.657	0.974	0.685	0.381	1.310	5
`flip_20`	–	0.20	0.730	0.490	0.890	0.573	1.161	1.880	5

Relative to clean, SVHN loss increments are 0.26, 0.30, and 0.83 nats for five edits, shift, and twenty edits. The corresponding MNIST increments are 0.30, 0.38, and 1.08. The ordering reproduces, but the magnitudes are not equal. The five- and twenty-edit repair outcomes differ under the same certificates; the negative twenty-edit SVHN result is retained rather than excluded from the domain comparison.

Table 23: Original domain repair comparison, five seeds per condition. M/S are mean joint hard concept accuracies on MNIST/SVHN.

Knowledge	Ident.	Exact MNIST	Exact SVHN	Joint M	Joint S
`clean`	Yes	5/5	5/5	0.974	0.710
`perm_both`	No	0/5	0/5	0.357	0.276
`flip_5`	Yes	5/5	5/5	0.976	0.711
`flip_20`	Yes	4/5	0/5	0.956	0.637

C.7 Loss-scale dynamics of the repair objective

Both terms of Equation 6 are summed before each Adam step on the table logits. A gradient-norm ratio alone does not fix the direction of an Adam update, which also depends on the sign and on each coordinate's history, so we measure both the gradients and the resulting trajectory of the soft table. Training uses the unclipped NLL; the clip $B = 10$ enters only evaluation. For the five-seed `flip_20` SlotCNN repairs on MNIST and SVHN, we rebuilt each model at its initialization and at its selected checkpoint and measured gradients on ten fixed training batches without updating any parameter (`scripts/rev_b5_gradients.py`). At initialization, each cell starts with $Q(K(a,b) \mid a,b) = 0.957$, and the penalty gradient per cell is $\lambda\, Q(1-Q)\sqrt{1+1/18} = 4.2\times10^{-4}$; over the twenty corrupted cells its norm is 0.0019 in every run. The task-loss gradient on the same cells has norm 0.0010–0.0012, and its median per-cell ratio to the penalty gradient is 0.40–0.46 across all ten runs. The task loss instead acts on the encoders, with gradient norm 0.34–0.39 (MNIST) and 0.20–0.22 (SVHN), two orders of magnitude larger.

To see what the optimizer actually does, we replayed the first three epochs of these repairs with the same data, initialization, and Adam settings on CPU, logging the soft table every ten steps (235 steps per epoch, logged every ten steps; `scripts/rev2_r6_adam_replay.py`). These are replays of the protocol, not bit-for-bit copies of the saved accelerator runs. On SVHN, the mean mass on the supplied label of the twenty corrupted cells first *rises*, from 0.957 to 0.996 by the end of epochs one and two, with no corrupted cell below its initial value. The first cell weakens only at step 490–600 (epoch three), when training joint concept accuracy is 0.27–0.53. By the end of epoch three the supplied-label mass is 0.77–0.86 and the true-label mass 0.13–0.20. On MNIST, the first corrupted cell weakens at step 120–160 of epoch one, where concept accuracy is already about 0.8–0.9. The true label overtakes the supplied one on average at step 430–570, and by the end of epoch three it carries 0.54–0.76. So the soft table does move in epoch one, even though no argmax edit occurs: at first it moves toward the supplied labels. The repair starts only after perception has formed, and on SVHN it starts two epochs later and from weaker concepts. This is the opposite of the intuition that large initial task-loss gradients overwhelm the penalty on the table. The measurements are descriptive and do not by themselves establish why SVHN then collapses.

Table 24: Gradient norms on ten training batches, median over five seeds. "Cor." restricts to the twenty corrupted table cells; ratio is the median per-cell ratio of task-loss to penalty gradient.

Domain	State	Task loss, cor.	Penalty, cor.	Ratio	Task loss, encoders
MNIST	initialization	0.0011	0.0019	0.43	0.371
MNIST	selected	0.0072	0.0051	0.93	0.781
SVHN	initialization	0.0011	0.0019	0.44	0.206
SVHN	selected	0.0178	0.0071	0.74	1.962

C.8 Longer-budget and second-encoder controls

E13 retains the twenty-edit table, $e^\star = 20$, $e_{\mathrm{alt}} = 31$, the SlotCNN architecture, optimizer, head learning rate, and data protocol. Ten-epoch runs are reused from E3b/E12; forty and one hundred epochs contribute twenty additional records. Five seeds repeat at each domain and budget; these are not independent draws of corruption tables.

Table 25: Budget control: exact recoveries and mean wrong cells refer to the best-validation checkpoint. The ten-epoch controls are reused.

Domain	Epochs	Exact	Wrong cells	Label	Test $\widehat{R}_B$
MNIST	10	4/5	1.4	0.9561	0.1296
MNIST	40	4/5	1.4	0.9576	0.1299
MNIST	100	4/5	1.4	0.9582	0.1298
SVHN	10	0/5	6.6	0.6435	1.1714
SVHN	40	0/5	6.6	0.6429	1.1761
SVHN	100	0/5	6.6	0.6429	1.1761

The selected SVHN heads leave six or seven wrong entries at each budget. Their shared residual cells are $(5,2),(5,3),(5,5),(5,7),(5,9),(6,1)$. The forty- and hundred-epoch SVHN summaries agree because validation selects the same early checkpoints. This does not mean training stopped changing the head: per-epoch wrong counts continue oscillating between six and seven. Among the hundred-epoch SVHN runs, four last count changes occur at epochs 97–100; the remaining run last changes at epoch 66. Validation loss and discrete table recovery are different selection criteria; no exact recovery occurs in the stored SVHN traces. Reading the stored traces at their last epoch instead of the selected checkpoint gives the same counts (`scripts/rev_b6_traces.py`): at epoch 100, 4/5 MNIST and 0/5 SVHN tables are exact, and likewise at epoch 40. Every recovering MNIST run is exact from epoch three or four onward, in 97 or 98 of its 100 epochs. No SVHN run is exact at any of its 100 epochs; its fewest wrong cells at any epoch is six, and its last ten epochs average 6.0–6.9 wrong cells. Within this budget, longer training does not escape the collapsed state. The restricted-support long-budget controls (E6L) have no learnable table and saved no traces, so they do not bear on this question.

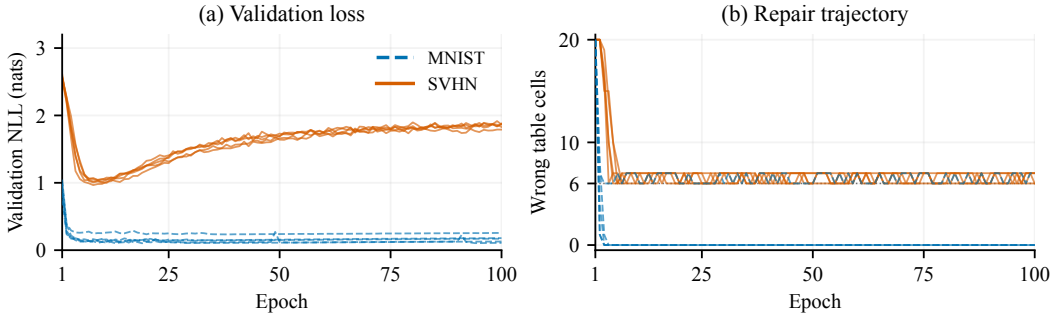


Figure 3: Saved 100-epoch SlotCNN trajectories on the identifiable twenty-edit table ($e^\star = 20$, $e_{\text{alt}} = 31$, $\Delta = e_{\text{alt}} - e^\star = 11$), five runs per domain; each line is one run. The failing MNIST trace overlaps the SVHN band.

Table 26: Two encoders on the original twenty-edit table, five seeds per condition, ten epochs; all columns are primary metrics. Fixed-table label/hard concept means are separate from exact repair counts.

Domain	Encoder	Clean label	Perm. label	Perm. hard	Exact repair
MNIST	SlotCNN	0.9750	0.9753	0.3581	4/5
MNIST	SmallResNet	0.9794	0.9821	0.3594	5/5
SVHN	SlotCNN	0.7124	0.7110	0.2762	0/5
SVHN	SmallResNet	0.8481	0.8497	0.3209	0/5

Table 27: Complete SmallResNet results (E17), five seeds per row, mean ± sample SD. Repair uses the twenty-edit table; other rows keep the table fixed.

Domain	Condition	Label	Hard	Soft	Test $\widehat{R}_B$
MNIST	repair	0.981 ± 0.003	0.981 ± 0.003	0.967 ± 0.003	0.078 ± 0.010
MNIST	clean	0.979 ± 0.004	0.979 ± 0.004	0.967 ± 0.005	0.068 ± 0.010
MNIST	flip_20	0.753 ± 0.007	0.939 ± 0.019	0.616 ± 0.018	1.127 ± 0.015
MNIST	perm_both	0.982 ± 0.003	0.359 ± 0.001	0.356 ± 0.002	0.061 ± 0.010
SVHN	repair	0.736 ± 0.012	0.737 ± 0.006	0.621 ± 0.027	0.794 ± 0.070
SVHN	clean	0.848 ± 0.006	0.848 ± 0.006	0.774 ± 0.016	0.510 ± 0.017
SVHN	flip_20	0.556 ± 0.012	0.644 ± 0.021	0.362 ± 0.019	1.633 ± 0.029
SVHN	perm_both	0.850 ± 0.011	0.321 ± 0.006	0.296 ± 0.006	0.508 ± 0.033

E17 changes the encoder at the original ten-epoch budget. On SVHN the five selected repair heads leave 7, 6, 6, 8, and 6 wrong cells, respectively. All share the six cells listed above. Seed zero additionally misses $(6, 8)$; seed three additionally misses $(0, 3)$ and $(0, 6)$. Thus the common residual structure reproduces, but the entire wrong-cell set is not identical in every run. The single failed SlotCNN MNIST run contains those six plus $(6, 8)$. Five common errors lie in the most corrupted row. Its clean SVHN digit accuracy is about 0.859, which does not single it out as the least legible class; this observation does not exclude perception–optimization interactions or establish a causal explanation of repair failure.

C.9 Per-digit concept collapse and the oracle-pretrained control

`scripts/analyze_digit_collapse.py` reloads every saved `flip_20` edit-head checkpoint from E3b, E12, E13, E16 (edit-penalty arm) and E17, rebuilds the fixed test split, and records per-digit hard concept accuracy for both slots, the slot-1 prediction distribution of true fives, and the mean slot-1 probability of concept 5. It performs inference only. Longer budgets often reselect a state already on record, so tensor hashes are used to count distinct selected states. Table 28 summarizes the 50 checkpoints.

Table 28: Slot-1 concepts in all saved `flip_20` repair checkpoints, distinct selected states in parentheses. Columns give the range of slot-1 accuracy on true fives, the range of the fraction of true fives predicted as 6, and the lowest slot-1 accuracy on any other digit.

Domain	Encoder	Outcome	Ckpts	Acc. on 5	$5 \to 6$	Other min.
MNIST	SmallResNet	repaired	5 (5)	0.992–0.997	0.000–0.002	0.942
MNIST	SlotCNN	failed	5 (3)	0.000–0.000	0.991–0.997	0.968
MNIST	SlotCNN	repaired	20 (12)	0.984–0.998	0.002–0.006	0.963
SVHN	SmallResNet	failed	5 (5)	0.000–0.000	0.924–0.960	0.787
SVHN	SlotCNN	failed	15 (6)	0.000–0.000	0.873–0.918	0.706

In every failing checkpoint no test image is assigned to slot-1 concept 5, and the mean slot-1 probability of concept 5 is at most 0.0013 (the maximum on any single image is 0.2424, rounded). These test-set probabilities do not measure training gradients; absent argmax assignments do not make the soft head's label gradient identically zero. The other slot-1 digits remain recognized. All failing heads keep the corrupted labels of the six common residual cells. In the runs with exactly one further error, the extra cell is $(6, 8)$, whose label 13 equals the sum for a true five; the single run with two further errors leaves $(0, 3)$ and $(0, 6)$ at their corrupted labels. These are observations on one corruption table and do not by themselves show that the collapse causes the failure.

Oracle-pretrained control (E18). The protocol in `results/audit/e18_freeze.json` (specification-body SHA-256 prefix `49e60a7f`) records an intended pre-training freeze, but it was not committed before training. Its timestamp alone does not establish immutable preregistration; this is an exploratory oracle control. For each domain and seed 0–4, a `concept_oracle` parent trains both SlotCNN encoders on true concepts for ten epochs, selected by validation concept accuracy; the child then runs the unchanged edit-head repair of the same twenty-edit table, initialized

from the parent encoders, for ten epochs with per-epoch traces. The protocol fixes all outcomes and requires reporting every run. Pretraining adds concept supervision and compute, so this is an oracle control, not a repair method.

Table 29: E18 against the scratch runs on the same table, five seeds each. Slot-1 accuracy on true fives is given as a range over seeds.

Domain	Encoder start	Exact	Wrong cells by seed	Slot-1 acc. on 5	Label
SVHN	scratch	0/5	7, 6, 7, 7, 6	0.000–0.000	0.6435
SVHN	oracle-pretrained	5/5	0, 0, 0, 0, 0	0.784–0.868	0.7208
MNIST	scratch	4/5	7, 0, 0, 0, 0	0.000–0.998	0.9561
MNIST	oracle-pretrained	5/5	0, 0, 0, 0, 0	0.967–0.994	0.9774

All ten children recover $K^\star$ exactly, each by epoch 3 of its trace. Their slot-1 accuracy on true fives is 0.784–0.868 on SVHN and 0.967–0.994 on MNIST, and no child collapses fives into sixes. On SVHN the children reach joint hard concept accuracy 0.709–0.730, below the 0.732–0.747 of the five failing SmallResNet repairs. The control therefore separates the recovered and failed runs by the slot-1 digit-5 state rather than by overall concept accuracy. It does not show that the collapse is the only possible barrier, and its concept supervision is unavailable in weakly supervised training.

C.10 Matched-composition construction and adaptive provenance

E15 contains 45 tables and 135 runs, with three seeds per table. In the $s_1 = 8$ arm, actual margins 0, 2, and 4 each contain five tables. The last five-table construction group, intended to reach margin twelve, recertifies at margins eight (one table), nine (two), and ten (two). Consequently the 0.87 summary pools those five tables and is not a five-table estimate at each of the three margins. At $s_1 = 9$, margins zero and one each have five tables. The fixed-margin arm has five tables at each $s_1 = 2, 5, 6$.

The first arms A and B did not cover ties and were inconclusive. Arm C was added after inspecting them to probe the low-margin region; the final $s_1 = 8$ summary incorporates its added tie constructions. The reported p-values are exploratory, without a correction for this adaptive expansion or for all comparisons. Fixing s_1 removes that specific covariate's variation, but the remaining edit types and alternative-table distance still vary.

Table 30: Matched-composition study: ten edits and three seeds per table. The high-margin construction splits into actual margins 8, 9, and 10; table counts are shown explicitly. Stars hold margin fixed and vary s_1.

s_1	Δ	Tables	Mean per-table recovery	sd	Runs
8	0	5	0.07	0.15	1/15
8	2	5	0.60	0.43	9/15
8	4	5	0.80	0.45	12/15
8	8	1	1.00	–	3/3
8	9	2	0.67	0.47	4/6
8	10	2	1.00	0.00	6/6
9	0	5	0.27	0.28	4/15
9	1	5	0.87	0.30	13/15
2*	8	5	0.93	0.15	14/15
5*	8	5	0.67	0.41	10/15
6*	8	5	1.00	0.00	15/15

At $s_1 = 8$, recovery rises from 0.07 to 0.60, 0.80, and 0.87 at margins 0, 2, 4, and 8–10 (20 tables; Spearman 0.663, $p = 0.0014$); at $s_1 = 9$, margins zero and one give 0.27 and 0.87 (Mann–Whitney $p = 0.0285$). Holding the margin at eight while varying s_1 over 2, 5, and 6 gives 0.93, 0.67, and 1.00, with no monotone trend. Recovery remains associated with margin after controlling edit count and s_1 in this exploratory, adaptively expanded study; the remaining composition variables are not orthogonal.

C.11 INDEPENDENT CERTIFICATE VALIDATION

The saved validator compares CP-SAT with exhaustive enumeration over arbitrary n^n maps per slot on 80 tables at $n = 3, 4$, covering clean, permutation, one-flip, two-flip, and uniformly random tables (eight per family and size). It records zero mismatches at tolerance 10^{-6} for violation, fixed-mass, repair, ceiling Φ, and lower-margin γ quantities. A separate enumeration over all ordered permutation pairs at $n = 2, \ldots, 6$ gives minimum distance $2(n-1)$ and explicit attaining witnesses and ties. The radius checks sample forty corruptions per tested edit count; they do not enumerate every possible corruption. These are implementation checks accompanying the proof, not a replacement for it. We inspected the saved CP-SAT comparison without rerunning those solves; a separate arithmetic check re-enumerated the addition distances for $n = 2, \ldots, 6$.

D CONTROLLED EXPERIMENTS AND REPAIR DIAGNOSTICS

This section examines repair across corruption tables, concept-collapse interventions, margin replication, perception, optimization, and alternative learning methods. The protocols, table certificates, primary tests, and predictions were frozen in two pre-registration files before the corresponding training (`results/revision/prereg.json` and `prereg_c3.json`, with SHA-256 hashes). Exploratory extensions designed after inspecting a result are labeled as such. Every number below is generated from the run records by `scripts/rev_report.py` and checked independently by `scripts/rev_validate.py`. The repair protocol is the paper's: ten epochs, encoder learning rate 10^{-3}, head learning rate 10^{-2}, $\lambda = 0.01$, and table logits initialized at K. The confirmatory family {multi-table SlotCNN, multi-table SmallResNet, read-out patch, margin replication} is Holm-corrected; every other test is exploratory.

D.1 MULTI-TABLE TWENTY-EDIT REPAIR

Tables `flip_20@c100–c109` were drawn with corruption seeds disjoint from the independent-table study, and kept until ten had a certified positive margin. Four of them (`c100–c103`) had already been used by the frozen detector evaluation (E14), so only six are new; Appendix E.1 completes a never-used set of ten. The first ten drawn all qualified, with $e^\star = 20$, $e_{\mathrm{alt}} \in \{30, \ldots, 33\}$, and $\Delta \in \{10, \ldots, 13\}$ (OPTIMAL and validated), so no table was rejected. Both encoders were trained on both domains, five seeds per table (200 runs).

Table 31: Exact recoveries out of five seeds per table.

Table	Δ	MNIST SlotCNN	MNIST ResNet	SVHN SlotCNN	SVHN ResNet
`flip_20@c100`	12	3/5	2/5	1/5	0/5
`flip_20@c101`	10	5/5	5/5	5/5	0/5
`flip_20@c102`	13	5/5	5/5	5/5	0/5
`flip_20@c103`	11	5/5	5/5	4/5	0/5
`flip_20@c104`	12	5/5	5/5	5/5	0/5
`flip_20@c105`	11	5/5	5/5	5/5	4/5
`flip_20@c106`	12	5/5	5/5	5/5	0/5
`flip_20@c107`	11	5/5	5/5	5/5	4/5
`flip_20@c108`	12	5/5	5/5	5/5	0/5
`flip_20@c109`	11	3/5	3/5	5/5	0/5

Pre-registered two-sided run-level Fisher tests (descriptive, since seeds share tables; see Appendix E.1 for table-level tests) of MNIST against SVHN give $p = 1$ for SlotCNN (odds ratio 1.28) and $p = 2.8 \times 10^{-14}$ for SmallResNet (odds ratio 47.2; Holm-adjusted 1.1×10^{-13}). Wilcoxon signed-rank tests on per-table fractions give $p = 1$ and $p = 0.002$. The prediction that SVHN recovers less often than MNIST for both encoders is therefore supported for SmallResNet only. We call a checkpoint collapsed when some digit's test hard accuracy in some slot is below 0.5 (threshold fixed in the pre-registration). Of 56 failing checkpoints, 41 are collapsed; none of the 144 successes is (Fisher $p = 2.0 \times 10^{-30}$). The collapsed slot and digit vary: slot-2 digit 4 in 14 runs, slot-2 digit 0 in six, and nine other slot–digit pairs in the rest. In 25 of the 41, the collapsed digit indexes a

most-corrupted row (slot 1) or column (slot 2) of the supplied table. The 15 non-collapsed failures (13 SmallResNet, 2 SlotCNN, all SVHN) have minimum per-digit accuracy 0.52–0.85, overlapping the successes' minimum of at least 0.678, so the 0.5 threshold separates most but not all failures. Failing tables keep 1–15 wrong cells (median 8.5).

D.2 Interventions on the slot-1 collapse

Read-out patch (pre-registered). Starting from the ten failing SVHN checkpoints of the original table (SlotCNN E12, SmallResNet E17), the patch takes 200 Adam steps (learning rate 10^{-3}) on the slot-1 read-out layer only, with all other parameters frozen and batch normalization in evaluation mode. It uses cross-entropy on the true slot-1 concept of training examples whose true slot-1 digit is 5 or 6 (or 2 or 3 for the dose-matched control). The unchanged repair objective then continues for ten epochs. A scratch variant applies the same patch to a fresh SVHN SlotCNN. The recorded parameter differences confirm that only `encoder1.fc2.*` (SlotCNN) or `encoder1.fc.*` (SmallResNet) change.

Table 32: Read-out patch. "After patch" is slot-1 training accuracy on digit 5 immediately after the patch; "final" is test accuracy of the selected checkpoint.

Start	Encoder	Patch	Exact	Digit 5 after patch	Digit 5 final	Wrong cells
failing ckpt	SlotCNN	none	0/5	–	0.00	6.6
failing ckpt	SlotCNN	5, 6	0/5	0.82	0.00	6.2
failing ckpt	SlotCNN	2, 3	0/5	0.00	0.00	6.6
failing ckpt	SmallResNet	none	0/5	–	0.00	6.4
failing ckpt	SmallResNet	5, 6	0/5	0.43	0.02	6.2
failing ckpt	SmallResNet	2, 3	0/5	0.00	0.00	6.6
scratch	SlotCNN	none	0/5	–	0.00	6.6
scratch	SlotCNN	5, 6	0/5	0.23	0.00	16.8
scratch	SlotCNN	2, 3	0/5	0.00	0.04	6.2

All 45 runs fail. The pre-registered comparison of the 5/6 patch against the 2/3 patch on continuation starts gives 0/10 against 0/10 ($p = 1$). The patch works on its target, but ten repair epochs return slot-1 digit-5 accuracy to 0.00–0.02, with 0.89–0.91 of true fives mapped to six. On a fresh model the 5/6 patch skews early slot-1 predictions and leaves 16.8 wrong cells.

Frozen patched encoder (exploratory, added after the patch result). The same patch is followed by freezing the entire slot-1 encoder, including batch-normalization statistics, for all ten repair epochs (20 runs).

Table 33: Patch, then freeze slot 1 during repair.

Encoder	Patched digits	Exact	Digit 5 final	Wrong cells	Hard
SmallResNet	2, 3	0/5	0.00	6.6	0.598
SmallResNet	5, 6	0/5	0.39	2.0	0.607
SlotCNN	2, 3	0/5	0.00	8.2	0.390
SlotCNN	5, 6	1/5	0.80	1.4	0.487

With digits 5 and 6 patched, final slot-1 digit-5 accuracy stays at 0.80 (SlotCNN) and 0.39 (SmallResNet), and the selected tables have 0–5 wrong cells. With 2 and 3 patched they have 6–9. Both arms start from the same ten checkpoints, and the 5/6 arm has fewer wrong cells in all ten pairs (paired Wilcoxon $p = 0.002$). One SlotCNN run recovers $K^\star$ exactly (1/10 against 0/10). The patch sees only fives and sixes and so distorts the read-out for other digits; once frozen, slot-1 accuracy on digits 3 and 8 averages 0.28 and 0.38, a plausible but untested source of the remaining errors. This arm alone therefore does not isolate digit 5: it also changes the other digits. The digit-specific control of Appendix E.3, whose two arms differ only in digit-5 supervision, addresses that.

D.3 PRE-REGISTERED MATCHED-MARGIN REPLICATION

Tables follow the construction of Appendix B.1 with exactly ten wrong cells, of which $s_1 = 8$ agree with a coherent alternative $T_{p,q}$, $s_2 \in \{0, 1, 2\}$ carry a third label inside its disagreement set D, and $s_3 = 2 - s_2$ lie outside D. The structured permutation catalogue reaches $|D| \in \{18, 20, 28, 32, \ldots\}$ but nothing between 21 and 27. Tables were certified in construction order and binned by certified margin; the first ten per bin were kept, and construction stopped once all five bins were full (299 candidates certified, all OPTIMAL). Because $\Delta \leq |D| - 2s_1 - s_2$, bins 0 and 2 come from $|D| = 18$, bin 4 from 20, and bins 8 and ≥ 9 from 28 (margins 9, 10, 11, 12 for 2, 4, 2, 2 tables).

Table 34: Pre-registered replication: 50 tables, three seeds each.

Certified Δ	Tables	Exact	Rate	95% CI
0	10	17/30	0.57	[0.39, 0.73]
2	10	15/30	0.50	[0.33, 0.67]
4	10	29/30	0.97	[0.83, 0.99]
8	10	25/30	0.83	[0.66, 0.93]
$\geq$9	10	22/30	0.73	[0.56, 0.86]

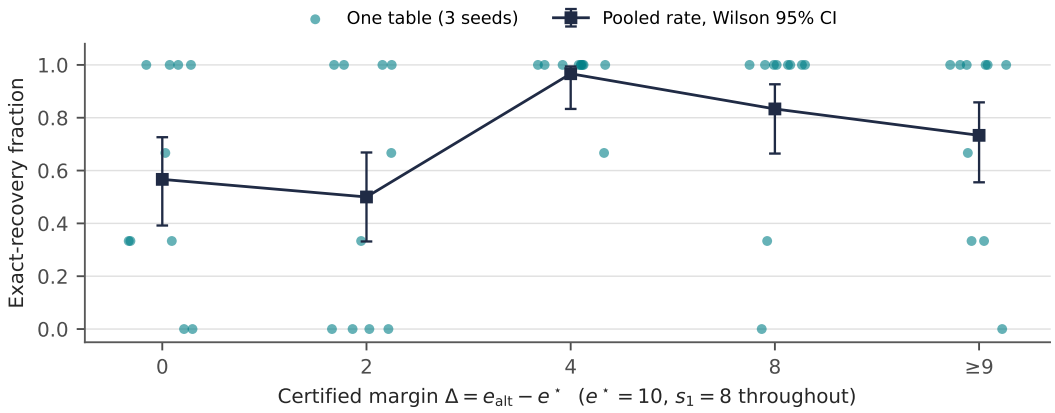


Figure 4: Pre-registered matched-margin replication. Circles are single tables; squares are pooled rates with Wilson intervals.

The pre-registered primary test, a one-sided table-level Spearman correlation, gives $\rho = 0.257$ with permutation $p = 0.038$ (20,000 permutations), Holm-adjusted $p = 0.11$. The secondary Cochran–Armitage trend test gives $z = 2.57$, $p = 0.010$, and a logistic likelihood-ratio test on bin gives $\chi^2(4) = 24.6$, $p = 5.9 \times 10^{-5}$, driven by the jump at $\Delta = 4$. Both secondary tests, like the pooled Wilson intervals, treat seeds as independent and are descriptive. The pattern is not monotone. Ties recover in 17 of 30 runs, although the exploratory study's five tie tables, built with the same recipe ($|D| = 18$, $s_2 = 2$, $s_3 = 0$), recovered in only 1 of 15; recovery varies strongly between tables that share every construction parameter. The alternative's distance cannot be held fixed: at $e^\star = 10$, $e_{\text{alt}} = 10 + \Delta$.

D.4 PERCEPTUAL DIFFICULTY

MNIST images receive fixed Gaussian pixel noise ($\sigma \in \{0.1, 0.3, 0.5\}$ in $[0, 1]$ pixel units, clipped, drawn once per image) before normalization. Each σ has a clean fixed-table run, which sets pre-repair concept accuracy, and `flip_5` and original `flip_20` repairs; five seeds each (45 runs). The $\sigma = 0$ and SVHN rows are the paper's runs.

Table 35: Perception against repair.

Condition	Clean hard concept	`flip_5` exact	`flip_20` exact	`flip_20` hard
MNIST σ=0 (paper)	0.975	5/5	4/5	0.956
MNIST σ=0.1	0.973	5/5	5/5	0.974
MNIST σ=0.3	0.962	5/5	5/5	0.963
MNIST σ=0.5	0.919	5/5	3/5	0.883
SVHN (paper)	0.710	5/5	0/5	0.637

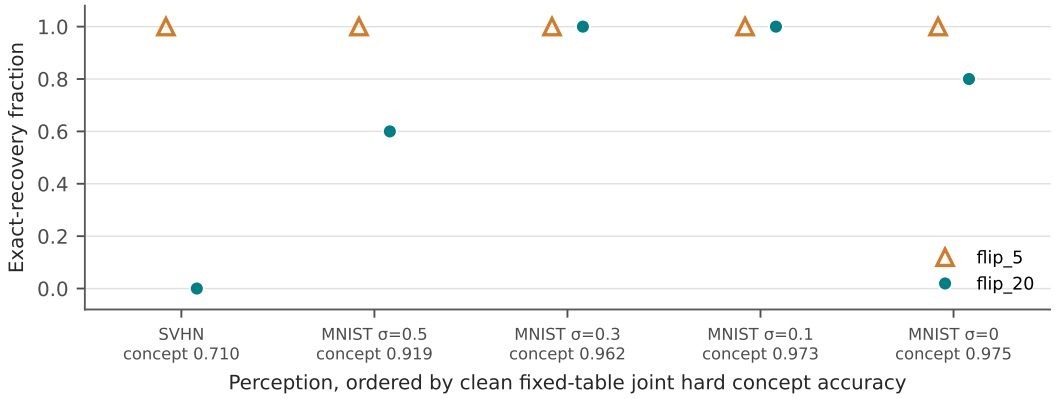


Figure 5: Exact recovery against pre-repair perception, ordered by clean fixed-table joint hard concept accuracy.

Noise up to $\sigma = 0.5$ lowers clean concept accuracy only to 0.919, and the twenty-edit repair falls to 3/5. This range does not approach SVHN's 0.710, so the sweep does not show that lower perception accuracy alone produces the SVHN failure. The multi-table result (SlotCNN recovers 45/50 on SVHN) shows that SVHN-level perception does not prevent repair either.

D.5 Optimization sensitivity

The original twenty-edit table is repaired with SlotCNN under Adam or SGD with momentum 0.9, encoder learning rates 10^{-4}, 10^{-3}, 10^{-2}, and weight decay 0 or 10^{-4} on encoder parameters only, with head learning rate 10^{-2}, on both domains, three seeds each (72 runs).

Table 36: Exact recoveries out of three seeds; hard is mean joint hard concept accuracy.

Optimizer	lr	wd	MNIST exact	MNIST hard	SVHN exact	SVHN hard
Adam	0.0001	0	0/3	0.797	0/3	0.183
Adam	0.0001	0.0001	0/3	0.839	0/3	0.189
Adam	0.001	0	2/3	0.941	0/3	0.634
Adam	0.001	0.0001	2/3	0.941	0/3	0.636
Adam	0.01	0	2/3	0.613	0/3	0.010
Adam	0.01	0.0001	3/3	0.923	0/3	0.010
SGD	0.0001	0	0/3	0.011	0/3	0.010
SGD	0.0001	0.0001	0/3	0.011	0/3	0.010
SGD	0.001	0	0/3	0.708	0/3	0.010
SGD	0.001	0.0001	0/3	0.708	0/3	0.010
SGD	0.01	0	0/3	0.922	0/3	0.304
SGD	0.01	0.0001	0/3	0.921	0/3	0.304

No setting recovers the SVHN table (0/36). On MNIST only Adam at learning rate 10^{-3} or 10^{-2} recovers it (9/12); SGD at learning rate at most 10^{-3} barely trains in ten epochs. Because no SVHN

run and no SGD run succeeds, a logistic model is completely separated, so we report counts and exact tests only: on MNIST, Adam recovers 9/18 and SGD 0/18 (Fisher $p = 0.001$); Adam recovers 0/6, 4/6, and 5/6 at learning rates 10^{-4}, 10^{-3}, and 10^{-2}, and 4/9 and 5/9 without and with weight decay. The paper's setting is therefore not an unusually unfavourable one for SVHN on this table.

D.6 JOINT CONCEPT ENCODER

The joint encoder keeps one SlotCNN trunk per slot but reads both feature vectors into one softmax over all 100 concept pairs, so slot concepts may be correlated. This violates the product structure used by Proposition 1, so no bound is claimed. Concept metrics use the argmax of each slot marginal (30 runs).

Table 37: Joint encoder against the paper's product encoders (five seeds).

Domain	Condition	Encoder	Label	Hard	Exact
MNIST	`clean`	product (paper)	0.975	0.975	–
MNIST	`clean`	joint	0.956	0.178	–
MNIST	`perm_both`	product (paper)	0.975	0.358	–
MNIST	`perm_both`	joint	0.958	0.090	–
MNIST	`flip_20`	product (paper)	0.956	0.956	4/5
MNIST	`flip_20`	joint	0.952	0.141	0/5
SVHN	`clean`	product (paper)	0.712	0.710	–
SVHN	`clean`	joint	0.597	0.121	–
SVHN	`perm_both`	product (paper)	0.711	0.276	–
SVHN	`perm_both`	joint	0.583	0.057	–
SVHN	`flip_20`	product (paper)	0.644	0.637	0/5
SVHN	`flip_20`	joint	0.590	0.088	0/5

The joint encoder fits labels almost as well as the product encoders on MNIST (0.956 clean), but it does not recover the intended concepts even with correct knowledge (joint hard accuracy 0.178), and it repairs no twenty-edit table. With a joint read-out, every pair with the right label is a zero-loss choice, so correct knowledge no longer identifies the slot concepts. The slotwise locality assumed in Section 3 is thus what makes concepts identifiable in the first place, not only a convenience for the analysis.

D.7 ABL-STYLE BASELINES

The ABL-style learner follows the abductive-learning loop (Dai et al., 2019) in its simplest form; it is not a reproduction of safe abductive learning (Yang et al., 2024) or of any published knowledge-revision method. For each training example it abduces the most probable concept pair under the product of the two slot distributions among pairs whose current table entry equals the task label, falling back to the unconstrained argmax when none does, and trains both encoders by cross-entropy on the abduced pair. The fixed-K variant keeps the supplied table. The revised variant, after each epoch, sets every table cell to the majority task label among training examples whose predicted concept pair is that cell, when at least 20 examples fall there and the majority share is at least 0.6. Both variants use ten epochs, Adam at 10^{-3}, and validation-NLL model selection.

From scratch (36 runs on `flip_5`, `flip_10`, and `flip_20`, three seeds each), no run recovers any table; joint hard concept accuracy stays at 0.18–0.30 on MNIST and 0.09–0.16 on SVHN. The same cold start fails with the correct table too (joint hard accuracy 0.257 on MNIST, 0.139 on SVHN), so these runs measure the cold-start problem of abduction, not wrong knowledge. We therefore repeated the comparison from the concept-pretrained encoders of the oracle control (Appendix C.9), the setting in which abductive learning is usually applied; this exploratory extension was designed after observing the cold-start result.

Table 38: ABL-style learners and the learned table from concept-pretrained encoders (three seeds; the learned-table `flip_20` rows are the oracle control, five seeds), with the cold-start correct-table check. Exact recovery is undefined for the correct table.

Domain	Table	Start	Method	Exact	Wrong	Hard
MNIST	`clean`	scratch	ABL-style, fixed K	–	0.0	0.257
MNIST	`clean`	scratch	ABL-style, revised	–	0.0	0.257
MNIST	`clean`	pretrained	ABL-style, fixed K	–	0.0	0.978
MNIST	`clean`	pretrained	ABL-style, revised	–	0.0	0.978
MNIST	`flip_5`	pretrained	ABL-style, fixed K	0/3	5.0	0.970
MNIST	`flip_5`	pretrained	ABL-style, revised	3/3	0.0	0.976
MNIST	`flip_5`	pretrained	learned table	3/3	0.0	0.977
MNIST	`flip_10`	pretrained	ABL-style, fixed K	0/3	10.0	0.967
MNIST	`flip_10`	pretrained	ABL-style, revised	3/3	0.0	0.978
MNIST	`flip_10`	pretrained	learned table	3/3	0.0	0.978
MNIST	`flip_20`	pretrained	ABL-style, fixed K	0/3	20.0	0.935
MNIST	`flip_20`	pretrained	ABL-style, revised	3/3	0.0	0.979
MNIST	`flip_20`	pretrained	learned table	5/5	0.0	0.977
SVHN	`clean`	scratch	ABL-style, fixed K	–	0.0	0.139
SVHN	`clean`	scratch	ABL-style, revised	–	0.0	0.139
SVHN	`clean`	pretrained	ABL-style, fixed K	–	0.0	0.711
SVHN	`clean`	pretrained	ABL-style, revised	–	0.0	0.711
SVHN	`flip_5`	pretrained	ABL-style, fixed K	0/3	5.0	0.692
SVHN	`flip_5`	pretrained	ABL-style, revised	3/3	0.0	0.715
SVHN	`flip_5`	pretrained	learned table	0/3	4.7	0.711
SVHN	`flip_10`	pretrained	ABL-style, fixed K	0/3	10.0	0.669
SVHN	`flip_10`	pretrained	ABL-style, revised	3/3	0.0	0.721
SVHN	`flip_10`	pretrained	learned table	2/3	2.7	0.722
SVHN	`flip_20`	pretrained	ABL-style, fixed K	0/3	20.0	0.604
SVHN	`flip_20`	pretrained	ABL-style, revised	3/3	0.0	0.718
SVHN	`flip_20`	pretrained	learned table	5/5	0.0	0.720

With pretrained perception, majority revision recovers the supplied table in all 18 runs (three corruption tables, two domains, three seeds each), and keeps concept accuracy at the pretrained level. The fixed-K variant keeps every wrong cell and loses concept accuracy as corruption grows (SVHN 0.692, 0.669, 0.604), because abduction forces concepts to fit the wrong entries. From the same start, the learned table recovers 11/11 on MNIST but 7/11 on SVHN (`flip_5` 0/3, `flip_10` 2/3, `flip_20` 5/5). The discrete revision is the stronger repair mechanism once concepts are separated, which is consistent with the paper's use of the learned table as an experimental instrument rather than a competitive method. Cold-started ABL recovers nothing, even with correct knowledge, whereas the learned table does recover from scratch in many conditions (Section 7.4); given separated concepts, both mechanisms recover most tables.

E Replication studies and baseline comparisons

This section studies repair on previously unused tables, matched-composition replication, digit-specific interventions, perception on repair runs, and Safe-ABL. The tables were certified and their primary tests frozen before training (`results/revision2/prereg.json`, `prereg_r3.json`); results are generated by `scripts/rev2_report.py` and checked by `scripts/rev2_validate.py`. The confirmatory family {never-used-table SlotCNN, never-used-table SmallResNet, digit-5 isolation, matched margin} is Holm-corrected.

E.1 Ten never-used twenty-edit tables

Four of the ten tables of Appendix D.1 (`flip_20@c100–c103`) had been used, with three fixed-table runs each, by the frozen detector evaluation (E14). We therefore drew four more tables, `flip_20@c110–c113`, checked them against every earlier run record, and certified them before training ($e^\star = 20$, $\Delta = 11$ each). Together with `c104–c109` they form ten tables that no earlier experiment used; the main text reports this set.

Table 39: Exact recoveries out of five seeds on the ten never-used tables.

Table	MNIST SlotCNN	MNIST ResNet	SVHN SlotCNN	SVHN ResNet
`flip_20@c104`	5/5	5/5	5/5	0/5
`flip_20@c105`	5/5	5/5	5/5	4/5
`flip_20@c106`	5/5	5/5	5/5	0/5
`flip_20@c107`	5/5	5/5	5/5	4/5
`flip_20@c108`	5/5	5/5	5/5	0/5
`flip_20@c109`	3/5	3/5	5/5	0/5
`flip_20@c110`	3/5	5/5	1/5	0/5
`flip_20@c111`	5/5	5/5	5/5	0/5
`flip_20@c112`	5/5	5/5	5/5	1/5
`flip_20@c113`	5/5	5/5	5/5	1/5

Inference is at the table level, since seeds share a table: the pre-registered paired Wilcoxon test over the ten tables gives $p = 1$ for SlotCNN (SVHN lower on one table, higher on one) and $p = 0.002$ for SmallResNet (SVHN lower on all ten). A bootstrap over tables gives an MNIST$-$SVHN recovery-rate difference of 0.00 [-0.12, 0.12] for SlotCNN and 0.76 [0.56, 0.94] for SmallResNet. Of 50 failing checkpoints, 35 are collapsed and none of 150 successes is. Run-level Fisher tests and Wilson intervals in Appendix D.1 treat seeds as independent and are descriptive only.

E.2 Matched-composition margin replication

In the replication of Appendix D.3, s_1 and t were fixed but the alternative's disagreement size $|D|$ and the number s_2 of third-label edits changed with the margin bin. Holding e_{alt} fixed is impossible at fixed $e^\star$, but $|D|$ need not vary in principle. Before any training we certified 55 tables at five fixed compositions: at $|D| \le 20$ the certified margin equalled its construction bound $|D| - 2s_1 - s_2$ in 46 of 48 tables, so a fixed composition there essentially fixed the margin. Among the compositions explored, the margin varied at a fixed composition only when the alternative was far, at $|D| = 28$. We therefore fixed $|D| = 28$, $s_1 = 8$, $s_2 = 0$, and $s_3 = 2$ for every table, so tables differ only in which cells are corrupted and in the alternative's permutation pair. Certifying 101 candidates (all optimal) filled five margin bins with ten tables each.

Table 40: Matched-composition replication: 50 tables, three seeds each. The ≥ 12 bin contains only $\Delta = 12$, the construction maximum. Wilson intervals pool seeds and are descriptive.

Certified Δ	Tables	Exact	Rate	95% CI
≤ 8	10	17/30	0.57	[0.39, 0.73]
9	10	29/30	0.97	[0.83, 0.99]
10	10	29/30	0.97	[0.83, 0.99]
11	10	23/30	0.77	[0.59, 0.88]
≥ 12	10	29/30	0.97	[0.83, 0.99]

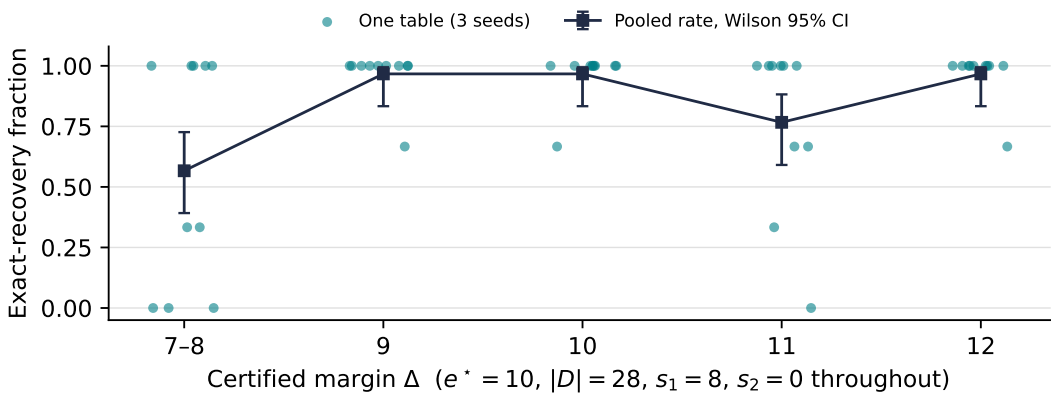


Figure 6: Matched-composition replication. Circles are single tables; squares are pooled rates with Wilson intervals.

The pre-registered primary one-sided table-level Spearman correlation is $\rho = 0.216$ (permutation $p = 0.064$; Holm-adjusted 0.19 within the confirmatory family specified for this section). The secondary Cochran–Armitage trend test, which like the pooled Wilson intervals treats seeds as independent and is therefore descriptive, gives $z = 2.88$, $p = 0.004$, and the lowest bin differs from the highest (table-level Mann–Whitney $p = 0.04$), but the rates are not monotone (0.57, 0.97, 0.97, 0.77, 0.97). The design restricts margins to 7–12 ($\Delta \leq |D| - 2s_1 - s_2 = 12$) and so cannot address low margins, where the unmatched studies show the largest differences.

E.3 Digit-specific isolation of the collapse

The exploratory frozen-encoder intervention of Appendix D.2 patched only fives and sixes, which also damaged other digits. Here the slot-1 read-out of each of the same ten failing SVHN checkpoints is patched for 200 steps either on all ten digits or on all digits except 5, and the whole slot-1 encoder is then frozen through the unchanged ten-epoch repair. The two arms differ only in whether digit 5 is supervised.

Table 41: Digit-specific isolation, ten shared failing checkpoints (five per encoder). "Other digits" is the lowest final slot-1 accuracy over the nine digits other than 5.

Encoder	Patched digits	Exact	Wrong cells	Digit 5 final	Other digits (min)
SmallResNet	all ten	1/5	2.2	0.24	0.68
SmallResNet	all but 5	0/5	6.6	0.00	0.87
SlotCNN	all ten	4/5	0.2	0.69	0.64
SlotCNN	all but 5	0/5	6.6	0.00	0.78

Supervising digit 5 turns 0/10 exact recoveries into 5/10 (SlotCNN 4/5, SmallResNet 1/5), with 1.2 against 6.6 mean wrong cells. All five discordant pairs favour the all-digit arm, but the pre-registered paired sign test gives $p = 0.0625$, the smallest value attainable with five discordant pairs, so the primary test is not significant at 0.05. The all-digit arm has fewer wrong cells in all ten pairs (paired Wilcoxon $p = 0.002$). Its other digits are, if anything, recognized slightly worse than in the control, so their damage does not explain the difference. Recovery follows how far digit 5 is restored: SlotCNN reaches final digit-5 accuracy 0.69 and recovers 4/5, SmallResNet 0.24 and 1/5. On this table, including digit 5 in the patch therefore reduces residual errors and partly restores exact repair. The intervention retrains the whole slot-1 read-out and freezes the whole slot-1 encoder, and other digits change as well, so it supports a causal role of the digit-5 representation without isolating it; the primary exact-recovery test is not significant, the wrong-cell comparison is secondary, and one table was tested.

E.4 PERCEPTION MEASURED ON THE REPAIR RUNS

The perceptual sweep of Appendix D.4 used separately trained clean-table models as its measure of pre-repair perception and stopped at concept accuracy 0.919. We extended the noise to $\sigma = 1.5$ and ran the original twenty-edit repair with per-epoch traces, so that pre-repair perception is read from each repair run itself: its validation joint concept accuracy at the last epoch before the table first changes (five seeds per condition, plus clean fixed-table runs at the new noise levels and a traced SVHN reference).

Table 42: Repair against the run's own perception on the original twenty-edit table. Pre-edit concept is validation joint concept accuracy at the last epoch without a table edit; first edit is the mean epoch of the first edit.

Condition	Exact	Epoch-1 concept	Pre-edit concept	First edit
MNIST σ=0	4/5	0.885	0.885	2.0
MNIST σ=0.1	5/5	0.893	0.893	2.0
MNIST σ=0.3	5/5	0.750	0.750	2.0
MNIST σ=0.5	3/5	0.568	0.568	2.0
MNIST σ=0.7	5/5	0.333	0.333	2.0
MNIST σ=0.9	5/5	0.196	0.458	2.8
MNIST σ=1.2	5/5	0.044	0.269	3.6
MNIST σ=1.5	0/5	0.024	0.194	5.8
SVHN	0/5	0.051	0.196	3.0

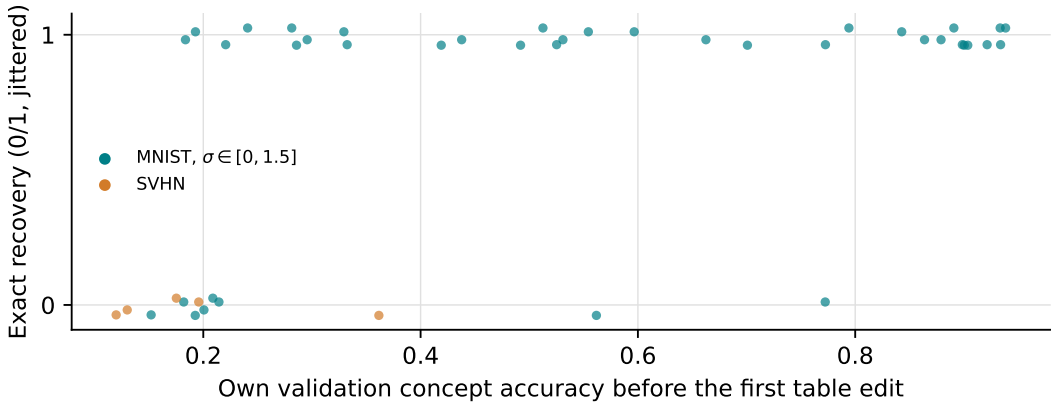


Figure 7: Exact recovery of each run against its own pre-edit concept accuracy.

Clean fixed-table concept accuracy at $\sigma = 0.7$, 0.9, 1.2, and 1.5 is 0.822, 0.676, 0.454, and 0.305, yet repair succeeds 5/5 down to $\sigma = 1.2$ and fails only at $\sigma = 1.5$; perception well below SVHN's does not by itself prevent repair. Noise delays the first edit from epoch 2 to 5.8. The run's own pre-edit accuracy is more informative on average: the two conditions with 0/5, $\sigma = 1.5$ and SVHN, start editing at mean accuracy 0.19–0.20, while every fully recovering condition averages 0.27 or more (run-level Spearman $\rho = 0.59$, exploratory). Individual runs overlap, however: successes start as low as 0.18, and failures occur up to 0.77 (the $\sigma = 0$ failure), so this accuracy is associated with recovery without determining it. The SVHN reference uses the original table, whose SlotCNN failure is specific to that table.

E.5 SAFE-ABL

Safe abductive learning (Yang et al., 2024) gives every uncertain rule r_i a weight $w_i \in [0, 1]$, initialized at 0.5, defines the abduction distribution $Q_w(z \mid y) \propto 1 - \prod_i \big(1 - \mathbf{1}\{r_i, z, y\} w_i\big)$ over concept assignments, and minimizes $-\log \sum_z P_\theta(z \mid x) Q_w(z \mid y)$ over the perception parameters and the weights, clamping w to $[0, 1]$ after every step; rules with $w \geq 0.5$ are kept. We take each

supplied table entry "pair (a, b) has label t" as a rule. In the Safe-ABL paper's tasks every rule is a total function of the concepts, so consistency of a rule with (z, y) means that the rule maps z to y. A single-entry rule, by contrast, is vacuously consistent with every other pair, and a literal reading would give all pairs nearly equal abduction mass. We therefore count a rule only when it derives y from z, that is, when $z = (a, b)$ and $t = y$; this coincides with the paper's definition for total rules. With no expert-trusted rules the distribution becomes $Q_w(z \mid y) = w[z, y] / \sum_{z'} w[z', y]$, an adaptation rather than a literal transcription. Weights may reach zero, so we fix the boundary case: if every weight in column y is zero, no rule can abduce y, the ratio is undefined, and we take $Q_w(\cdot \mid y)$ uniform over the n^2 concept pairs. Q_w is then a distribution for every label, and such an example contributes the constant $\log n^2$. The implementation normalizes whenever the column carries any positive mass, so Q_w is a distribution for every label. The reported runs use this convention. All 72 runs were also executed under the earlier variants of the head: every run keeps its exact-recovery outcome, while per-run joint concept accuracy differs by at most 0.0392 and label accuracy by at most 0.024. Those differences combine the change of code with re-execution, so we report them without attributing them to either. The supplied-rule variant contains only the supplied entries, so it can flag wrong entries but proposes no replacement. Our candidate extension adds a rule for every (cell, label) pair, with non-supplied weights starting at 0.05, and reads the repaired table as the highest-weight label per cell. The weights use Adam at learning rate 0.01. For label prediction, a cell whose rule weights are all zero predicts uniformly over labels, so every predictive distribution is normalized; an earlier version left such cells at zero, which affected validation-loss checkpoint selection, and all 72 runs were repeated with the corrected evaluation (the table reports the repeat; conclusions did not change). Following the paper's protocol with a pre-trained perception model, we start either from the concept-pretrained encoders of the oracle control or from scratch (three corruption tables, two domains, three seeds; 72 runs).

Table 43: Safe-ABL on `flip_5`, `flip_10`, and `flip_20`, pooled (nine runs per row). AUROC scores wrong entries by low weight; TPR/FPR use the paper's threshold 0.5. Exact recovery is not defined for the supplied-rule variant, which keeps the supplied table.

Variant	Start	Domain	Exact	Wrong	AUROC	TPR/FPR	Hard
supplied rules	pretrained	MNIST	–	–	0.95	0.93/0.28	0.959
supplied rules	pretrained	SVHN	–	–	0.97	0.96/0.35	0.653
supplied rules	scratch	MNIST	–	–	0.64	0.95/0.71	0.123
supplied rules	scratch	SVHN	–	–	0.56	0.91/0.80	0.040
candidates	pretrained	MNIST	9/9	0.0	1.00	1.00/0.18	0.975
candidates	pretrained	SVHN	9/9	0.0	1.00	1.00/0.00	0.713
candidates	scratch	MNIST	0/9	98.1	0.49	0.98/0.99	0.010
candidates	scratch	SVHN	0/9	98.0	0.51	1.00/0.99	0.010

From pretrained encoders, supplied-rule Safe-ABL ranks wrong entries well (AUROC 0.95 on MNIST, 0.97 on SVHN) and flags 93–96% of them, but also flags 28–35% of the correct entries. With candidate rules it recovers every table (18/18 runs). From scratch, detection is close to chance (AUROC 0.64 and 0.56), concepts collapse, and the candidate variant repairs nothing. Like the ABL-style learner, Safe-ABL solves the revision problem once concepts are separated and does not address reaching them.

F Detection and oracle auditing

F.1 Development calibration and contaminated evaluation

We report area under the receiver operating characteristic curve (AUROC) and precision–recall curve (AUPRC). The detection score is $D_{ks} = \widehat{R}^{\text{fixed}}_{B,ks} - \widehat{R}^{\text{free}}_{B,s}$ for table k and seed s. Its reference target is $\mathbf{1}\{\delta_{\text{det},k} > 0\}$, not whether K contains any wrong cell. Positive and negative calibration run scores define the midpoint threshold, which is also applied to the seed mean $\bar{D}_k$ within each table. Odd corruption seeds are assigned to calibration and even seeds to the designated holdout, with clean assigned to calibration. Ten random-permutation tables add zero-floor negatives with 88–95 wrong cells.

Table 44: Development detection with one seed-mean score per table. TPR/FPR are true- and false-positive rates at the fixed calibration-run threshold. The new-table subset removes the previously inspected original-sweep conditions.

Table set	Tables (pos/neg)	AUROC	Correct	TPR	FPR
Calibration	21 (15/6)	1.000	21/21	1.000	0.000
Designated holdout	25 (18/7)	1.000	24/25	0.944	0.000
New-table subset	16 (11/5)	1.000	16/16	1.000	0.000

The free baseline's mean training NLL is approximately 0.66 versus test NLL 1.10. This gap does not identify under-training as the cause. It also shifts the score substantially below zero; a naive threshold of zero fails despite separation after calibration. The one designated-holdout miss is the two-edit positive-floor table, whose floor is smaller than any calibration positive's. Nine of the 25 designated test tables had already appeared in the original sweep, and the statistic and threshold rule were chosen after those outcomes were inspected. These are development results, including the post-hoc new-table subset, not a clean held-out validation. The new-table subset contains only five negative tables.

F.2 Frozen evaluation on fresh corruption tables (E14)

The protocol file was committed as `534e090` before generation, certification, and training of the evaluation tables. It freezes the statistic, baseline, threshold -0.8679218403995038, preprocessing, ten-epoch budget, positive-floor target, generation scheme, and reported metrics. Three seeds per table give 144 records over 48 fresh tables: four tables at each positive floor $0.01, 0.02, 0.03, 0.05, 0.10, 0.20$, and 24 certified zero-floor permutation tables. Corruption seeds start at 100. No threshold was refitted.

Table 45: Frozen positive-floor detector on 48 fresh tables. Wilson 95% intervals use tables as the unit; image pools and free baselines are reused.

Quantity	Value
Tables	48 (24 pos / 24 neg)
AUROC	1.000
AUPRC	1.000
Accuracy at frozen threshold	0.854 [0.728, 0.928]
Sensitivity	0.708 [0.508, 0.851]
Specificity	1.000 [0.862, 1.000]
Confusion (TP/FN/FP/TN)	17 / 7 / 0 / 24
Smallest positive $\bar{D}$	-0.9525
Largest negative $\bar{D}$	-1.0072

All 24 zero-floor tables are negative at the fixed threshold. Positive counts are 0/4, 1/4, 4/4, 4/4, 4/4, and 4/4 in increasing floor order. All seven misses therefore occur at floors 0.01 or 0.02; calibration had no positive floor below 0.05. Spearman correlation with the floor is 0.934, and with the number of wrong cells is -0.646. AUROC 1.000 means the two target classes are perfectly ranked here, not that the score exactly orders positive floors. The free baseline does not drive this separation: the table-mean label accuracy of the fixed-table models alone also ranks the classes perfectly, with positives between 0.674 and 0.965 and negatives between 0.971 and 0.977 (`results/audit/revision_additions_checks.json`). This evaluation is confirmatory with respect to its frozen protocol and fresh table conditions. It reuses the previously inspected test-image pool and free baseline and supplies no independent confirmation on new image data.

F.3 Query policies and intervention semantics

These policies are a downstream diagnostic application of the certificates (Section 4), not a theoretical contribution of the paper. Oracle disambiguation and minimal constraint-set repair are already

analyzed by Takemura et al. (2026); BEARS also uses concept uncertainty to guide annotation (Marconato et al., 2024). Here the target is recall of wrong table cells through true-concept feedback. All policies select *examples*, not arbitrary cells in the true table. The oracle reveals the selected example's two true digit concepts, and its observed task label verifies $K^\star$ at that cell. Repeated queries can hit the same true cell. Random selection samples examples without replacement. The residual heuristic sorts by clipped example NLL while preferring distinct predicted cells. Entropy and confidence policies use label entropy and negative joint concept confidence respectively, with the same predicted-cell diversity rule. Those two scores do not consult task labels.

Coverage computes a normalized mean residual per predicted cell and selects one maximum-residual representative from each highest-weight cell. With fixed nonnegative weights, selecting the top cells maximizes that weight-coverage objective, but this supplies no corrupted-cell recall guarantee. Masked coverage processes batches of five examples and masks verified true cells between batches, without recomputing suspicion. Predicted cells can differ from the true cells revealed by the oracle, so this is not guaranteed adaptive coverage of true corruption. Both the original table and the repaired current table are retained to classify queries consistently.

For each budget prefix, queried true cells are repaired and a fresh fixed-table model is trained. Metrics are computed on the usual test pairs. There are 144 audit/retraining rows in total: six policies, two corruption conditions, four budgets, and three seeds. The entropy/confidence extension contributes 48 of those rows. Budget fifty on twenty edits gives masked-coverage label accuracy 0.9738 versus 0.7353 for the matching three unrepaired seeds; using the five-seed original mean for that comparison would change its unit.

Table 46: Corrupted-cell recall for all query budgets. Values are three-seed means; a query reveals the true concepts of one selected example.

Knowledge	Policy	$m = 5$	$m = 10$	$m = 20$	$m = 50$
flip_5	random	0.000	0.067	0.067	0.333
flip_5	heuristic	0.533	1.000	1.000	1.000
flip_5	entropy	0.333	0.600	0.800	0.933
flip_5	confidence	0.467	0.533	0.733	0.933
flip_5	coverage	1.000	1.000	1.000	1.000
flip_5	masked_coverage	1.000	1.000	1.000	1.000
flip_20	random	0.050	0.067	0.200	0.400
flip_20	heuristic	0.133	0.300	0.450	0.833
flip_20	entropy	0.167	0.267	0.367	0.617
flip_20	confidence	0.167	0.267	0.350	0.583
flip_20	coverage	0.250	0.500	0.867	0.933
flip_20	masked_coverage	0.250	0.500	0.900	1.000

Table 47: Budget fifty: recall and post-retraining label, hard, and soft concept metrics for each policy.

Knowledge	Policy	Recall	Label	Joint hard	Joint soft
flip_5	random	0.333	0.9412	0.9729	0.9088
flip_5	heuristic	1.000	0.9738	0.9738	0.9648
flip_5	entropy	0.933	0.9693	0.9725	0.9583
flip_5	confidence	0.933	0.9695	0.9727	0.9587
flip_5	coverage	1.000	0.9738	0.9738	0.9648
flip_5	masked_coverage	1.000	0.9738	0.9738	0.9648
flip_20	random	0.400	0.8369	0.9569	0.7208
flip_20	heuristic	0.833	0.9362	0.9720	0.8990
flip_20	entropy	0.617	0.8895	0.9682	0.8122
flip_20	confidence	0.583	0.8767	0.9619	0.7967
flip_20	coverage	0.933	0.9589	0.9729	0.9395
flip_20	masked_coverage	1.000	0.9738	0.9738	0.9648

G ARTIFACT PROVENANCE AND SCOPE OF VERIFICATION

The frozen repository contains 1,469 unique training records, including pretraining parents, pilot runs, and development records. Soft metrics exist for 1,150. The 319 without them comprise E14 (144), E15 (135), E18 repair runs and their init18 parents (20), concept pretraining parents in init6 (10), one development record, and nine free baselines. Only the free baselines lack a defined per-slot concept metric; the other 310 are measurable but were not measured. Neither the E14 loss score, the E15 exact-table outcome, nor the E18 comparison requires soft metrics; the per-digit hard accuracies used for the collapse analysis were computed separately (Appendix C.9).

On the measured subset, saved hard metrics match raw records and all saved pointwise and aggregate confidence checks pass. Of these records, 552 use the fixed head and 598 use learned heads or concept-pretraining objectives. The latter are not tests of the fixed-K risk theorem. The integrity pass loaded all 1,469 checkpoints with restricted weight loading, checked finite tensors, validated loss-array shapes, ranges and means against CSVs, and hashed artifacts. It performed no fresh inference or training. Separate checks recomputed final summary arithmetic from stored records and argmax tables, and inspected the per-epoch traces.

The evidence builder joins records by configuration-derived run identifiers, checks uniqueness and repair matrices, verifies the stored construction arrays, recomputes means and table-level intervals, and records hashes of inputs. This audit does not claim fresh inference of every checkpoint or fresh solution of every certificate. The inference measurements and solver records are saved experimental artifacts, with explicit independent spot checks.

Table 48: Experiment-to-claim map. Identifiers refer to source directories or run groups. Extended artifacts include configurations and per-example losses.

Group	Main question	Evidence location under `results/`
E1/E2	Compensation and initialization	root CSV; `predictions_v2.json`
E3/E3b	Head and repair behavior	root and `e3b/` CSVs
E4/E4b	Query targeting and repair	`e4_repair.csv`; `repaired/`
E5	Size and label-noise robustness	root CSV; soft metrics
E6/E6i/E6L	Support and optimization	`diagnostics/e6/`; related CSVs
E7/E7i	Comparison and initialization	`e7/`, `e7i/`; certificates
E8	Fixed-edit boundary	`e8/constructions.csv`; table arrays
E9	Variation across tables	`e9/`; `diagnostics/e9_per_table.csv`
E10	Positive-floor detection	`diagnostics/e10_tables.csv`
E11	Initialization and penalty controls	`e11/`; `diagnostics/e11_runs.csv`
E12	Image-domain comparison	`e12/`; `diagnostics/e12_summary.csv`
E13	Training budget	`e13/traces/`; budget diagnostics
E14	Frozen floor detection	`audit/detector_freeze.json`; `e14/`
E15	Matched composition	`e15/constructions.csv`; table arrays
E16	Ten-seed penalty ablation	`diagnostics/e16_runs.csv`
E17	Second encoder	`e17/`; soft metrics
Collapse	Per-digit concepts	`diagnostics/flip20_digit_collapse*`
E18	Oracle-pretrained repair	`audit/e18_freeze.json`; `e18/`

Pilot and development records, superseded head settings, concept-pretraining parents, and longer-budget controls remain in the inventory rather than being silently counted as independent confirmations. The main certificates compare known benchmark programs; restricted-support certificates include post-hoc analyses. File timestamps and source comments do not by themselves establish preregistration. All table and figure generators used for this manuscript operate on saved records and do not launch training.