GENERALISED RULES FROM COMPOSITIONAL STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning reusable components from complete-task answers is a central challenge in compositional generalisation. Understanding failures requires distinguishing what training examples reveal from whether learners can use that information. We study this distinction in finite-group compositions with independently hidden input and output meanings. By varying selected input around an anchor, we construct complete-task examples that provably determine the entire rule and enable an exact reconstruction algorithm. These examples also improve neural generalisation over equally sized random datasets with matching answer frequencies. Even when the target is expressible, comparisons of neural, discrete and constructive learning on identical sufficient examples show substantial differences. To investigate the remaining learning difficulty, we derive a cancellation law showing how joint uncertainty in component meanings suppresses input-dependent predictions. Together with matched interventions, these findings suggest that compositional generalisation depends on both the structure revealed by training examples and the learner's ability to coordinate component meanings.

1 INTRODUCTION

Compositional generalisation means solving new problems by recombining known components, but when training supplies only a complete task's answer, component meanings must be inferred jointly. Because equally sized training sets can reveal different relationships between those meanings, we ask: *how can relationships between complete examples expose reusable structure, and how effectively do learners use it?*

A model may have parameter settings that implement the correct rule, while training fails to find them. This gap has been studied in parity learning and permutation composition (Shalev-Shwartz et al., 2017; Barak et al., 2022; Wang et al., 2025). Providing intermediate answers can guide learning by revealing how parts of a task are solved. We ask whether carefully chosen complete-task examples can also provide useful guidance through the relationships between their inputs and final answers. We therefore examine which examples contain sufficient information to determine the rule, and whether learners use that information to predict unseen combinations.

We formulate a framework for learning hidden dictionaries over finite group compositions, in which each input position and the output have independently relabelled symbol meanings. To examine how complete-task answers support learning these hidden meanings, we compare three learning systems. A soft neural composer receives the correct composition table and learns probabilistic dictionaries through gradient training, while a discrete learner uses the same table and searches directly over one-to-one dictionaries. A constructive recovery learner infers an executable rule from selected complete-task answers without receiving the latent table.

The constructive recovery learner selects complete inputs that differ from a fixed reference input in at most two positions. Varying one position at a time maps each symbol at that position to its resulting answer. These maps are bijections, allowing all positions to be represented in the same answer alphabet. Queries that vary two positions reveal how these answer-based representations combine; group structure allows the combination rule to be completed from selected queries. We prove that mapping each input through its position-specific bijection and applying the recovered rule in input order reproduces the target function on every combination, using only complete-task answers.

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.

Empirically, constructive recovery succeeds exactly on all 162 associative task instances, including 82 with order-sensitive compositions.

We then give the same examples to the neural and discrete learners, enabling controlled comparisons when the examples determine the target function and both models can express it. In the 36-task study, the discrete learner achieves approximately 90% accuracy on unseen combinations, compared with 28% for the neural composer. Additionally, the structured examples improve neural generalisation by approximately 16 percentage points over equally sized random sets with matching answer frequencies. These results establish a replicated benefit from training example structure and substantial differences in how learners exploit sufficient information under their respective fixed budgets.

We investigate these differences through theoretical analysis and controlled interventions. Group operations have the *Latin property*: each element occurs once in every row and column of their tables. A uniformly uncertain input therefore makes the composed latent result uniform. Near uniform input and output dictionaries, we prove that input-dependent prediction differences require deviations in all dictionaries, giving a first possible term of degree $p + 1$ for p input positions. This cancellation law motivates removing uncertainty in individual dictionaries. On a paired 16-task subset, revealing one input dictionary raises accuracy from 30% to 64%, while revealing the output dictionary raises it to 82%. Their different benefits despite identical local orders show that dictionary identity matters beyond this order. Enforcing dictionary rows and columns to sum to one, to a verified numerical tolerance, yields little average improvement.

Our main contributions are threefold:

- **Exact reconstruction from complete-task answers.** We derive an ordered factorisation in answer-based representations that enables exact recovery of independently relabelled finite-group compositions. We give query upper and lower bounds and prove that varying two positions around a reference suffices, whereas varying one is insufficient in the worst case.
- **Understanding learning failures and improving generalisation.** We compare three learning systems on identical, provably sufficient examples with established neural expressibility. Reference-centred supervision improves neural generalisation beyond dataset size and answer balance, while substantial differences remain between learning systems.
- **A cancellation law for uncertain representations.** We establish the exact local prediction order for uncertain dictionaries under finite Latin compositions. Matched interventions show that resolving different dictionaries produces different learning benefits despite identical predicted orders.

2 Related work

Compositional learning and supervision. Systematic generalisation studies document failures on unseen combinations (Lake & Baroni, 2018; Dziri et al., 2023), while parity and permutation-composition theory distinguish expressibility from learnability (Shalev-Shwartz et al., 2017; Daniely & Malach, 2020; Barak et al., 2022; Wang et al., 2025). Auxiliary objectives, intermediate answers and easier compositions can support learning (Lee et al., 2015; Lightman et al., 2024; Wang et al., 2025). Instead of providing shorter-task or intermediate labels, our main supervision strategy selects related full-length examples whose final answers determine the rule, then tests their benefit beyond dataset size and answer balance.

Algebraic representations and reconstruction. Group-operation tasks reveal learned representations and generalisation dynamics (Chughtai et al., 2023; Nanda et al., 2023; Stander et al., 2024; Wu et al., 2025); Huh (2025) learns group operations and unitary representations from partial tables. Classical defining-set theory reconstructs group-derived binary tables (Bedford et al., 2003), while local membership-query learning restricts queries to neighbourhoods of sampled examples (Awasthi et al., 2013). We extend binary reconstruction to independently relabelled full-length compositions through an ordered factorisation. For the neural and discrete learners, we supply the operation table rather than learn it, isolating dictionary learning on the same sufficient examples.

Hidden meanings and learning dynamics. Neuro-symbolic systems combine learned representations with supplied reasoning structure (Manhaeve et al., 2018; Badreddine et al., 2022). Reasoning-shortcut studies examine whether those representations match intended concepts (Marconato et al., 2023; Bortolotti et al., 2024); our recovery objective instead concerns the observable function, allowing equivalent dictionaries. Quasigroup theory studies convergence towards uniformity (Yashunsky, 2024), whereas we analyse local prediction changes around uniform dictionaries, including uncertainty in the decoder. Building on differentiable assignment methods (Mena et al., 2018; Eisenberger et al., 2022), we compare enforcing continuous dictionary constraints with revealing correct meanings to distinguish their effects on learning.

3 FRAMEWORK FOR LEARNING HIDDEN COMPONENT MEANINGS

We study compositional generalisation from complete-task answers when the meanings of individual input symbols are hidden. Given observed input–answer pairs, the learning goal is to predict the correct answer for unseen combinations. Our framework has three main elements: a composition system that specifies how latent elements combine, unknown dictionaries that assign meanings to visible symbols, and learning systems that use observed answers to predict new combinations. Studying these systems within the same setting allows us to diagnose how they use the available information and to test ways of improving generalisation.

3.1 FINITE COMPOSITION SYSTEMS

Let G be a finite set of $q \geq 2$ elements and let $\oplus : G \times G \to G$ be a binary operation. $\oplus$ denotes the underlying composition operation, which is not necessarily commutative. Its *operation table* records $r \oplus s$ for every pair $r, s \in G$.

Combining a sequence. A complete task contains $p \geq 2$ input positions; we call p its *arity*. To combine several elements using the binary operation, let $m_{G,k} : G^k \to G$ denote the composition of an ordered sequence of k elements. Starting from $m_{G,1}(r_1) = r_1$, we define

$$m_{G,k}(r_1, \ldots, r_k) = m_{G,k-1}(r_1, \ldots, r_{k-1}) \oplus r_k, \qquad k \geq 2. \tag{1}$$

Latin operations and groups. A *Latin operation* has every element of G exactly once in each row and column of its operation table. Equivalently, for each fixed $r \in G$, both maps $s \mapsto r \oplus s$ and $s \mapsto s \oplus r$ are bijections. Figure 1a shows the operation table for addition modulo three.

Our principal tasks use finite *groups*. Their operation is associative, $(r \oplus s) \oplus t = r \oplus (s \oplus t)$, has an identity e satisfying $e \oplus r = r \oplus e = r$, and gives each element an inverse r^{-1} satisfying $r \oplus r^{-1} = r^{-1} \oplus r = e$. Every group operation satisfies the Latin property. Group structure enables our exact-recovery construction, whereas the local analysis of dictionary uncertainty requires only the Latin property. We include both *noncommutative* groups and *commutative* groups in our setting.

3.2 HIDDEN-SYMBOL TASKS AND THEIR MATHEMATICAL FAMILIES

Let Q and Y be the visible input and answer alphabets. Both contain q symbols, but may contain different symbols. Each input position, or *role*, $i \in \{1, \ldots, p\}$ has an unknown bijection $g_i : Q \to G$, and an unknown bijection $\phi : G \to Y$ maps the composed result to an answer. We call these bijections *dictionaries*. The target function $F : Q^p \to Y$ is

$$F(x_1, \ldots, x_p) = \phi(m_{G,p}(g_1(x_1), \ldots, g_p(x_p))) \,. \tag{2}$$

Thus, the task maps visible inputs to latent elements, composes those elements, and maps the result to an answer symbol using ϕ. For each task, the input and output dictionaries are sampled independently and held fixed across training and evaluation.

Task families. All families follow equation 2 with different composition systems. We use modular compositions over $\mathbb{Z}_7$ with $(q, p) = (7, 5)$ and $\mathbb{Z}_3^2$ with $(q, p) = (9, 4)$. To examine order-sensitive composition, we use the square-symmetry group D_4 and quaternion group Q_8, both with $(q, p) = (8, 4)$, together with cube rotations with $(q, p) = (24, 3)$ and planar translations and rotations with

$q = 36$ and $p \in \{3, 5, 7\}$. As a boundary control, the Fano quasigroup ($q = 7$) satisfies the Latin property without associativity. Its compositions follow the left-associated order in equation 1. Full details of operations and their implementation conventions are defined in Appendix B.

3.3 Training set, sufficiency and generalisation

A training set $\mathcal{S} = \{(x^{(t)}, y^{(t)})\}_{t=1}^{n}$ contains complete input tuples $x^{(t)} \in Q^p$ and their final answers $y^{(t)} = F(x^{(t)})$. The learning objective is to predict $F(x)$ for unseen combinations under the same hidden dictionaries and arity.

Determining the rule from training examples. We assess whether the training examples determine the rule within the possible functions allowed by our task assumptions. For a fixed group G and arity p, let $\mathcal{H}_{G,p}$ contain all functions in equation 2 obtained by varying the input and output dictionaries over bijections. When the group operation is unknown, we consider $\mathcal{H}_{q,p} = \bigcup_{G:\,|G|=q} \mathcal{H}_{G,p}$, which allows any group of size q.

Training set sufficiency A training set $\mathcal{S}$ is *sufficient* within a specified class if F is the only function in that class consistent with every training example. *Sufficiency* concerns the input-answer function, so different dictionaries may still represent that same function.

Exact recovery We evaluate trained models by classification accuracy on held-out input tuples. We use *exact recovery* for the stronger guarantee that a constructed predictor equals F on every input in Q^p. These criteria distinguish performance on the evaluated test set from a guarantee covering the entire input domain.

3.4 Learning systems

We construct three systems for learning F from complete-task answers. The *soft neural composer* learns probability distributions over symbol meanings through gradient training, while the *discrete learner* searches directly over one-to-one dictionaries. Both receive the Latin operation table, allowing us to study dictionary learning when the composition mechanism is already available. The *constructive recovery algorithm* selects related complete examples and uses their answers to infer an executable rule. It receives the group assumption and dimensions p, q, and constructs its own operation in the answer alphabet.

3.4.1 Soft neural composer

The soft composer represents the unknown dictionaries by trainable probability matrices and keeps the supplied operation table fixed.

For input role i, let $P_i \in \mathbb{R}^{q \times q}$, where $P_i[x, r]$ is the probability assigned to latent meaning $r \in G$ for visible symbol $x \in Q$. Let $D \in \mathbb{R}^{q \times q}$ represent the output dictionary, where $D[r, y]$ is the probability assigned to answer $y \in Y$ for latent result r. Every row of P_i and D sums to one. We also call D the *decoder*.

The operation table is encoded by the fixed tensor $T_{rsc} = \mathbf{1}[r \oplus s = c]$. If two latent elements are sampled independently from distributions v and w over G, their composition has distribution

$$\mathcal{B}(v, w)_c = \sum_{r,s \in G} T_{rsc} v_r w_s. \tag{3}$$

Let $\mathcal{B}_p$ apply this operation repeatedly in the same left-associated order as $m_{G,p}$. The model's answer distribution is

$$f_\theta(x) = D^\top \mathcal{B}_p\big(P_1^\top e_{x_1}, \ldots, P_p^\top e_{x_p}\big), \tag{4}$$

where e_x is the one-hot vector selecting the dictionary row for symbol x. The predicted answer is $\widehat{F}_\theta(x) = \arg\max_{y \in Y} f_\theta(x)_y$.

We parameterise the dictionaries by row-wise softmaxes, $P_i = \mathrm{softmax}_{\mathrm{row}}(W_i/\tau)$ and $D = \mathrm{softmax}_{\mathrm{row}}(W_o/\tau)$, with trainable logits $\theta = (W_1, \ldots, W_p, W_o)$ and temperature $\tau > 0$. Training updates the logits while keeping T fixed.

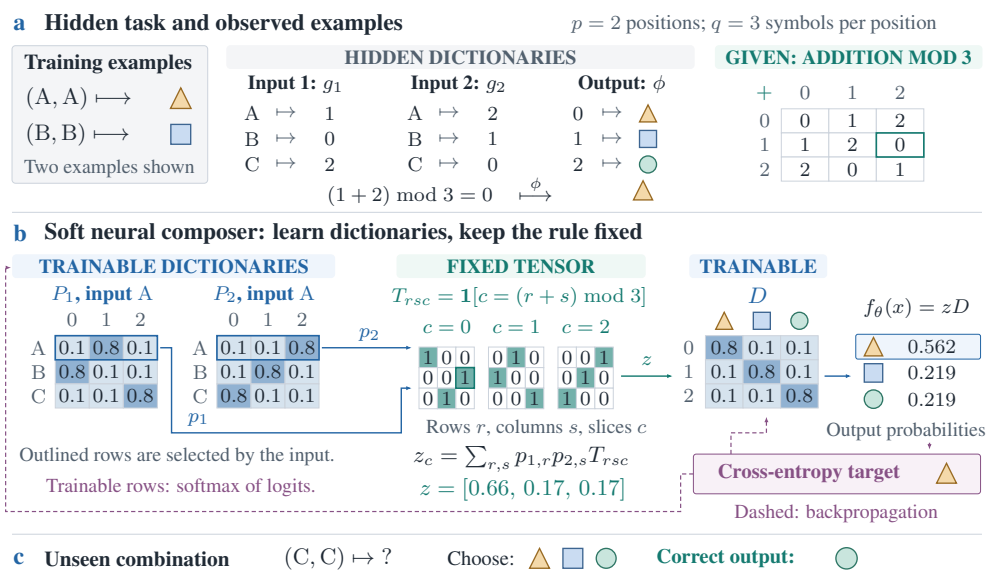

Figure 1: **Learning hidden component meanings from complete-task answers.** The example uses addition modulo three, $r \oplus s = (r + s) \bmod 3$, with two input positions and three symbols per alphabet. (a) Hidden position-specific dictionaries translate visible symbols into latent elements; an output dictionary names the composition result. (b) The soft composer receives the operation table as a fixed tensor and learns probability matrices for the input and output dictionaries. (c) It predicts an answer for an unseen input combination.

The model can express the target function: setting $P_i[x, r] = \mathbf{1}[r = g_i(x)]$ and $D[r, y] = \mathbf{1}[y = \phi(r)]$ reproduces F exactly. These permutation matrices are reached in the deterministic limit of the softmax parameterisation; sufficiently concentrated finite logits already give the correct classification on every input tuple. Because its probabilistic dictionaries also permit predictions outside $\mathcal{H}_{G,p}$, a soft predictor can fit a training set that is *sufficient* within this class yet fail to generalise. Figure 1 illustrates the task and the model.

3.4.2 Discrete learner

The discrete learner searches directly over one-to-one dictionaries, allowing us to compare discrete inference with gradient learning of probability matrices. It receives the true operation table and learns bijections $\gamma_i : Q \to G$ and $\rho : G \to Y$. Its predictor $\widehat{F}_{\gamma,\rho}$ follows equation 2, with g_i replaced by γ_i and ϕ by ρ. The objective $J_{\mathcal{S}}(\gamma, \rho) = \sum_{t=1}^{n} \mathbf{1}[\widehat{F}_{\gamma,\rho}(x^{(t)}) = y^{(t)}]$ counts correctly answered training examples.

With the other dictionaries fixed, maximising this objective over one dictionary is a linear assignment problem, solved exactly by the Hungarian algorithm. The learner alternates input-dictionary and decoder updates from several random initialisations, retaining the candidate with the highest training score. Although each update preserves the bijective structure and every resulting predictor belongs to $\mathcal{H}_{G,p}$, the alternating search may fall short of a joint optimum.

3.4.3 Constructive recovery learner

The constructive recovery learner uses the group assumption, the dimensions p, q and access to complete-task answers, with the latent operation table and dictionaries kept hidden. It can select an input tuple $x \in Q^p$ and request $F(x)$; we call this request a *query*.

The learner constructs bijections $s_i : Q \to Y$ that represent each role's input symbols using answer symbols, together with an operation $\odot : Y \times Y \to Y$ that combines these representations. Its predictor is $\widehat{F}_{\mathrm{rec}}(x) = s_1(x_1) \odot \cdots \odot s_p(x_p)$. We denote the collected input-answer pairs by $\mathcal{S}_{\mathrm{ref}}$. Section 5.1 describes how these examples are selected and proves that $\mathcal{S}_{\mathrm{ref}}$ is *sufficient* within $\mathcal{H}_{q,p}$.

4 EXPERIMENTAL SETUP AND RESEARCH QUESTIONS

Using the tasks and learning systems in Section 3, we examine whether complete-task examples determine the rule, whether their structure helps learning, and what affects the learner's ability to generalise.

Research questions. The study addresses five connected questions:

RQ1: How can we construct a training set that guarantees *sufficiency*, as defined in Section 3.3?

RQ2: Does the structure of these training examples improve neural generalisation beyond training-set size and answer balance?

RQ3: How effectively do different learners exploit the same sufficient training set?

RQ4: How does the combined uncertainty across the dictionaries affect the neural composer's local input-dependent predictions?

RQ5: What interventions can improve the learning of compatible component meanings?

RQ1 establishes sufficiency, providing the basis for the learning comparisons. RQ2 examines the benefit of structured examples for a fixed learner, while RQ3 compares learning systems on identical sufficient examples. RQ4 analyses how dictionary uncertainty affects predictions, and RQ5 tests interventions that address the remaining learning difficulties.

4.1 STUDY DESIGN AND EVALUATION

Constructive recovery supplies $\mathcal{S}_{\mathrm{ref}}$, whose *sufficiency* is established in Section 5.1 (RQ1). Training-set sizes vary by task, averaging approximately 55 examples. For RQ2, we compare structured training with equally sized random and answer-frequency-matched random sets. RQ3 compares all three learning systems on identical $\mathcal{S}_{\mathrm{ref}}$. Matching requires additional labelled candidates; selection costs are reported in Appendix C.

Neural comparisons evaluated 36 tasks across $\mathbb{Z}_7$, $\mathbb{Z}_3^2$, D_4, Q_8 and cube rotations (72 tasks in total under 2 independent seed). Dictionary-revelation and normalisation controls (RQ5) use a paired 16-task subset with identical training examples and initialisations across conditions. The neural composer uses 6,000 AdamW updates with label-balanced cross-entropy, column-sum and entropy penalties, and temperature annealing. Discrete inference uses 128 restarts of at most 80 sweeps. Development sets of 320 labelled examples per task select the learning rate and restart budget; the learning rate is frozen for independent confirmation. Constructive recovery uses no development data.

We report training accuracy and accuracy on 640 held-out tuples per task. Initialisations are averaged within tasks before computing paired differences and 95% task-level bootstrap confidence intervals. Full configurations, statistical procedures and intervention protocols appear in Appendix C.

5 RESULTS AND ANALYSIS

5.1 A1: STRUCTURED COMPLETE-TASK ANSWERS ARE SUFFICIENT FOR EXACT RECOVERY

The successful construction of the Recovery Learner answers RQ1: it exactly determines the rule from complete-task outcomes by querying around a single reference input, using $O(q(p + \log q))$ queries (fig. 3(b)).

In our task F, the complete-task outcome depends jointly on all roles, obscuring the contribution of any individual role dictionary. We therefore construct a binary operation $\odot : Y \times Y \to Y$ to express composition directly in the output alphabet using only complete-task answers. Choose a reference input tuple $a = (a_1, \ldots, a_p)$, called an *anchor*. We then define:

$$\text{Anchor map :} \qquad s_i(t) := F(a_1, \ldots, a_{i-1}, t, a_{i+1}, \ldots, a_p), \tag{5}$$

$$\text{Reference output :} \qquad e_Y := F(a) = s_i(a_i), \tag{6}$$

$$\text{Common reference element :} \qquad H := m_{G,p}(g_1(a_1), \cdots, g_p(a_p)) = \phi^{-1}(e_Y), \tag{7}$$

$$\text{Observed operation :} \qquad y \odot z := \phi\big(\phi^{-1}(y) \oplus H^{-1} \oplus \phi^{-1}(z)\big)\,. \tag{8}$$

Table 1: **Structural validation (A1).** Counts use the conservative implementation. The Fano accuracy evaluates a naive two-role extension after sparse recovery rejects the instance. The reference-correction ablation uses cases with $H \neq e$.

System or intervention	Sparse queries	Full slice	Accuracy
162 associative instances	66.7	190.4	1.00
Cube rotations ($q = 24$)	177.3	645.0	1.00
Planar transformations ($q = 36$)	269.3	1401.0	1.00
Fano quasigroup ($q = 7$)	Rejected	65.5	0.26
Shared role maps	Structural ablation		0.11
Omit reference correction	Conditional ablation		0.20

Each $s_i : Q \to Y$ is a bijection because the dictionaries and composition with fixed group elements are invertible. The reference output e_Y is shared across all roles since $s_i(a_i) = F(a)$ for every i. By construction, $H = \phi^{-1}(e_Y)$, so e_Y is the identity element of $\odot$.

Theorem 1 (Exact factorization)**.** *For every group-based task in equation 2, if the maps s_i and operation $\odot$ are defined by equation 5–equation 8, then for every $x = (x_1, \ldots, x_p) \in Q^p$,*

$$\widehat{F}_{\mathrm{rec}}(x) = s_1(x_1) \odot \cdots \odot s_p(x_p) = F(x). \tag{9}$$

The factor H^{-1} removes the repeated contribution of the anchor each time two anchor maps are combined. Repeated application of $\odot$ therefore cancels the duplicated anchor terms while preserving the original role order, leaving exactly $F(x)$. The full cancellation proof is given in Appendix D.

Moreover, any function in $\mathcal{H}_{q,p}$ consistent with the collected examples yields the same reconstructed function. Hence $\mathcal{S}_{\mathrm{ref}}$ is sufficient within $\mathcal{H}_{q,p}$.

The maps s_i can be acquired by queries changing one input from the anchor. For $1 \leq i < j \leq p$, let $a^{i,j}_{t,v}$ denote the anchor with positions i, j replaced by t, v. A radius-2 query then gives

$$F(a^{i,j}_{t,v}) = s_i(t) \odot s_j(v).$$

Thus we reduce the unknown operation to a binary operation that can be directly queried and reconstructed using established group-based defining-set methods (Bedford et al., 2003). The full derivation is in Appendix E. Exhaustive small-order audits find multiple observed group laws consistent with all radius-one answers, whereas the radius-two examples leave a unique law in all 76 enumerated trials (Appendix F).

5.2 A2: Structured examples improve neural generalisation

Across 72 tasks from independently seeded evaluations, held-out accuracy is 0.28 ± 0.21 for structured examples, 0.12 ± 0.03 for random examples and 0.12 ± 0.03 for histogram-matched random examples (mean $\pm$ task-level SD, after averaging initialisations within tasks). The paired improvement over matched random examples is 0.16 (95% bootstrap CI $[0.12, 0.21]$). It remains positive with equal task family weighting and when any task family is omitted. Example structure therefore improves neural generalisation beyond training-set size and answer balance. Seed-specific results and robustness checks appear in Appendix J.2 and table 8.

5.3 A3: Learning systems differ in how effectively they use sufficient examples

On the same sufficient examples in the matched 36-task comparison, constructive recovery predicts every combination correctly, while discrete inference reaches 0.90 held-out accuracy and the soft composer reaches 0.28 under their respective fixed budgets. Discrete inference recovers the function exactly on 32 tasks; the remaining four cube-rotation tasks underfit. Task family breakdowns appear in Appendix tables 7 and 10. These results show substantial differences in using sufficient information, even when both dictionary-learning models can express the target.

The neural difficulty includes fitting the training answers: across the pooled evaluations, mean structured-training accuracy is 0.79, and only 14 of 136 optimiser runs reach 0.99. Seed-specific training results and learning curves appear in Appendix J and fig. 4.

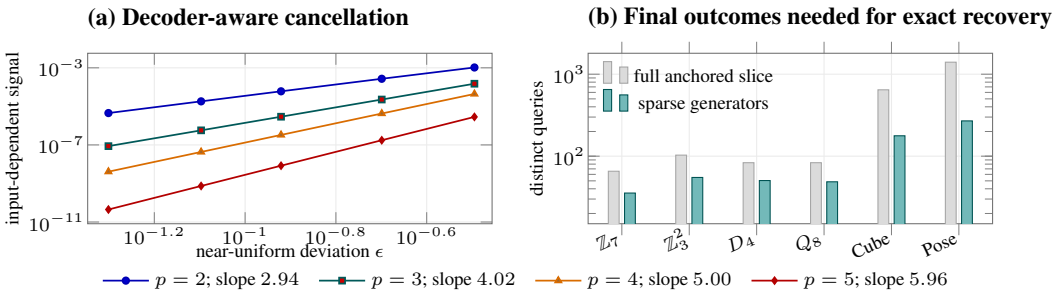

Figure 2: **Numerical signal checks (A4) and reconstruction cost (A1).** Signal slopes follow the decoder-aware local degree. Query counts describe the conservative archived implementation, which queries the final entry of a permutation instead of completing it by elimination.

5.4 A4: Uncertainty propagation suppresses local signals

We analyse the composer near uniform dictionaries. Let $u = \mathbf{1}/q$, where $\mathbf{1}$ is the all-ones vector, and write each input-role distribution as $P_i^\top e_{x_i} = u + \delta_i(x_i)$, with $\mathbf{1}^\top \delta_i(x_i) = 0$. For a Latin operation, combining a uniform distribution with any input distribution returns a uniform distribution, and near uniform distributions behave similarly.

Theorem 2 (Cancellation and local signal order)**.** *For a finite Latin operation with left-associated composition,*

$$\mathcal{B}_p(\mathbf{u} + \delta_1, \ldots, \mathbf{u} + \delta_p) = \mathbf{u} + \mathcal{B}_p(\delta_1, \ldots, \delta_p), \tag{10}$$

If $k \geq 1$ dictionaries (input or output) deviate from uniformity by $O(\epsilon)$ while others remain $O(1)$, input-dependent prediction variation is $O(\epsilon^k)$ Paths towards the correct dictionaries attain these orders.

Multilinearity and the Latin property cancel lower-order mixed terms. With dictionaries uncertain, the first possible input-dependent term has degree $p + 1$. This theorem applies to non-associative quasigroups as well.

Numerical audits verify the aligned-path predictions to a maximum residual of 1.11×10^{-16} (Appendix N). For degrees 3,4,5,6, the actual measured slopes agree with values 2.94, 4.02, 5.00, and 5.96 (fig. 3(a)). Proofs and parameter-gradient analysis appear in Appendices K, L, and M.

5.5 A5: Resolving Component Meanings Improves Learning

5.5.1 Revealing Input or Output Dictionaries

On the same 16-task subset over 4 task families, we fix correct dictionaries and train the remainder with the original examples and budget. For the one input revealed runs, we average over two prechosen input positions per task. Revealing one input improves held-out accuracy from 0.30 to 0.64 (table 2), while revealing only the output dictionary increases it to 0.82. Thus, output-only revelation exceeds input-only by 0.18 $[0.07, 0.28]$. The advantage is positive in all four task family means. If all dictionaries except one input dictionary are revealed, every run achieves perfect training and held-out accuracy.

The interventions have matched local orders but different outcomes. Their effects depend on which component is supplied under the tested training objective, and the input-map penalties are averaged across roles while the decoder has its own penalty term. This comparison shows the combined effect of component identity and regularisation differences.

Decoder revelation increases initial task gradient median from 4.9×10^{-9} to 3.08×10^{-7}, while the first actual AdamW role update remains approximately 0.05 in both conditions. Thus the optimiser can produce similar update magnitudes despite differences in local task-gradient scale. Full diagnostics appear in section O.3.

Table 2: **Dictionary revelation (A4–A5).** Known dictionaries are supplied and fixed. Degree is the local order attained along paths towards correct dictionaries.

Input dictionaries	Output dictionary	Degree	Train	Held-out
All unknown	Unknown	$p+1$	0.84	0.30
One known	Unknown	p	0.89	0.64
All unknown	Known	p	0.97	0.82
One known	Known	$p-1$	0.96	0.89
All but one known	Known	1	1.00	1.00

5.5.2 Enforcing continuous assignment constraints

We test whether enforcing a continuous relaxation of one-to-one dictionaries improves gradient learning. Tolerance-converged Sinkhorn normalisation with implicit differentiation (Mena et al., 2018; Eisenberger et al., 2022) enforces unit row and column sums, with maximum numerical residual 2.38×10^{-7}. Across 16 paired tasks, the mean held-out accuracy change is -0.002 (95% CI $[-0.079, 0.073]$; Appendix table 19). Effects depend on the task family: accuracy improves by 0.281 for $\mathbb{Z}_3^2$ but declines for D_4 and Q_8. Accurate continuous constraints therefore yield no clear aggregate improvement and leave substantial learning difficulty. Numerical validation and post-training projection checks appear in Appendix O.4.

5.5.3 Providing Intermediate Result Supervision

We also test intermediate supervision, where the learner is given the output of a contiguous subcomposition using the same operation and hidden input and output dictionaries. We compare unary, binary, and ternary intermediate supervision, histogram-matched shuffled binary supervision, and budget matched additional complete-task supervision. The experiment crosses seven operation-arity settings with ten optimiser seeds and three learning rates (Appendix P.1).

Binary intermediate supervision has higher mean competence than the low complete-answer baseline, but the contrast is uncertain $(+0.24[-0.04, 0.49])$. Binary supervision has mean advantage over equal label budget of additional complete answers, but this contrast is also uncertain $(+0.20[-0.05, 0.42])$. Correct binary labels strongly outperform histogram-preserving shuffled binary labels $(+0.64[+0.33, +0.87])$, showing that the semantic consistency of the intermediate labels matters.

6 Discussion and Conclusion

The results connect example selection with representation design by showing how relationships between complete-task answers can define shared coordinates in which components combine into an executable rule. These coordinates can differ from the original hidden dictionaries while preserving the composition needed to predict new combinations, exposing reusable structure when intermediate labels are unavailable.

Even when this structure is available, learning compatible meanings can remain difficult. The cancellation law describes their joint dependence near uniformity, while the interventions show that local prediction order alone cannot explain learning outcomes. Interpreting these outcomes also requires considering which meanings are supplied, their regularisation and the learning procedure. A next step is to infer partial correspondences from complete answers and use them to guide neural training, connecting constructive inference with representation learning.

The guarantees apply to finite-group compositions with bijective dictionaries and selectable queries; neural comparisons use a supplied composition table and finite training budgets. Extending this approach to non-invertible meanings and passively collected examples requires further work. Comparing learners on complete-task examples that guarantee exact reconstruction makes the gap between sufficient supervision and successful compositional learning directly measurable.

AI USE STATEMENT

In this work, we used generative AI tools for improving language, identifying potentially relevant or related publications, suggesting mathematical concepts and tools for the authors to investigate, for assistance with Python-based analysis of raw experimental results, and for assistance with coding, converting author-derived mathematical expressions into LaTeX, tables and plotting code. We did not use generative AI tools to autonomously conduct the research, to generate the core research ideas or hypotheses of the work, or to write the manuscript in full. All AI-assisted material was reviewed by the authors before inclusion. Suggested publications and mathematical tools were independently checked by the authors for relevance and correctness. AI-assisted code, equations, and plotting code were inspected, tested, and revised by the authors as appropriate, and the resulting experimental outputs and mathematical content were verified as part of our normal research process. AI-assisted language edits were reviewed to ensure that they accurately reflected the authors' intended meaning and claims. We take responsibility for the final content of this work, including all text, claims, code, mathematical expressions, figures, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study uses synthetic mathematical tasks and involves no human participants or personal data. We foresee no specific ethical concerns arising from the reported methods or experiments.

REPRODUCIBILITY STATEMENT

Operation definitions and experimental protocols, including task generation, random seeds, hyperparameters and evaluation, appear in Appendices B and C. The appendix also provides complete proofs and algorithmic details, with seed-specific results and additional analyses in Appendix J.

REFERENCES

Pranjal Awasthi, Vitaly Feldman, and Varun Kanade. Learning using local membership queries. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pp. 398–431. PMLR, 2013. URL https://proceedings.mlr.press/v30/Awasthi13.html.

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022. doi: 10.1016/j.artint.2021.103649. URL https://doi.org/10.1016/j.artint.2021.103649.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21750–21764. Curran Associates, Inc., 2022. doi: 10.52202/068431-1581. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/884baf65392170763b27c914087bde01-Abstract-Conference.html.

David Bedford, Matthew Johnson, and M. A. Ollis. Defining sets for Latin squares given that they are based on groups. *European Journal of Combinatorics*, 24(1):129–135, 2003. doi: 10.1016/S0195-6698(02)00115-4. URL https://www.sciencedirect.com/science/article/pii/S0195669802001154.

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 *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d1d11bf8299334d354949ba8738e8301-Abstract-Datasets_and_Benchmarks_Track.html.

Hoyeon Chang, Jinho Park, Hanseul Cho, Sohee Yang, Miyoung Ko, Hyeonbin Hwang, Seungpil Won, Dohaeng Lee, Youbin Ahn, and Minjoon Seo. Characterizing pattern matching and its limits on compositional task structures. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/29f1595d10229ee89dcc8abbbd9c5d87-Abstract-Conference.html.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6243–6267. PMLR, 2023. URL https://proceedings.mlr.press/v202/chughtai23a.html.

Amit Daniely and Eran Malach. Learning parities with neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20356–20365. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/eaae5e04a259d09af85c108fe4d7dd0c-Abstract.html.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3081. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/deb3c28192f979302c157cb653c15e90-Abstract.html.

Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, Florian Bernard, and Daniel Cremers. A unified framework for implicit Sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Dongsung Huh. Discovering group structures via unitary representation learning. In *International Conference on Learning Representations*, pp. 3512–3536, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/09d320a5b92d74bbde3d0c4f52e680a9-Abstract-Conference.html.

Denis S. Krotov and Vladimir N. Potapov. On connection between reducibility of an n-ary quasigroup and that of its retracts. *Discrete Mathematics*, 311(1):58–66, 2011. doi: 10.1016/j.disc.2010.09.023. URL https://www.sciencedirect.com/science/article/pii/S0012365X10003742.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882. PMLR, 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 562–570. PMLR, 2015. URL https://proceedings.mlr.press/v38/lee15a.html.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/dc5d637ed5e62c36ecb73b654b05ba2a-Abstract.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 *Advances in Neural Information Processing Systems*, volume 36, pp. 72507–72539, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e560202b6e779a82478edb46c6f8f4dd-Abstract-Conference.html.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Byt3oJ-0W.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3067–3075. PMLR, 2017. URL https://proceedings.mlr.press/v70/shalev-shwartz17a.html.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4506–4515. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1458. URL https://aclanthology.org/D19-1458/.

Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking group multiplication with cosets. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46441–46467. PMLR, 2024. URL https://proceedings.mlr.press/v235/stander24a.html.

Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D. Lee, and Denny Wu. Learning compositional functions with transformers from easy-to-hard data. In *Proceedings of the 38th Annual Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 5632–5711. PMLR, 2025. URL https://proceedings.mlr.press/v291/wang25a.html.

Wilson Wu, Louis Jaburi, Jacob Drori, and Jason Gross. Towards a unified and verified understanding of group-operation networks. In *International Conference on Learning Representations*, pp. 61683–61714, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9a583178bd18b2e55095c3a53f8fe88a-Abstract-Conference.html.

Aleksey D. Yashunsky. On the rate of convergence of quasigroup convolutions of probability distributions. *Discrete Mathematics and Applications*, 34(1):51–59, 2024. doi: 10.1515/dma-2024-0005. URL https://doi.org/10.1515/dma-2024-0005.

A Notation, terminology and assumptions

The appendices follow the research questions in the main paper. The construction and information requirements for RQ1 are proved in sections D to F; section D.3 gives a complete order-dependent example. The boundary and noise analyses appear in sections G and H. The local-signal result for RQ4 and its connection to the dictionary interventions of RQ5 are developed in sections K to M.

Throughout, Q is the visible input alphabet, G the latent alphabet and Y the output alphabet, with $|Q| = |G| = |Y| = q \geq 2$. The full-task construction assumes $p \geq 2$. We write

$$\mathbf{u} = \mathbf{1}/q, \qquad T_0 = \{v \in \mathbb{R}^q : \mathbf{1}^\top v = 0\}, \qquad J = \mathbf{1}\mathbf{1}^\top/q.$$

An occurrence of $\mathbf{u}$ is indexed by the alphabet of the surrounding expression. Likewise, e_g is a one-hot vector for a latent element, whereas e_y is indexed by output symbols. The composition operator $\mathcal{B}_p$ uses a left-associated bracket convention. This convention is relevant for quasigroups; associativity makes it immaterial for groups.

The recovery result assumes a finite group, bijective dictionaries and adaptive queries for complete answers. The local-signal result requires only a deterministic Latin operation and independent categorical operands. The group promise used by recovery supplies associativity; the neural composer additionally receives the true operation table. All asymptotic signal estimates hold for fixed p and q in finite-dimensional norms.

B Formal definitions of the task families

Task families. The families below share equation 2, with different composition systems $(G, \oplus)$.

Modular compositions. For $G = \mathbb{Z}_n = \{0, \ldots, n-1\}$, composition is addition modulo n: $r \oplus s = (r + s) \bmod n$. Product groups apply modular addition separately to each coordinate. The principal neural studies use $\mathbb{Z}_7$ with $(q, p) = (7, 5)$ and $\mathbb{Z}_3^2$ with $(q, p) = (9, 4)$, giving 16,807 and 6,561 possible input combinations, respectively.

Table 3: Terminology for the controlled learning problems.

Term	Meaning
Arity p	Number of input positions in a complete task.
Cardinality q	Number of elements in each input, latent and output alphabet.
Operation table	The result of combining every ordered pair of latent elements.
Latin operation	Each result occurs once in each row and each column of the table. Either input and the result determine the other input.
Quasigroup	A set equipped with a Latin operation; associativity is not assumed.
Group	An associative operation with an identity and inverses. Its table is Latin.
Noncommutative	Reversing the order of two elements can change their product.
Dictionary	A bijection assigning meanings to visible input or output symbols. Different positions have independent dictionaries.
Complete-answer label	The final output for all p inputs.
Intermediate label	An answer for a specified shorter partial composition.
Local signal order	Lowest total degree at which an input-dependent term can appear near uniform dictionary probabilities.
Membership query	A complete input chosen by the learner and labelled by an answer oracle.
Anchor	A fixed complete reference input.
Intervention radius	Number of positions changed relative to the anchor. Every query still evaluates a complete task.
Anchor map (star)	Map from a position's symbols to the answers obtained by changing only that position.
Binary retract	A binary operation obtained by fixing the other arguments of a multi-input operation.
Generator	An element of a set whose products and inverses reach the entire group.
Held-out accuracy	Accuracy on unobserved complete tuples at the same length and under the same dictionaries. Archived result files use "OOD" for this metric.
Training competence	Training accuracy of at least 0.99, the predefined diagnostic threshold.

Order-sensitive compositions. We use noncommutative groups D_4 and Q_8, both with $(q, p) = (8, 4)$. The dihedral group D_4 has underlying set $\mathbb{Z}_4 \times \mathbb{Z}_2$ and represents rotations and reflections of a square, with operation

$$(r, b) \oplus (s, c) = \big((r + (-1)^b s) \bmod 4,\ (b + c) \bmod 2\big).$$

The quaternion group $Q_8 = \{\pm 1, \pm i, \pm j, \pm k\}$ uses quaternion multiplication, determined by $i^2 = j^2 = k^2 = ijk = -1$. These tasks require preserving the order of latent elements, since reordering can change the result.

Spatial transformation compositions. We study cube rotations with $(q, p) = (24, 3)$ and planar transformations with $q = 36$ and $p \in \{3, 5, 7\}$. The cube-rotation group

$$G_{\text{cube}} = \{R \in \mathbb{Z}^{3\times 3} : R^\top R = I_3,\ \det R = 1\}$$

contains the 24 rotations of a cube, with $R \oplus S = RS$; I_3 is the identity matrix. The planar group consists of $(v, r) \in \mathbb{Z}_3^2 \times \mathbb{Z}_4$, representing a translation on a periodic 3×3 grid and a rotation by r quarter-turns. Its operation is

$$(v, r) \oplus (w, s) = \big((v + R_r w) \bmod 3,\ (r + s) \bmod 4\big),$$

where R_r rotates w by r quarter-turns and modulo three applies to each translation coordinate. Nine translations and four rotations give $q = 36$ elements. Both groups are noncommutative.

Non-associative quasigroup compositions. As a boundary test, we use the Fano quasigroup with $G = \{0, \ldots, 6\}$ and

$$r \oplus s = \begin{cases} r, & r = s, \\ \big((r + 1)\ \text{XOR}\ (s + 1)\big) - 1, & r \neq s, \end{cases}$$

where XOR is bitwise exclusive OR. A quasigroup is a set with a Latin operation. This operation is non-associative: $(0 \oplus 1) \oplus 2 = 2$, whereas $0 \oplus (1 \oplus 2) = 0$. Compositions follow the left-associated order in equation 1.

C TASK GENERATION, EXPERIMENTAL ACCESS AND REPRODUCIBILITY

The experiments address different parts of the research-question sequence. The structural audit checks the construction in RQ1; the locked neural studies address RQ2–RQ3; and the signal and intervention suites address RQ4–RQ5. Their cohorts and budgets are recorded separately below. Throughout, a task unit specifies an operation, an arity and an independent set of input and output relabellings. Optimiser initialisations are repeated measurements within that unit.

C.1 TASK FAMILIES AND INDEPENDENT SYMBOL DICTIONARIES

The broad reconstruction suite contains cyclic groups of sizes 5, 7 and 11; products $\mathbb{Z}_2^3$ and $\mathbb{Z}_3^2$; dihedral groups D_3, D_4, D_5 of sizes $6, 8, 10$; the quaternion group Q_8; the order-24 cube-rotation group; and an order-36 planar-motion group. For each task, independent random permutations encode every input position and the output. The cube and planar-pose tasks model exact finite transformations. Their inputs are symbols for transformations, without image perception or continuous physical dynamics. The complete reconstruction coverage is given in table 5.

C.2 FROZEN BROAD PAPER-PROFILE STUDY

The original paper-profile archive uses master seed 20260823. Its local theorem sweep uses 40 seeds, arities two through eight, and initialisation scales

$$.002, .004, .008, .015, .03, .06, .12, .25, .5, .8.$$

The identifiability suite uses five independently generated tasks and five neural initialisations per condition, with up to 3,500 updates. The supervision factorial crosses ten optimiser seeds with learning rates $\{.003, .008, .02\}$ and a maximum of 3,500 updates. Its seven operation–arity settings are $\mathbb{Z}_7$ at arities three and five; $\mathbb{Z}_2^2$ and $\mathbb{Z}_3^2$ at arity four; D_3 and D_4 at arity four; and a nonassociative seven-element quasigroup at arity four. Architecture controls use ten seeds and the same maximum horizon.

The soft composer learns a $q \times q$ logit matrix at each input position and one for the output dictionary, with the true operation tensor fixed. This implementation uses entropy and column penalties of .002 and .05. The MLP has position-specific learned token embeddings and a feedforward classifier. The Transformer has width 72, two encoder layers, four attention heads, feedforward width 216 and no dropout. The separate operation/architecture-specificity suite uses learning rates .012 for the composer, .002 for the MLP and .0008 for the Transformer. The visible-affine comparison uses .02 for the composer, with the same MLP and Transformer rates. The supervision factorial uses the three crossed rates above. AdamW weight decay is 10^{-4}.

C.3 LOWER-BUDGET DISCOVERY AND CONFIRMATION

The exploratory reconstruction and learning runs use master seeds 2026082519 and 2026082619, respectively, with a frozen common configuration. They use four surface relabellings for ordinary structural settings, three neural-task relabellings, arities $\{3, 4, 5, 7\}$, a maximum exact-evaluation size of 2,048, a neural evaluation size of 640, and batch size 96. Larger transformation groups use the coverage listed in table 5. Direct neural methods receive 1,100 AdamW updates at learning rate .02, weight decay 10^{-4} and a temperature schedule from 1 to .12. Student controls receive 650 updates at learning rate .0015; their small Transformer has two encoder layers and four attention heads. The later locked studies extend these pilots and are reported separately.

C.4 TWO LOCKED PAPER-SCALE NEURAL STUDIES

The first paper-scale study preserves master seed 2026082619, the task constructor, permutation namespaces, composer, temperature schedule, penalties and test-set seeds. Eighteen historical 1,100-update records are reused only where task, initialisation, learning rate, horizon and label histogram match exactly. Fresh conditions cover $\mathbb{Z}_7$ at $p = 5$; $\mathbb{Z}_3^2, D_4, Q_8$ at $p = 4$; and cube rotations at $p = 3$. The first four families each have eight independent relabellings and two initialisations; cube rotations have four relabellings and one initialisation. This yields 36 task units and 68 trajectories per condition.

Learning rates $\{.008, .02, .05\}$ are compared at $1{,}100$ updates. The pooled rate $.05$ is selected exclusively on a disjoint development split. Incomplete historical blocks without saved development predictions are excluded from selection. Final runs use $6{,}000$ updates. One prespecified relabelling per family also receives $12{,}000$ updates. Training, development and test tuples are pairwise disjoint.

The independent paper-scale study uses master seed 2026090601, fixes the learning rate at $.05$ and preserves the five families, task counts, label budgets, architecture, optimiser, batch size, evaluation checkpoints and $6{,}000$-update horizon. Task dictionaries, anchors, data splits, initialisations and minibatch sequences are regenerated. All 36 task units pass the predefined structural and leakage audits. The two studies are reported separately; the second uses no current-seed hyperparameter selection.

C.5 Information, computation and selection budgets

Learning objective. At temperature τ, let P_i and D denote the row-softmax dictionary probabilities. The inherited column and entropy penalties are

$$\mathcal{C} = \frac{1}{pq} \sum_{i=1}^{p} \sum_{r \in G} \left(\sum_{x \in Q} P_i[x, r] - 1 \right)^2 + \frac{1}{q} \sum_{y \in Y} \left(\sum_{r \in G} D[r, y] - 1 \right)^2, \tag{11}$$

$$\mathcal{E} = -\frac{1}{pq} \sum_{i=1}^{p} \sum_{x \in Q, r \in G} P_i[x, r] \log P_i[x, r] - \frac{1}{q} \sum_{r \in G, y \in Y} D[r, y] \log D[r, y]. \tag{12}$$

The minibatch objective is $\mathcal{L} = |\mathcal{M}|^{-1} \sum_{(x,y) \in \mathcal{M}} - \log f_\theta(x)_y + .05\mathcal{C} + .002\mathcal{E}$. In the two locked principal studies, each minibatch entry first samples an output label uniformly from the labels present in training, then samples an example uniformly from that label's bucket, with replacement. Let n_y count training examples with answer y and $Y_{\mathcal{S}} = \{y : n_y > 0\}$. The expected objective is

$$\mathcal{L}(\theta) = -\frac{1}{|Y_{\mathcal{S}}|} \sum_{y \in Y_{\mathcal{S}}} \frac{1}{n_y} \sum_{t:\, y^{(t)} = y} \log f_\theta(x^{(t)})_y + .05\mathcal{C} + .002\mathcal{E}. \tag{13}$$

The three neural data conditions share this label-balanced sampler. The implementation clamps probabilities below 10^{-30} before taking logarithms. The role terms are averaged over all p dictionaries, whereas the decoder contributes one separate term. This is the component-specific regularisation retained in the matched revelation experiment.

The soft composer and discrete assignment procedure receive the true latent operation table. Constructive recovery receives p, q and the logged complete-answer query interface under the group-class assumption, with the latent table and permutations kept hidden.

The answer-frequency-matched random condition is an experimenter-generated control. Candidate outcomes are labelled by the simulator until the delivered dataset has the same size and exact output histogram as the structured dataset. On average, 653.3 labelled candidates yield 54.4 delivered examples. These screening calls are additional information costs. This control also has broader position-wise symbol coverage: its mean role-marginal total variation is $.041$, compared with $.596$ for the structured data.

The broad paper-profile run takes 576.16 CUDA-minutes. The first paper-scale supplement, including development, structural checks, robustness conditions and figure exports, takes 184.3 minutes. The independent locked neural run takes 29.17 minutes. An independent CPU audit of recovery and associated checks takes 2.48 minutes. Intervention-specific costs are given with their results.

C.6 Metrics, leakage audits and statistical units

Direct comparisons evaluate complete tuples absent from the relevant training or oracle-query set, at the same arity. A training-competent run has training accuracy at least $.99$. We report training accuracy alongside held-out accuracy, since failure to fit and failure after fitting answer different questions. Area under the learning curve (AULC) averages the predefined held-out checkpoints.

Initialisations, and role-subset choices where relevant, are averaged within task units before inference. Bootstrap intervals resample tasks rather than individual optimisation trajectories. The paper-scale

Table 4: Resource accounting for the principal learning comparison. Wall times describe the archived hardware. Selection uses disjoint development data.

Procedure	Training or search	Development use	Structural information
Soft composer, each data condition	68 runs, 6,000 updates; median 31.4 s/run; 36.3 GPU-min per condition	One rate selected from $\{.008, .02, .05\}$; frozen for independent confirmation	True operation table
Matched-random data	653.3 candidates screened for 54.4 delivered labels, on average	Same frozen learner settings	Simulator labels for screening
Discrete assignment	128 restarts; at most 80 alternating Hungarian sweeps; 2.06 CPU-min total	Restart budget selected on development data; candidate selected by training fit	True operation table
Constructive recovery	Deterministic algebraic reconstruction; zero optimiser updates	None	p, q, group promise and answer queries

studies use 36-unit bootstraps separately within each master seed; a pooled 72-unit analysis, when used, resamples task units in seed blocks. The 16-task control studies resample four tasks within each of four fixed families, so their intervals describe those families. The supervision factorial uses task-clustered contrasts. Each result states the applicable cohort.

The mean query reduction is the average of the per-instance reductions,

$$\frac{1}{n}\sum_{j=1}^{n}\left(1-\frac{N_{\mathrm{sparse},j}}{N_{\mathrm{full},j}}\right).$$

It differs from taking the ratio of the two mean query counts, which weights larger tasks differently. Raw query counts are retained alongside this summary.

D Observable coordinates and exact recovery

This section proves the factorisation used to answer RQ1. It explains why the selected complete answers determine an executable rule even when the original dictionaries and operation table are hidden from the recovery learner. We retain $\oplus$ for the latent operation and $\odot$ for the recovered operation on answer symbols. In compact products below, juxtaposition abbreviates $\oplus$ and powers denote repeated composition; no additional group operation is introduced.

D.1 Observed group law

Fix an anchor $a = (a_1, \dots, a_p)$ and write

$$h_i = g_i(a_i), \qquad H = h_1 \cdots h_p, \qquad e_Y = \phi(H).$$

Define the observed operation and a change of coordinates by

$$y \odot z = \phi\big(\phi^{-1}(y)H^{-1}\phi^{-1}(z)\big), \qquad \psi(y) = \phi^{-1}(y)H^{-1}.$$

Both ϕ^{-1} and right multiplication by H^{-1} are bijective. Moreover,

$$\psi(y \odot z) = \phi^{-1}(y)H^{-1}\phi^{-1}(z)H^{-1} = \psi(y)\psi(z).$$

Thus ψ is a group isomorphism. In particular, $\odot$ is associative, its identity is e_Y, and each observed symbol has an inverse.

The anchor map for position i is

$$s_i(t) = F(a_1, \dots, a_{i-1}, t, a_{i+1}, \dots, a_p) = \phi(V_i(t)),$$

where

$$V_i(t) = h_1 \cdots h_{i-1} g_i(t) h_{i+1} \cdots h_p.$$

The map $s_i : Q \to Y$ is bijective, since it is a composition of a bijection, fixed left and right group multiplications, and the output bijection. Also $s_i(a_i) = e_Y$ for every position.

D.2 ORDERED FACTORISATION

For each $1 \le k \le p$, we claim that

$$V_1(x_1)H^{-1}V_2(x_2)\cdots H^{-1}V_k(x_k) = g_1(x_1)\cdots g_k(x_k)h_{k+1}\cdots h_p. \tag{14}$$

The case $k = 1$ follows from the definition of V_1. For the induction step, write $A_k = h_1 \cdots h_k$, $S_k = h_{k+1}\cdots h_p$ and $S_{k+1} = h_{k+2}\cdots h_p$. Then $H = A_k S_k$ and $V_{k+1}(x_{k+1}) = A_k g_{k+1}(x_{k+1})S_{k+1}$. Consequently,

$$\begin{aligned}
&[g_1(x_1)\cdots g_k(x_k)S_k]H^{-1}V_{k+1}(x_{k+1}) \\
&\quad = g_1(x_1)\cdots g_k(x_k)\,\underbrace{S_k(A_kS_k)^{-1}A_k}_{e}\,g_{k+1}(x_{k+1})S_{k+1} \\
&\quad = g_1(x_1)\cdots g_{k+1}(x_{k+1})S_{k+1}.
\end{aligned}$$

Setting $k = p$ proves

$$F(x_1,\ldots,x_p) = s_1(x_1)\odot\cdots\odot s_p(x_p).$$

Every cancellation respects the original order. The proof therefore covers noncommutative groups. Setting all but the first two positions to their anchor values also gives

$$s_1(t)\odot s_2(v) = F(t, v, a_3, \ldots, a_p).$$

This identifies the observed law through complete-task queries and proves the factorisation part of theorem 1.

Role of the reference correction. Each s_i includes all the unchanged anchor inputs, so the factor H^{-1} removes their repeated contribution when two observed coordinates are combined. The observed identity is therefore $F(a)$, which can differ from the output label of the original latent identity. Choosing another anchor changes the recovered coordinates while leaving the predicted function unchanged. Exact recovery is defined at the level of this observable function.

D.3 WORKED EXAMPLE: ROTATIONS AND REFLECTIONS OF A TRIANGLE

Consider the six symmetries of an equilateral triangle. Write e for doing nothing, r for a one-third turn and s for a reflection. We fix the algebraic convention

$$r^3 = e, \qquad s^2 = e, \qquad sr = r^2 s.$$

The last relation shows that order matters. The hidden elements are e, r, r^2, s, rs, r^2s, which the output dictionary names $\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{F}$, respectively. Three positions use tokens $0, \ldots, 5$ and the dictionaries

input token	0	1	2	3	4	5
g_1	r	s	e	r^2	rs	r^2s
g_2	s	r^2	r	e	r^2s	rs
g_3	r	r^2	s	r^2s	e	rs

The learner receives complete answers; the displayed hidden meanings let us check the construction. At $a = (0, 0, 0)$, the latent product is $H = rsr = s$, so $F(a) = \mathrm{D}$ is the observed identity.

For $x = (1, 1, 1)$, the anchor queries give

queried tuple	latent product	answer
$(1, 0, 0)$	$ssr = r$	B
$(0, 1, 0)$	$rr^2r = r$	B
$(0, 0, 1)$	$rsr^2 = r^2s$	F

Multiplying the latent star values without correction gives $r\,r\,(r^2s) = rs$ and predicts E. The actual answer is $\phi(sr^2r^2) = \phi(r^2s) = \mathrm{F}$.

Since $H^{-1} = s$, the observed operation is

$$y \odot z = \phi\big(\phi^{-1}(y)s\phi^{-1}(z)\big)\,.$$

It gives $\mathrm{B}\odot\mathrm{B} = \mathrm{D}$ and $\mathrm{D}\odot\mathrm{F} = \mathrm{F}$. In latent coordinates this is $r\,s\,r\,s\,(r^2s) = r^2s$, the required ordered cancellation. Changing the role order gives a different result: $\mathrm{B}\odot\mathrm{F} = \mathrm{C}$ and $\mathrm{C}\odot\mathrm{B} = \mathrm{E}$. The reference correction aligns the dictionaries while preserving the order-sensitive rule.

E ADAPTIVE RECOVERY AND EXACT QUERY ACCOUNTING

The factorisation reduces RQ1 to learning the anchor maps and an observed binary group law. We now give the complete query algorithm and its cost.

Theorem 3 (Pairwise recovery and information requirements)**.** *For $p \geq 2$ and $q \geq 2$, every group-based task in eq.* (2) *can be recovered by an adaptive algorithm using complete-answer queries of radius at most two. Its query count is at most*

$$N \leq 1 + (p + d)(q - 2), \qquad d \leq \lfloor \log_2 q \rfloor. \tag{15}$$

For a hypothesis class containing two distinct observed group laws on the same output alphabet with a common identity, radius-one queries are insufficient in the worst case. Even when the observed law is supplied, arbitrary anchored position-wise bijections require

$$N \geq \left\lceil p \log_q((q-1)!) \right\rceil = \Omega(pq) \qquad (q \geq 3) \tag{16}$$

labels for worst-case exact identification.

E.1 ALGORITHM

Given Q, Y, p and the group-and-bijection promise, the learner receives an oracle for complete inputs, with the table and dictionaries kept hidden.

1. Query the anchor a and set $e_Y = F(a)$.
2. For each position i, choose a token $b_i \neq a_i$. Query the anchor map $s_i(t)$ for every $t \notin \{a_i, b_i\}$. Set $s_i(a_i) = e_Y$ and infer $s_i(b_i)$ as the unique unused output symbol. Store the map and its inverse.
3. Initialise a set of known right translations with the identity permutation. The generated subgroup initially contains only e_Y.
4. If the generated subgroup is not all of Y, choose c outside it and set $v = s_2^{-1}(c)$. Choose any $b \in Y \setminus \{e_Y\}$. For each $y \notin \{e_Y, b\}$, set $t = s_1^{-1}(y)$ and query $F(t, v, a_3, \ldots, a_p)$. These replies are entries of the right translation $R_c(y) = y \odot c$. The entry $R_c(e_Y) = c$ is already known; infer $R_c(b)$ by permutation completion.
5. Close the known translations under permutation composition and inversion. Identify each resulting translation R_g by $R_g(e_Y) = g$. Under the group promise, equal identity images give equal translations. An implementation may reject inconsistent maps.
6. Repeat steps 4–5 until all q translations are known. Assemble the recovered table with entry $y \odot g = R_g(y)$, check the group axioms and predict by left-folding $s_1(x_1), \ldots, s_p(x_p)$ in position order.

Every inferred last entry is justified by the corresponding map being a permutation. The anchor maps therefore use exactly $1 + p(q - 2)$ distinct queries. Each new generator translation uses $q - 2$ additional queries; these vary two positions and occupy a previously unqueried column. For d chosen generators the total is $1 + (p + d)(q - 2)$.

This also covers $q = 2$. A bijection on two symbols is determined by its anchor entry, and the group law on two labelled symbols is determined by its identity. All maps and translations can therefore be completed from the single anchor answer, giving the stated bound $N \leq 1$.

E.2 EXACT CLOSURE OF THE GENERATOR TRANSLATIONS

For observed group elements g, c,

$$R_c \circ R_g = R_{g \odot c}, \qquad R_g \circ R_c = R_{c \odot g}.$$

These identities follow by applying the translations to an arbitrary element and using associativity. Inversion gives $R_g^{-1} = R_{g^{-1}}$. Hence permutation closure recovers precisely the translations of the subgroup generated so far. The identity image identifies the element represented by each translation.

Let H_j be the subgroup after j generator columns. Choosing a new generator outside H_j gives $H_j \subsetneq H_{j+1}$. Lagrange's theorem makes $|H_{j+1}|/|H_j|$ an integer of at least two. Thus $2^d \le q$ and $d \le \lfloor \log_2 q \rfloor$.

In array notation, if `current` represents R_g and `generator` represents R_c, then `generator[current]` represents $R_c \circ R_g$. Preserving this order is essential for the noncommutative cases.

E.3 Implementation query counts and binary reconstruction

The archived implementation queried every nonanchor star entry and every nonidentity entry of a selected translation. Its count is therefore

$$N_{\text{arch}} = 1 + (p + d)(q - 1).$$

All empirical query tables and the noise experiment use this conservative implementation. Equation 15 removes one redundant entry from each permutation in the noiseless setting.

For comparison, acquiring the entire first-two-position slice and the remaining anchor maps costs

$$N_{\text{full}} = q^2 + (p - 2)(q - 1).$$

The first two stars and the anchor already belong to that slice. When $p = 2$, the archived sparse count is

$$1 + (2 + d)(q - 1) = q + (q - 1)(d + 1),$$

the group-based Latin-square defining-set count of Bedford et al. (2003). Binary retracts are classical algebraic objects (Krotov & Potapov, 2011). Our full-task construction uses these ingredients after recovering the common observable coordinates.

F Intervention radius and label complexity

The remaining parts of theorem 3 distinguish the spatial reach of a query from the length of the underlying task. Radius is measured from a single fixed anchor, as defined in section 5.1; every oracle call still evaluates all p positions.

F.1 Limits of radius-one queries

Radius-one statement in theorem 3. Let $\odot_1$ and $\odot_2$ be different group laws on Y with a common identity e. Choose bijections $s_i : Q \to Y$ satisfying $s_i(a_i) = e$, and define

$$F_j(x) = s_1(x_1) \odot_j \cdots \odot_j s_p(x_p), \qquad j \in \{1, 2\}.$$

Both are valid hidden-group tasks, using the corresponding observed group as the latent group and an identity output dictionary. Their anchor answers are e. On the ith radius-one arm, both return $s_i(x_i)$, since every other input maps to the identity. Their answers to all radius-one queries are therefore equal.

Because the laws differ and share their identity, some nonidentity pair y, z satisfies $y \odot_1 z \neq y \odot_2 z$. Set $x_1 = s_1^{-1}(y)$, $x_2 = s_2^{-1}(z)$ and leave all later positions at the anchor. This radius-two tuple distinguishes the functions. A set of radius-one examples cannot identify both members of the class. □

The proposition concerns task classes with an unknown observed law that contain the two laws in the construction. For the neural learner, which receives a latent operation table, its applicability depends on the supplied-table restriction and group order.

Radius of the constructive algorithm. The anchor has radius zero; the star queries have radius one; and translation queries have the form $F(t, v, a_3, \ldots, a_p)$, with radius at most two. All subsequent calculations use permutations and request no new labels. This proves the upper radius bound in theorem 3.

F.2 A LOWER BOUND FROM INDEPENDENT DICTIONARIES

Label-count statement in theorem 3. Supply the learner with a fixed group $(Y, \odot)$ and its identity e. For each position there are $(q-1)!$ bijections $s_i : Q \to Y$ with $s_i(a_i) = e$. The resulting family

$$F_s(x) = s_1(x_1) \odot \cdots \odot s_p(x_p)$$

contains $((q-1)!)^p$ different functions: the radius-one arm for position i directly reveals s_i, so different map tuples yield different functions.

A deterministic adaptive decision tree with at most N queries has at most q^N response leaves. Exact identification requires a distinct leaf for each function, whence $q^N \geq ((q-1)!)^p$. Taking logarithms proves eq. (16). The same counting argument applies after fixing the random seed of any procedure required to be exact for every task and every random seed. Stirling's formula gives $\log_q((q-1)!) = \Theta(q)$ as q grows. This establishes the stated $\Omega(pq)$ dependence while already supplying the observed operation law to the learner. □

Finite audit. The archived radius audit replays the 162 associative recovery instances. All requested tuples have radius at most two, and each audited task uses an informative radius-two query. Exhaustive small-order enumeration finds 4, 80, 2,760 and 7,560 observed laws compatible with the complete set of radius-one examples at orders 4, 6, 8 and 9, respectively. The radius-two examples leave one law in all 76 enumerated trials. These enumerations illustrate the proposition and are reported separately from the independently relabelled learning tasks.

G NONASSOCIATIVE AND INTERFACE BOUNDARIES

The local-signal theorem uses the Latin property; reconstruction also uses associativity. A concrete counterexample makes this distinction explicit. On $G = \{0, \ldots, 6\}$, define the Fano quasigroup by

$$a \oplus b = \begin{cases} a, & a = b, \\ ((a+1) \operatorname{XOR} (b+1)) - 1, & a \neq b. \end{cases}$$

The operation is Latin and nonassociative. Take identity input and output dictionaries, $a = (0, 0, 0)$ and the left-associated function $F(x_1, x_2, x_3) = (x_1 \oplus x_2) \oplus x_3$. Form the stars and the observed binary law from the first-two-position slice. For $x = (0, 1, 3)$, direct evaluation gives

$$F(0, 1, 3) = 6, \qquad (s_1(0) \odot s_2(1)) \odot s_3(3) = 5.$$

Thus the complete anchored slice does not support the group's ordered extension on this task. The example identifies a substantive recovery assumption while remaining within the scope of the Latin cancellation theorem.

Other interface boundaries follow directly. At $p = 1$, an arbitrary bijection needs $q - 1$ queried values, with the last supplied by permutation completion. At $p = 2$, a full two-position slice exhausts the input domain and has no held-out tuples. Nonbijective dictionaries can prevent inversion of the anchor maps. The algorithm needs the alphabet cardinality to recognise completed closure. Passive independent samples may omit the structured neighbourhoods requested by the active construction.

H RECOVERY WITH INDEPENDENT ANSWER CORRUPTION

RQ1 concerns what selected complete answers determine. This extension allows the same complete input to be queried repeatedly when replies are independently corrupted. We analyse the archived conservative algorithm, which observes every nonanchor star entry and every nonidentity entry of each requested translation.

Theorem 4 (Independent answer corruption)**.** *Let $0 < \delta < 1$ and $\overline{N} = 1 + (p + \lfloor \log_2 q \rfloor)(q-1)$. Suppose that, conditional on any prior query history, repeated replies to the current tuple are independent and each is wrong with probability at most $\eta < 1/2$. Query each distinct tuple r times and return a label with the most votes, breaking ties by a fixed rule. If*

$$r \geq \left\lceil \frac{\log(\overline{N}/\delta)}{2(1/2 - \eta)^2} \right\rceil, \tag{17}$$

the resulting recovery is exact with probability at least $1-\delta$. An implementation capped at $\overline{N}$ distinct queried tuples uses at most $r\overline{N}$ raw oracle calls. Under independent symmetric q-class corruption, a corresponding guarantee holds for $\eta < (q-1)/q$ using the plurality bound below.

Proof. Condition on a history in which every earlier vote was correct. The next tuple then agrees with the noiseless algorithm's next query. Each reply is correct with probability at least $1-\eta$. A wrong returned label requires the correct label to receive at most $r/2$ votes. Hoeffding's inequality gives

$$\Pr[\widehat{F}(x) \neq F(x) \mid \text{earlier votes correct}] \leq \exp[-2r(1/2-\eta)^2].$$

The clean algorithm requests at most $\overline{N}$ distinct tuples. Applying a union bound to the first incorrect decision along the clean query sequence gives

$$\Pr[\text{some incorrect decision}] \leq \overline{N}\exp[-2r(1/2-\eta)^2]. \tag{18}$$

Equation 17 bounds this probability by δ. On the event of no incorrect decisions, all maps and the recovered operation equal their noiseless values, so the exact recovery proof applies. A cap of $\overline{N}$ distinct tuples guarantees the stated call budget also on inconsistent histories; reaching the cap or failing a consistency audit may return failure. If the realised clean query sequence uses $N \leq \overline{N}$ distinct tuples, its raw call count is rN.

For symmetric noise, each wrong label has probability $\eta/(q-1)$. For a specified wrong label, let a single-vote variable be 1 on the correct label, -1 on that wrong label and 0 otherwise. Its expectation is

$$\mu = (1-\eta) - \frac{\eta}{q-1} > 0,$$

and its range has length two. Hoeffding's inequality bounds the probability that this wrong label matches or exceeds the correct vote count by $\exp(-r\mu^2/2)$. Union bounds over wrong labels and the first wrong decision yield

$$\Pr[\text{some plurality error}] \leq \overline{N}(q-1)\exp(-r\mu^2/2).$$

Thus $r \geq \lceil 2\log(\overline{N}(q-1)/\delta)/\mu^2 \rceil$ suffices in the symmetric model. □

H.1 Permutation repair and the experimental protocol

For each star or translation, the experiments optionally project the vote-count matrix onto a permutation: they choose the bijection with the largest total vote score. This repair returns the true permutation whenever that permutation is the unique maximiser. A strict correct plurality in every row is sufficient, since choosing a different label in any changed row strictly reduces the total score. Some inconsistent rowwise decisions can also be repaired by the shared one-to-one constraint. Errors with large vote margins can instead favour an incorrect permutation, so empirical repair performance is reported separately from the voting guarantee.

The noise pilot uses $\mathbb{Z}_7$, D_4 and $\mathbb{Z}_3^2$, corruption rates 0.05 and 0.10, eight independent trials, and $r \in \{1,3,5\}$. A typical $\mathbb{Z}_7$ task of arity four requests 31 distinct tuples, costing 31, 93 or 155 raw calls at these repetition levels. The observed success at five repetitions is an empirical result for this protocol; the theorem selects repetitions from the specified noise rate, query bound and target failure probability.

I RQ1: Structural coverage and reconstruction results

This section records the extent of the exact-recovery audit and the conditions under which it is interpreted. Each instance is a composition system and arity with independently relabelled dictionaries. The 162 instances therefore cover 42 operation–arity settings rather than 162 different algebras.

I.1 Discovery and independently seeded confirmation

The anchor-minus-random AULC intervals are $[.031, .220]$ and $[.004, .101]$. Both runs use the same task-generation procedure. The different neural effect sizes motivated the larger locked comparison; exact recovery succeeds in both structural audits. Noise results refer to the tested independent-corruption settings and the largest tested repetition count, rather than a universal five-repeat guarantee.

Table 5: Coverage of the confirmatory reconstruction archive. Every listed instance is recovered exactly.

Group	q	Commutative?	Arities p	Relabels per p	Instances
$\mathbb{Z}_5$	5	Yes	3,4,5,7	4	16
$\mathbb{Z}_7$	7	Yes	3,4,5,7	4	16
$\mathbb{Z}_{11}$	11	Yes	3,4,5,7	4	16
$\mathbb{Z}_2^3$	8	Yes	3,4,5,7	4	16
$\mathbb{Z}_3^2$	9	Yes	3,4,5,7	4	16
D_3	6	No	3,4,5,7	4	16
D_4	8	No	3,4,5,7	4	16
D_5	10	No	3,4,5,7	4	16
Q_8	8	No	3,4,5,7	4	16
Cube rotations	24	No	3,5,7	3	9
Finite planar pose	36	No	3,5,7	3	9
All associative			42 settings		162
Noncommutative subset			22 settings		82

Table 6: The two lower-budget runs, with frozen settings and different master seeds. These precede the locked 6,000-update studies. AULC averages the predefined held-out checkpoints.

Endpoint	Discovery	Confirmation
Associative instances	162	162
Noncommutative instances	82	82
Recovered associative accuracy	1.000	1.000
Mean sparse distinct queries	66.926	66.679
Mean full-slice distinct queries	190.432	190.432
Mean per-instance query reduction	.466	.468
Nonassociative naive extension	.263	.259
Random direct-composer held-out accuracy	.131	.129
Anchor direct-composer held-out accuracy	.268	.194
Recovered-composer held-out accuracy	1.000	1.000
Anchor-minus-random held-out AULC	+.104	+.045
Delayed-entropy AULC contrast	-.0017	-.0051
Generic distillation gain	+.0031	+.0016
Projected recovery at maximum noise repeats	1.000	1.000
Unprojected recovery at maximum noise repeats	.958	.979

I.2 Paper-supplement structural and component audits

After recovery finishes, an evaluator checks all pq role-map entries and all q^2 recovered operation entries against the hidden task. This evaluator is unavailable during reconstruction. All 40 associative supplement instances pass, including 24 noncommutative instances; all three Fano controls are rejected. The 36 recovered predictors in the neural comparison are exact on their common held-out splits. Removing position-specific maps lowers mean accuracy to $.108$. On instances whose hidden anchor differs from the identity, omitting the H^{-1} correction lowers mean accuracy to $.201$.

The deterministic policy that chooses a generator outside the known subgroup uses 70.1 queries on average in this supplement, compared with 257.0 for a complete slice. A blind random column order uses 68.3 on the same census. The policy achieves the proved query upper bound, while blind random ordering uses fewer queries on average in this census. The broad 162-instance and 40-instance supplement counts refer to different censuses.

Table 7: First-seed results after 6,000 updates. Initialisations are averaged within independently relabelled tasks. Constructive recovery uses the same anchor answers and no optimisation updates.

Family	Units	Random	Answer-matched	Anchor
$\mathbb{Z}_7$	8	.138	.143	.162
$\mathbb{Z}_3^2$	8	.114	.108	.390
D_4	8	.123	.131	.328
Q_8	8	.125	.124	.352
Cube rotations	4	.037	.038	.052
All units	36	.115	.117	.280
Recovered	36		1.000	

J RQ2–RQ3: PAPER-SCALE LEARNING RESULTS

J.1 RESULTS BY TASK FAMILY

The first study's anchor-minus-matched contrast is .163, with task-level bootstrap interval $[.104, .225]$. Every family has a positive mean contrast in this study, with smaller effects for $\mathbb{Z}_7$ and cube rotations. The matched control has the same delivered label count and exact output histogram, as well as broader unary coverage; its extra screening costs appear in section C.

J.2 RESULTS UNDER AN INDEPENDENT MASTER SEED

Table 8: Seed-specific locked studies. Intervals resample the 36 task units within each study after averaging their optimiser initialisations.

Quantity	Seed 2026082619	Seed 2026090601
Random held-out accuracy	.1152	.1168
Matched-random held-out accuracy	.1166	.1181
Anchor held-out accuracy	.2796	.2803
Recovered held-out accuracy	1.0000	1.0000
Anchor − matched	.1630	.1622
95% interval	$[.1045, .2249]$	$[.0949, .2411]$
Recovered − anchor	.7204	.7197
95% interval	$[.6605, .7768]$	$[.6355, .7895]$
Family-macro anchor − matched	.1481	.1459
Leave-one-family-out range	$[.1146, .1816]$	$[.0851, .1826]$

With the first study's development-selected learning rate fixed for confirmation, the two studies provide separate measurements of the same comparison under a common protocol. They share a benchmark design, and two master seeds give limited information about variability across seeds.

In the first study, exact recovery exceeds anchor training by .720 ($[.661, .777]$) and matched-random training by .883 ($[.874, .894]$). At 6,000 updates, the anchor condition reaches training competence in 6.25% of D_4, 18.75% of Q_8 and 12.5% of $\mathbb{Z}_3^2$ trajectories, and in none of the $\mathbb{Z}_7$ or cube trajectories. These are optimiser-run proportions, distinct from the task-averaged accuracy summaries.

For the independent seed, reaggregation of all 204 final records reproduces the reported means. Anchor training is competent in $8/68$ runs, while the two random conditions are competent in $0/68$ each. The aggregate report's $8/204 = 3.9\%$ combines all three conditions. The independent cube-family anchor-minus-matched contrast is $-.00078$; the overall and leave-one-family-out effects remain positive. Its three primary intervals are reproduced exactly with the original 10,000 task-bootstrap draws and seed namespaces.

J.3 TRAINING AND HELD-OUT TRAJECTORIES

The prespecified 12,000-update subset reaches held-out accuracies .113, .120 and .197 for random, matched and anchor training. This subset contains one relabelling per family and uses a horizon-

Table 9: Task-unit means from the first locked study. Training accuracy is measured on the delivered training examples and held-out accuracy on disjoint same-arity inputs. No condition is selected at an intermediate checkpoint.

	Random		Matched		Anchor	
Updates	Train	Held-out	Train	Held-out	Train	Held-out
40	.120	.117	.114	.117	.117	.116
120	.443	.116	.454	.119	.612	.172
250	.500	.117	.497	.118	.763	.226
500	.523	.117	.513	.117	.797	.268
1,100	.532	.117	.519	.118	.807	.276
2,000	.535	.117	.521	.117	.816	.282
3,000	.532	.117	.529	.118	.813	.281
4,500	.532	.117	.520	.119	.812	.279
6,000	.521	.115	.514	.117	.807	.280

dependent temperature schedule. It checks a longer-budget configuration; its lower anchor mean cannot be interpreted as a paired deterioration caused by extra updates.

J.4 Permutation-aware discrete inference

The discrete baseline receives the true latent table and the identical anchor answers. It alternates Hungarian assignment updates for input and output permutations, with at most 80 sweeps and 128 random restarts per task. Within a task and restart budget, the retained candidate maximises training accuracy, with lexicographic dictionary ordering breaking ties. The global restart budget is selected by family-macro accuracy on the disjoint development splits, choosing the smallest budget within .002 of the best score. The selected budget is 128. The procedure uses neither recovered initialisation nor teacher labels.

Table 10: Discrete permutation inference on the 36 first-seed task units. The cube-rotation condition does not fit its anchor labels, identifying a search failure in that condition.

Family	Train	Held-out	Perfect held-out tasks
$\mathbb{Z}_7$	1.000	1.000	$8/8$
$\mathbb{Z}_3^2$	1.000	1.000	$8/8$
D_4	1.000	1.000	$8/8$
Q_8	1.000	1.000	$8/8$
Cube rotations	.759	.092	$0/4$
Relabel weighted	.973	.899	$32/36$
Family macro	—	.818	—

The result shows that permutation-aware search often finds a successful predictor from these labels. Constructive recovery succeeds on every task, including cube rotations, while using no supplied latent table or development labels. The methods' computation and prior information are therefore reported alongside their common training examples.

J.5 Identifiability and generalisation in visible affine tasks

To examine the separation between training fit and held-out success, we use affine tasks whose identifiability can be checked independently. Over $\mathbb{F}_q$, restricted training supports produce a supervision matrix of known nullity A, leaving q^A coefficient vectors consistent with the labels. The symbolic audit enumerates the entire ambiguity-defining support and measures rule agreement on the complete domain, including that support. For $(q, p) = (5, 4)$, $A = 0$ gives one rule and agreement 1, while $A = 1$ gives five rules and arbitrary-extension agreement $1/5$. For $(q, p) = (7, 5)$, the corresponding counts are one, seven and 49 for $A = 0, 1, 2$, with agreements $1, 1/7, 1/7$. Full-domain agreement can remain unchanged as nullity increases. The domain sizes are $5^4 = 625$ and $7^5 = 16{,}807$; enu-

meration here measures agreement among consistent rules rather than specifying the neural training set.

We compute the neural certificate on the actual examples from five unscrambled $q = 7, p = 5$ tasks, regenerated under master seed 20260823. Each contains 1,400 distinct class-balanced training tuples, sampled without replacement, and a genuinely disjoint held-out set. For the affine class

$$y = \beta_0 + \sum_{i=1}^{5} \beta_i x_i \pmod 7,$$

the augmented matrix $[X\ \mathbf{1}]$ has rank six in every identifiable task. Its nullity is zero, certifying a unique consistent affine rule. The generating rule is consistent, so exact affine inference predicts every held-out input correctly. The five matched ambiguous tasks have rank five and nullity one, leaving exactly seven consistent affine rules.

The affine solver and correctly structured composer reach held-out accuracy 1.000; training-competent MLPs and Transformers reach .099 and .102. The certificate records domain size, unique rows, finite-field rank, nullity and consistent-rule count for each task. It applies to visible affine functions. Independently scrambled symbols may destroy that affine form, so the scrambled group tasks use the anchor-reconstruction certificate instead.

K Uniform cancellation in Latin composition

RQ4 asks how uncertainty in multiple dictionaries enters the composer's predictions. The following proof requires the Latin property, independent categorical operands and a fixed left-associated composition. Associativity and commutativity are unnecessary. This is why the signal analysis covers a broader operation class than the recovery construction.

For a deterministic operation on the latent alphabet G, define

$$\mathcal{B}(v, w)_c = \sum_{a,b \in G} \mathbf{1}[a \oplus b = c] v_a w_b.$$

When the table is Latin, fixing a, c leaves exactly one b such that $a \oplus b = c$. The analogous column condition holds when b, c are fixed. Therefore

$$\mathcal{B}(v, \mathbf{u})_c = \frac{1}{q} \sum_a v_a \sum_b \mathbf{1}[a \oplus b = c] = \frac{1}{q} \mathbf{1}^\top v,$$

$$\mathcal{B}(\mathbf{u}, w)_c = \frac{1}{q} \mathbf{1}^\top w.$$

Equivalently,

$$\mathcal{B}(v, \mathbf{u}) = \mathbf{u}(\mathbf{1}^\top v), \qquad \mathcal{B}(\mathbf{u}, w) = \mathbf{u}(\mathbf{1}^\top w). \tag{19}$$

For any deterministic operation, every pair contributes to one output, so

$$\mathbf{1}^\top \mathcal{B}(v, w) = (\mathbf{1}^\top v)(\mathbf{1}^\top w). \tag{20}$$

In particular, combining a zero-mass vector with any vector produces a zero-mass vector. Combining a zero-mass vector with $\mathbf{u}$ produces the zero vector by eq. (19).

K.1 Proof of the cancellation identity

Let $\delta_i \in T_0$. For two operands, bilinearity gives

$$\mathcal{B}(\mathbf{u} + \delta_1, \mathbf{u} + \delta_2) = \mathbf{u} + \mathcal{B}(\delta_1, \delta_2).$$

Suppose the corresponding identity holds for p operands. Set $\Delta = \mathcal{B}_p(\delta_1, \ldots, \delta_p)$, which has zero mass by eq. (20). Then

$$\begin{aligned} \mathcal{B}_{p+1}(\mathbf{u} + \delta_1, \ldots, \mathbf{u} + \delta_{p+1}) &= \mathcal{B}(\mathbf{u} + \Delta, \mathbf{u} + \delta_{p+1}) \\ &= \mathbf{u} + \mathcal{B}(\Delta, \delta_{p+1}) \\ &= \mathbf{u} + \mathcal{B}_{p+1}(\delta_1, \ldots, \delta_{p+1}). \end{aligned}$$

Induction proves the identity in theorem 2.

Multilinearity on a finite-dimensional space supplies a constant $C_{p,q}$ such that

$$\|\mathcal{B}_p(\delta_1, \dots, \delta_p)\| \le C_{p,q} \prod_i \|\delta_i\|.$$

For a tangent perturbation $h \in T_0$ at position i, differentiation of the identity gives

$$D_i\mathcal{B}_p[h] = \mathcal{B}_p(\delta_1, \dots, \delta_{i-1}, h, \delta_{i+1}, \dots, \delta_p),$$

with norm bounded by a constant times $\|h\| \prod_{j\neq i} \|\delta_j\|$.

Converse for deterministic square tables. The two absorption identities characterise the Latin property. Set $v = e_a$ in $\mathcal{B}(v, \mathbf{u}) = \mathbf{u}$: each output must occur once in row a. Setting $w = e_b$ in the other identity gives the column condition. Thus the cancellation arises from the table's combinatorial structure.

L Uncertain decoding and parameter gradients

The neural model also learns how latent results are named, so we track output uncertainty separately from operand uncertainty and retain the input-independent output bias in the centring convention.

Write the row-stochastic decoder as $D = J + \eta E$, where $E\mathbf{1} = 0$ and $\|E\|$ is bounded. For deviations $\delta_i(x_i) \in T_0$, the latent prediction is

$$z(x) = \mathbf{u} + \Delta(x), \qquad \Delta(x) = \mathcal{B}_p(\delta_1(x_1), \dots, \delta_p(x_p)) \in T_0.$$

Since $J^\top \Delta = 0$, the decoded prediction satisfies

$$\begin{aligned} f(x) - D^\top \mathbf{u} &= D^\top \Delta(x) \\ &= \eta E^\top \mathcal{B}_p(\delta_1(x_1), \dots, \delta_p(x_p)). \end{aligned} \tag{21}$$

When $\|\delta_i(x_i)\| = O(\epsilon_i)$,

$$\|f(x) - D^\top \mathbf{u}\| = O\left(\eta \prod_i \epsilon_i\right), \qquad \left\|D_i f|_{T_0}\right\| = O\left(\eta \prod_{j\neq i} \epsilon_j\right).$$

If every scale equals ϵ, the prediction term has order at least $p+1$, and the operand derivative has order at least p. These are upper bounds on magnitudes; coefficients can vanish along special directions. The aligned construction below attains the stated orders.

L.1 An aligned path attaining the signal order

Corollary 1 (Aligned mixtures with the true dictionaries)**.** *Mix each true input dictionary with uniformity using coefficient ϵ, and use $D_\epsilon = (1-\epsilon)J + \epsilon\Pi_\phi$, where Π_ϕ is the true permutation decoder. For an example with answer $y = F(x)$,*

$$f_\epsilon(x) = \mathbf{u} + \epsilon^{p+1}(e_y - \mathbf{u}), \qquad \ell_\epsilon(x) = \log q - \log\big(1 + (q-1)\epsilon^{p+1}\big). \tag{22}$$

All nonconstant loss derivatives of orders one through p vanish at $\epsilon = 0$ along this path.

Proof. For $a, b \in G$, let $\delta_a = e_a - \mathbf{u}$. Bilinearity and eq. (19) give

$$\mathcal{B}(\delta_a, \delta_b) = e_{a\oplus b} - \mathbf{u} = \delta_{a\oplus b}.$$

Consequently, along the aligned operand mixtures,

$$\mathcal{B}_p(\mathbf{u} + \epsilon\delta_{g_1(x_1)}, \dots, \mathbf{u} + \epsilon\delta_{g_p(x_p)}) = \mathbf{u} + \epsilon^p \delta_{g_1(x_1)\oplus\cdots\oplus g_p(x_p)}.$$

The latent expression uses the fixed left association when necessary. Both J and Π_ϕ are doubly stochastic, so $D_\epsilon^\top \mathbf{u} = \mathbf{u}$ and $D_\epsilon^\top \delta_h = \epsilon\delta_{\phi(h)}$. This proves the prediction identity. Its correct-class probability is $q^{-1}[1 + (q-1)\epsilon^{p+1}]$. Taking the negative logarithm gives the loss and its Taylor order. □

Centring the measured signal. A row-stochastic decoder can contain a lower-order class bias:

$$D^\top \mathbf{u} - \mathbf{u} = \eta E^\top \mathbf{u}.$$

Because this term is independent of the example, Equation 21 centres at $D^\top \mathbf{u}$ to isolate the effect of the composed inputs. Subtracting the average prediction across inputs also removes any input-independent bias. Centring directly at $\mathbf{u}$ gives the stronger order statement only when the decoder preserves uniformity, as a doubly stochastic decoder does.

L.2 From probability perturbations to task gradients

Let a dictionary row be $r(\ell) = \mathrm{softmax}(\ell/\tau)$, with fixed temperature $\tau \geq \tau_{\min} > 0$. Its Jacobian is

$$D_\ell r = \tau^{-1}(\mathrm{diag}(r) - rr^\top),$$

whose image lies in T_0. For a target y, the cross-entropy loss $L_y(f) = -\log f_y$ has a bounded derivative wherever f_y is bounded away from zero, including a neighbourhood of uniform predictions. The chain rule therefore gives the role-logit task-gradient bound

$$\|\nabla_{\ell_i} L_y\| = O\left(\frac{\eta}{\tau} \prod_{j \neq i} \epsilon_j\right). \tag{23}$$

The factor $1/\tau$ must be retained if temperature varies with the perturbation scale. Decoder gradients and regularisation gradients can have different orders. Likewise, an adaptive optimiser applies its own rescaling to a task gradient; the actual-update diagnostic in the main paper measures that additional step.

Entropy as a local comparison. Let $E(v) = -\sum_j v_j \log v_j$ denote the entropy of a probability vector. For centred logits $\ell \in T_0$ at temperature one,

$$E(\mathrm{softmax}\,\ell) = \log q - \frac{\|\ell\|_2^2}{2q} + O(\|\ell\|_2^3).$$

Along a fixed nonzero direction $\ell = \epsilon v$, its gradient has norm $\Theta(\epsilon)$. With dictionary and decoder deviations of order ϵ, the role task-gradient norm is $O(\epsilon^p)$. For a fixed nonzero entropy-penalty weight, their norm ratio along such a direction is therefore $O(\epsilon^{p-1})$. This quantifies competing local scales. The empirical initialisation and regularisation controls in section P.3 test whether altering these scales improves the full training outcome.

M Partially resolved dictionaries

To derive the matched local degree used in RQ5, fix $p - k$ input dictionaries to their true one-hot states and write the remaining $k \geq 1$ operands as $\mathbf{u} + \epsilon\delta_i$, with $\delta_i \in T_0$. Expand the multilinear composition over those k uncertain operands.

The term that chooses $\mathbf{u}$ at every uncertain position is uniform. To see this, composition of fixed one-hot operands yields a one-hot distribution until the first uniform operand; Latin absorption then makes the running distribution uniform, and subsequent mass-one operands preserve uniformity.

Every term choosing some, but not all, centred operands vanishes. If a centred operand comes first, the running vector has zero mass until it meets a uniform operand, which sends it to zero. If a uniform operand comes first, the running distribution remains uniform through the fixed operands until a centred operand sends it to zero. Subsequent bilinear composition preserves zero. Only the term choosing all k centred operands survives, with factor ϵ^k.

For aligned mixtures with the true dictionaries, setting $\epsilon = 1$ produces the correct latent one-hot state. The surviving coefficient is therefore that state minus $\mathbf{u}$. A fixed permutation decoder preserves the order k; an aligned decoder mixed with uniformity multiplies the coefficient by another ϵ, giving order $k + 1$. This proves the order and tightness statement associated with theorem 2. If every input dictionary is fixed and only the decoder is mixed, the aligned output order is one directly.

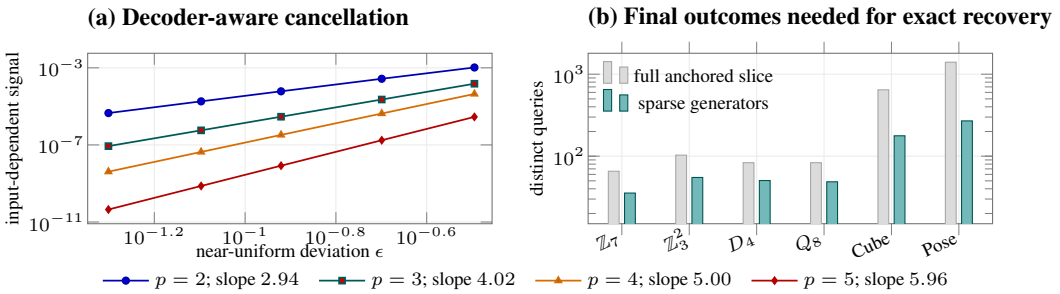

Figure 3: **Numerical signal checks and reconstruction cost.** Signal slopes follow the decoder-aware local degree. Query counts describe the conservative archived implementation, which queries the final entry of a permutation instead of completing it by elimination.

In particular, fixing the output dictionary leaves p uncertain inputs and order p, while fixing one input dictionary leaves $p-1$ uncertain inputs and an uncertain decoder, again giving order p. The comparison thus matches the number of resolved permutations and the aligned local degree. Its observed difference in learning success identifies an effect beyond that shared degree, while the training objective and the information revealed by each intervention remain as specified in the protocol.

N RQ4: NUMERICAL CHECKS OF LOCAL SIGNAL ORDER

The local theorem concerns specified probability paths near uniform dictionaries. We check both its algebraic identities and the exponents measured along those paths. These checks are separate from fitting an exponent to an optimiser trajectory.

Table 11: Independent-confirmation log–log slopes. The output decoder is learned and near uniform. The task-to-entropy ratio is a descriptive measurement, rather than evidence of an effective entropy intervention.

Arity	Predicted signal	Measured signal	Role gradient	Task/entropy
2	3	2.935	1.953	.968
3	4	4.018	3.013	2.003
4	5	5.003	3.963	2.930
5	6	5.961	4.997	4.018

The original sweep includes arities through eight and very small initialisation scales. Naively fitting all points gives numerical-slope pass rates of .504 for signals and .694 for gradients, as terms such as ϵ^p reach the floating-point floor. Its exact cancellation residual is 2.22×10^{-16}. The later precision-aware sweep uses moderate arities, centres predictions across the full input universe and measures the decoder-aware exponents directly.

The mechanism supplement adds a float64 aligned-path audit of 84 task/configuration combinations, with $\epsilon \in \{.05, .08, .12, .20, .32\}$. It verifies degree k for k uncertain roles and a known decoder, and degree $p+1$ when all roles and the decoder are uncertain. The maximum identity residual is 1.11×10^{-16}.

O RQ5: COMPONENT REVELATION AND ASSIGNMENT CONSTRAINTS

For the intervention studies, we use the first four relabellings of $\mathbb{Z}_7$ at $p = 5$ and $\mathbb{Z}_3^2, D_4, Q_8$ at $p = 4$, under master seed 2026082619, with two optimiser initialisations for each of the 16 task units. Conditions revealing input dictionaries also average two predefined role-subset choices within each task. The original mechanism study contains 352 final trajectories. All 32 soft-baseline replays match their archived held-out accuracies exactly.

All final runs use 6,000 AdamW updates, learning rate .05, batch size 96, weight decay 10^{-4}, entropy weight .002 and column-penalty weight .05. Temperature decreases linearly from 1 to .12. Evaluation uses the final temperature and soft predictions. Initial states, anchor examples and minibatch sequences are preserved across the matched conditions. The additional controls extend this same cohort; they are paired interventions rather than independent-seed replications.

O.1 Revealing the decoder and input dictionaries

The original revelation conditions fix the correct decoder, then additionally fix one, $\lfloor p/2 \rfloor$ or $p-1$ correct input maps. These interventions supply privileged information and reduce the number of free parameters. A random-half condition fixes the true decoder and random maps at half of the roles; those maps may conflict with the task, making it a constraint stress test.

Table 12: Original mechanism cohort: family-specific held-out accuracy. Every family has four task units. "One", "half" and "all but one" fix those input maps in addition to the correct decoder.

Family	Soft	Sinkhorn	Decoder	One	Half	All but one	Random half
$\mathbb{Z}_7$	.134	.167	.625	1.000	1.000	1.000	.168
$\mathbb{Z}_3^2$	.251	.528	1.000	1.000	1.000	1.000	.135
D_4	.345	.221	.901	.736	.747	1.000	.163
Q_8	.481	.312	.755	.815	.711	1.000	.139

Fixing the decoder improves accuracy by .518 (95% interval $[.392, .635]$). Relative to decoder-only revelation, fixing one input map adds .068 $[-.042, .189]$, half adds .044 $[-.092, .194]$, and all but one adds .180 $[.071, .298]$. Partial revelation is nonmonotonic: for D_4, .901 with the decoder alone becomes .736 with one additional input map; for Q_8, .815 with one supplied input becomes .711 with half supplied. All tested one-unknown-input conditions attain perfect held-out accuracy, but elsewhere a reduction in local degree can accompany lower finite-budget accuracy.

O.2 Matched input-versus-output comparison

The final supplement fixes one correct input dictionary while leaving the decoder learned. Let A_{ij} be final held-out accuracy, where $i = 1$ fixes that input dictionary and $j = 1$ fixes the output dictionary. The new A_{10} condition uses the same two role choices as the archived A_{11} condition. Baseline A_{00} and decoder-only A_{01} reuse the corresponding archived runs. Initialisations and role choices are averaged inside tasks before comparison. Both single-dictionary conditions reveal q correct entries and have the same predicted local degree, while the identities and task-relevant information of those entries differ.

Under the original objective, input-map regularisation is averaged over the roles, while the decoder contributes a separate entropy and column penalty. The comparison therefore measures the effect of fixing components with different per-map regularisation weights, with component identity and regulariser allocation varying together.

Table 13: Final controls, held-out accuracy by family. "Both" supplies one input dictionary and the decoder. "20 iter." is the original finite normalisation; "converged" is the tolerance-controlled retraining.

Family	Neither	Input only	Decoder only	Both	20 iter.	Converged
$\mathbb{Z}_7$	.134	.527	.625	1.000	.167	.122
$\mathbb{Z}_3^2$	.251	.537	1.000	1.000	.528	.532
D_4	.345	.811	.901	.736	.221	.235
Q_8	.481	.690	.755	.815	.312	.314
Mean	.303	.641	.820	.888	.307	.301

Input revelation improves accuracy by .339 $[.227, .456]$. Decoder revelation exceeds input revelation by .179 $[.072, .279]$, with a positive mean difference in all four families. The secondary interaction $A_{11} - A_{10} - A_{01} + A_{00}$ is $-.271$ $[-.456, -.082]$. We interpret the interaction only as a descriptive

measure of subadditivity on the bounded accuracy scale, where ceiling effects can contribute. The D_4 row also retains the decrease after additional revelation.

The final supplement contains 64 new input-only trajectories and 32 new converged-normalisation trajectories; 160 archived final trajectories provide the other comparison cells.

O.3 Local task gradients and actual optimiser updates

At initialisation, the soft composer's median task-only role-gradient RMS is 4.9×10^{-9} and its median first AdamW role update is .0496. With a supplied decoder, the task-gradient RMS is 3.08×10^{-7} and the first update is again .0496. Task gradients are evaluated on a fixed training minibatch. Regularisation, clipping, adaptive normalisation and weight decay also affect the actual update, allowing its magnitude to differ from the raw task-gradient scale.

Median soft-map departure from uniformity increases from .00577 at initialisation to .0471 at update 40 and .324 at the final checkpoint. The local identities and revelation experiments establish properties of dictionary uncertainty; these measurements leave the causal contribution of cancellation along the complete training trajectory unresolved.

O.4 Finite and tolerance-converged normalisation

The original continuous baseline uses 20 Sinkhorn iterations, with development accuracy after 1,100 updates selecting learning rate .05 from $\{.008, .02, .05\}$. Its held-out gain over row-softmax is .004 $[-.076, .082]$, with improvements for the two commutative families and declines for D_4, Q_8. Final maximum column-sum residuals have median .0103 and maximum .8612 over 32 runs. Post-training Hungarian projection changes aggregate held-out accuracy from .307 to .305. These residuals motivated a retrained condition with explicit numerical tolerance.

The converged operator performs 20 scaling steps followed by a damped Newton solve of the row-softmax dual. It uses internal residual tolerance 10^{-8} and float32 tolerance 10^{-6}. Its implicit derivative uses eigenvalue cutoff 10^{-12}; invalid solves are rejected. Three double-precision finite-difference checks pass on random 4×4 matrices with scales $.1, 1, 5$. A comparison with 500 differentiable scaling steps gives maximum forward difference 4.24×10^{-14} and gradient difference 2.89×10^{-14}. The maximum recorded residual across new development and final training is 2.38×10^{-7}, with no reported run errors. This verifies the continuous constraint and its derivatives in the tested implementation.

Ninety-six development runs compare $\{.008, .02, .05\}$ for 1,100 updates using only development accuracy. The selected rate is .05; final runs restart from update zero with the shared 6,000-update schedule. Converged normalisation attains .301, compared with .307 for 20 iterations and .303 for row-softmax. Paired differences are $-.006$ $[-.029, .009]$ and $-.002$ $[-.079, .073]$, respectively. Final projection also gives .301. Accurate normalisation therefore supplies no aggregate improvement in this protocol, with heterogeneous family effects and intervals that do not establish equivalence at an arbitrarily narrow margin.

O.5 Inference, runtime and reproducibility of the controls

Paired intervals use 4,000 bootstrap draws, resampling four tasks within each of the four fixed families after averaging initialisations and role choices. The original mechanism study's five intervals and the final controls' five distinct intervals are reproduced from raw comparison rows with their archived seed namespaces. The original mechanism run takes approximately 20,299 seconds (5.64 hours), including development and diagnostics; the final controls take 6.79 training hours. Source, environment, task audits, predictions, trajectories, selection records, baseline replay and CPU audit scripts are archived. The final-control material is under `evidence/final_controls`; its summaries and intervals can be recomputed without training models.

P RQ5: intermediate-supervision factorial

This separate study asks whether additional labels help because they expose correct component relationships. It uses the seven operation–arity settings and optimiser grid specified in section C.

P.1 TARGETS, LABEL BUDGETS AND TRAINING SCHEDULE

Every label uses the task's sampled output dictionary. A complete target combines all p inputs in their original order. A process target combines a named contiguous window, with the same hidden input maps and operation. Binary windows are $(1,2),\dots,(p-1,p)$; ternary windows are $(1,2,3),\dots,(p-2,p-1,p)$; unary windows contain one role. Process examples therefore supply intermediate labels beyond the final-answer interface used for reconstruction.

The low-outcome condition uses 25 class-balanced complete examples per output class; the binary, ternary and unary conditions add an auxiliary budget of $(p-1)q^2$, divided as evenly as possible among windows while retaining class balance. Shuffling targets within each binary window preserves its answer histogram while breaking semantic consistency. The matched-outcome condition uses the same total number of complete labels as the low set plus the auxiliary budget; the dense condition has 200 complete examples per class.

With one supervised minibatch per update, training with auxiliary groups follows a three-stage schedule: the first 30% of updates use auxiliary batches, the next 40% choose auxiliary or complete batches with equal probability, and the final 30% use complete batches. All batches minimise negative log likelihood plus the same regulariser. Temperature decreases linearly from 1 to .12, or from .22 in the low-temperature control. Gradient norm is clipped at five, and AdamW weight decay is 10^{-4}. Early stopping requires three consecutive final-phase evaluations with complete-task training accuracy at least .999. The archived scripts fix budget rounding, minibatching and random-number namespaces.

P.2 TRAINING COMPETENCE BY TASK AND SUPERVISION

Table 14: Training-competence rates in the frozen factorial. Each entry aggregates ten optimiser seeds and three learning rates. "Matched" uses additional complete outcomes; "shuffled" permutes binary targets within windows.

Task	Direct	Matched	Binary	Ternary	Shuffled	Dense
$\mathbb{Z}_7$, $p=3$	1.000	.967	.967	.800	.000	.967
$\mathbb{Z}_7$, $p=5$	.000	.233	.900	.800	.000	.767
$\mathbb{Z}_2^2$, $p=4$	1.000	1.000	1.000	1.000	.100	1.000
$\mathbb{Z}_3^2$, $p=4$	.167	.267	1.000	.567	.000	.967
D_3, $p=4$	.600	.600	.333	.400	.000	.700
D_4, $p=4$	.133	.133	.200	.167	.000	.267
Quasigroup, $p=4$	.000	.000	.167	.033	.000	.000

Table 15: Task-clustered competence contrasts from the factorial. The intervals retain variation across the seven task settings.

Contrast	Mean	95% interval
Correct binary − shuffled binary	+.638	$[+.328, +.870]$
Binary − ternary in the original design	+.114	$[+.006, +.239]$
Binary − matched complete outcomes	+.195	$[-.050, +.422]$
Binary − low-budget outcomes	+.238	$[-.044, +.493]$
Dense − low-budget outcomes	+.252	$[+.039, +.467]$
Large initialisation − low-budget outcomes	-.129	$[-.176, -.076]$

Correct intermediate semantics substantially improve competence relative to shuffled targets. The binary–ternary contrast describes this particular construction and varies under coverage and scheduling controls. The interval for binary versus matched complete outcomes includes zero, and D_3 reverses the average direction. These measurements support the role of informative auxiliary semantics, with task-dependent effects rather than a universal advantage for lower-order supervision.

Table 16: Lower-budget independent-confirmation controls. These use the pilot horizon rather than the 6,000-update locked study.

Method	Train	Held-out	Held-out AULC
Random direct composer	.570	.129	.133
Anchor direct composer	.822	.194	.178
Delayed entropy	.521	.127	.128
Arity-balanced penalty	.537	.126	.125
No entropy penalty	.539	.123	.128
No regulariser	.584	.130	.128
Large initialisation	.545	.122	.127
Late temperature cooling	.562	.133	.133

P.3 Additional regularisation and scheduling controls

Delayed entropy has paired AULC contrast $-.0051$, interval $[-.0136, .0026]$, and paired sign-flip $p = .301$. This intervention gives no evidence that postponing confidence-inducing regularisation resolves the observed training difficulty.

P.4 Generic students trained from the recovered predictor

Table 17: Student controls. Teacher-generated examples use no additional true oracle calls, but increase the number of training inputs.

Student	Training	True labels	Pseudo-labels	Held-out
MLP	Anchor replay	43	0	.117
MLP	Teacher examples	43	1,800	.143
Transformer	Anchor replay	43	0	.128
Transformer	Teacher examples	43	1,800	.106
Recovered composer	Exact representation	43	0	1.000

The aggregate student gain is $.00165$, with interval $[-.0292, .0383]$. The recovered predictor is exact, while this distillation protocol supplies no reliable aggregate improvement to the tested generic students.

Q A supplementary relational-composition check

The frozen paper-profile experiment also includes CLUTRR (Sinha et al., 2019). The extracted data contain 15,083 training stories, 290 length-five/six evaluation stories and 20 relation symbols, with zero exact relation-sequence overlap between training and evaluation. The extraction maps a sequence of relations to the final relation label; this mapping has a conflict rate of $.0117$.

Correct binary process labels give $.995$ held-out accuracy, budget-matched complete outcomes give $.989$, and shuffled binary labels give $.145$. The result supplies a relational check that meaningful auxiliary labels matter. Both correct-label conditions are near ceiling. CLUTRR also lacks the active complete-answer interface required by our reconstruction algorithm, so it is treated as supplementary supervision evidence rather than a test of the exact-recovery guarantee.

R Research questions, evidence and scope

The main paper follows five questions. The table below records which evidence answers each question and the conditions needed to interpret it. In particular, structural instances, independently relabelled neural tasks and optimiser trajectories are separate units of evidence.

Table 18: Evidence supporting the five research questions. The scope column states the setting in which each conclusion is used.

Question and result	Evidence	Scope
RQ1: full-function recovery	Ordered observable factorisation, constructive query bound, and 162/162 exact associative instances	Finite associative groups, bijective dictionaries and adaptive complete-answer access
RQ1: radius and label cost	Worst-case radius-one indistinguishability, radius-two recovery, version-space audits, and counting lower bound $\lceil p \log_q((q-1)!) \rceil$	Radius lower bound concerns classes with distinct laws sharing an identity; upper cost is $O(pq + q \log q)$
RQ1: answer-noise robustness	Majority-vote guarantee, repeated independent queries and empirical permutation repair	Independent corruption and explicitly counted repeated oracle calls
RQ2: informative example selection	Two locked studies compare random, answer-matched and reference-centred datasets	Same neural learner and delivered label budgets; screening costs recorded separately
RQ3: different learning outcomes	Soft composer, discrete inference and reconstruction on identical structured answers; training-fit and affine controls	Specified task classes, information access and finite optimisation/search budgets
RQ4: Latin cancellation	Algebraic identity, aligned-path loss expression, residual 2.22×10^{-16}	The categorical composer and its specified composition tree
RQ4: output-sensitive degree	Centred-output proof and reproduced slopes for arities two through five; uncertain-dictionary paths	Local input-dependent predictions after removing uniform-input decoder bias
RQ5: component identity	Decoder-only minus input-only contrast .179 $[.072, .279]$	Paired 16-task cohort, equal map count and local degree, with inherited component-specific regularisation
RQ5: continuous constraints	Held-out accuracy .301, maximum residual 2.38×10^{-7} and checked implicit derivatives	Tested continuous normalisation and optimiser, with family heterogeneity
RQ5: intermediate semantics	Binary-minus-shuffled competence contrast .638, task-clustered interval $[.328, .870]$	Specified auxiliary windows, schedules and operation–arity settings

R.1 Connections between the mathematical and learning results

The recovery theorem certifies the structured training set within $\mathcal{H}_{q,p}$; the model definition establishes expressibility. RQ3 uses both facts to compare learning systems. RQ2 instead tests example selection, without requiring that the random sets have the same certificate.

The local-signal theorem motivates the matched-degree interventions. Their different outcomes and the actual-update measurements establish that local degree is an incomplete account of training. Constructive recovery supplies an exact predictor; its selected examples also improve the separately trained neural model.

R.2 Relation to prior reconstruction and representation learning

Our full-task construction uses the binary generator step and conservative count of Bedford et al. (2003) together with the lower-arity retracts established by Krotov & Potapov (2011). Learning group representations from partial tables (Huh, 2025) and the role of context coverage (Chang et al., 2026) are complementary directions. Here, the ordered factorisation connects complete p-input examples to reconstruction, and the learning comparisons measure how this information is used.

S TRAINING TRAJECTORIES AND NORMALISATION SUMMARY

The following displays complement the main comparisons: training curves separate fitting from held-out performance, and the normalisation table reports fit and constraint residuals under both implementations.

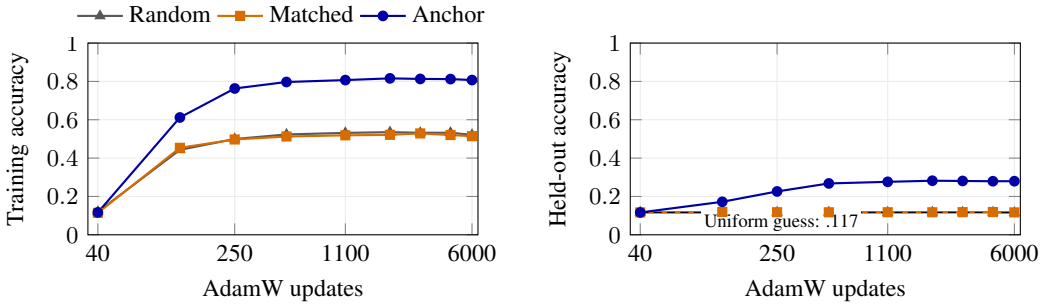

Figure 4: **Training fit and generalisation under the shared budget (RQ2–RQ3).** First-study task-mean accuracies over 6,000 updates, averaging initialisations within tasks. The task-weighted uniform-prediction reference is .1166.

Table 19: **Continuous assignment constraints on the paired subset (RQ5).** Column residuals are final-checkpoint maxima for finite iterations and the worst recorded value for the converged implementation.

Parameterisation	Train	Held-out	Column residual
Row-softmax	.840	.303	Unconstrained
20-iteration Sinkhorn	.833	.307	.861
Tolerance-converged Sinkhorn	.831	.301	2.38×10^{-7}