WHEN ATTENTION FAILS ON FINITE FIELDS: GALOIS ALGEBRAIC REASONING NETWORKS WITH HARD COSET EXECUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Exact algebraic relations need not align with similarity in a learned embedding space. We study this mismatch through a controlled coset-classification task over AES $\mathrm{GF}(2^8)$: labels depend on multiplicative ratios, while both classes have identical single-byte marginals. We introduce GARN, a Galois Algebraic Reasoning Network that separates learned interaction weights from exact field execution. Attention proposes interactions, fixed inverse and multiplication tables compute ratios, and learnable membership masks provide evidence for prediction and state updates. Across five seeds, GARN reaches 99.5% test accuracy with 256 labeled examples, whereas the tested Transformers, MLPs, and discrete relaxations remain near chance through 1,024 examples. With shuffled partners, GARN reaches 79.9%, compared with 67.0% for a fixed-adjacency probe. Template-free masks, alternative byte encodings, and corrupted-operator controls isolate the role of exact execution. These results identify operator access as a decisive source of sample efficiency and establish a practical interface between continuous interaction learning and discrete finite-field computation.

1 INTRODUCTION

Learning a relation and executing it are different problems. A neural network may have enough capacity to represent an algebraic operation yet require substantial supervision to discover its exact structure. This distinction matters when labels are scarce and small arithmetic errors change the answer. Inductive biases have made analogous problems tractable in other domains: convolution exploits translation, graph networks exploit permutation symmetry, and attention learns context-dependent interactions (Cohen & Welling, 2016; Bronstein et al., 2021; Vaswani et al., 2017; Battaglia et al., 2018). Neural algorithmic reasoners make the complementary point that a model can learn a procedure over structured states rather than only a static predictor (Veličković & Blundell, 2021; Ibarz et al., 2022; Bounsi et al., 2024). For finite-field reasoning, the corresponding question is which parts of an algebraic computation should be learned from examples and which should be supplied as executable structure.

Finite fields provide a useful setting in which to answer this question. They underlie error-correcting codes and cryptographic primitives (Lidl & Niederreiter, 1997; Daemen & Rijmen, 2002), and they are also a standard setting for finite-field constraint solving (Hader et al., 2023). Their arithmetic is not ordered by the numerical values of byte encodings. Two nearby byte values need not be algebraically related, and a small change in an embedding can alter an exact membership decision. Consider the ratio $r = a^{-1}b$ and a multiplicative subgroup H: deciding whether $r \in H$ is equivalent to deciding whether a and b occupy the same multiplicative coset. This simple predicate couples an exact operation with a learned decision rule, without the many sources of leakage present in a full cipher. It therefore complements neural cryptanalysis, whose distinguishers typically exploit distributional biases in a concrete primitive (Gohr, 2019; Gérault et al., 2024; Bellini et al., 2023).

We use that predicate to construct a controlled learning diagnostic (Figure 1). Each example contains several generating pairs. All their ratios lie in H for positive examples and outside H for negative examples. Sampling the first element uniformly and multiplying it by the selected ratio makes each byte marginal uniform in both classes. The construction removes class-specific single-

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

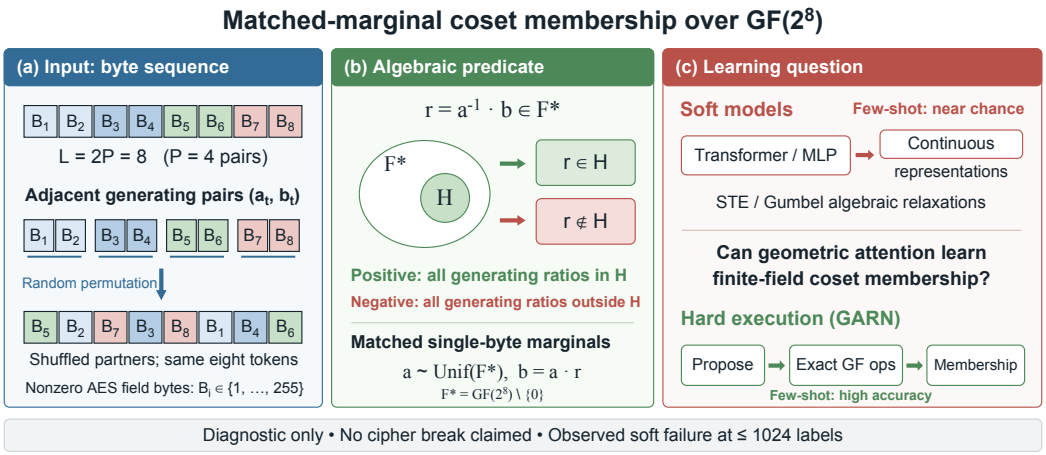


Figure 1: Coset-classification diagnostic. Left: adjacent generating pairs and the same tokens after shuffling. Middle: labels specify whether generating-pair ratios lie in H; single-byte marginals match across classes. Right: the tested soft models remain near chance at few-shot budgets, while exact field execution improves sample efficiency. Matching marginals removes first-order frequency bias, not all sequence-level statistics.

byte frequency bias while preserving the relational signal. This follows the broader use of relational inductive biases to control which dependencies a learner can exploit (Battaglia et al., 2018). An adjacent version exposes the generating pairs; a shuffled version removes that positional cue. Together, these settings distinguish access to the right arithmetic from the ability to exploit useful interactions.

We make the central design choice explicit: GARN separates *interaction learning* from *arithmetic execution*. The resulting Galois Algebraic Reasoning Network uses a Propose–Execute–Fold architecture. Multi-head attention assigns weights to candidate partners. Fixed AES $\mathrm{GF}(2^8)$ inverse and multiplication tables then compute pairwise ratios exactly. Learnable membership masks convert these discrete ratios into continuous evidence, which a gated update writes into the token states. An optional Fold broadcasts membership-weighted context. The tables require no gradient: supervision reaches the interaction weights and membership masks through the weighted evidence. This design is related to neural arithmetic units and symbolic modules, but keeps the field operation itself fixed and exact (Trask et al., 2018; Madsen & Johansen, 2020; Manhaeve et al., 2018; Yi et al., 2018).

This separation yields its clearest measured effect in the regime where operator discovery is most costly. On adjacent data, GARN with no global Fold reaches 99.5% test accuracy from 256 labels; the tested soft architectures remain near 50% through 1,024 labels. The gain persists without subgroup templates: free-mask GARN reaches 97.1% at 256 labels and 100.0% at 1,024. On shuffled data, learned interaction weights raise accuracy to 79.9%, compared with 67.0% for a fixed-adjacency hard probe. The learned weighting plays a role analogous to pointer-style routing over a changing set of candidates (Vinyals et al., 2015; Graves et al., 2014; Csordás et al., 2022). Exact-ratio feature models also improve sharply, locating the primary benefit in algebraic access rather than attention alone.

The controls narrow the interpretation of this effect. Bitwise and Fourier-log byte encodings do not close the few-shot gap, whereas replacing the exact tables by random tables reduces hard-model accuracy to chance. Soft Transformers improve at a much larger label budget, consistent with a difference in sample efficiency rather than a representational impossibility (Power et al., 2022; Barak et al., 2022; Lee et al., 2024; McLeish et al., 2024; Saxton et al., 2019; Zhou et al., 2024). Shuffled classification can also exploit aggregate algebraic evidence, so we measure partner recovery and compare all-pair aggregators separately. These analyses distinguish the benefit of exact execution from stronger claims about recovering the hidden generating pairs.

Our contributions are threefold.

1. **An executable algebraic inductive bias.** GARN connects differentiable interaction weights and membership learning to exact, non-differentiable finite-field operators, with an explicit state-update interface.
2. **A controlled relational diagnostic.** Matched single-byte marginals and adjacent/shuffled encodings separate frequency bias, arithmetic access, and the use of non-adjacent evidence.
3. **An empirical account of the sample-efficiency gap.** Comparisons across soft encoders, discrete relaxations, exact-ratio features, template priors, and operator corruption identify the conditions under which hard field execution is effective.

Taken together, these contributions provide both a model for using known algebraic operators and an evaluation protocol for isolating what those operators contribute.

2 RELATED WORK

Learning arithmetic and algebraic structure. Transformers can learn arithmetic from sufficient data, appropriate representations, and training curricula (Power et al., 2022; Lee et al., 2024; McLeish et al., 2024; Feng et al., 2023). Fourier and clock-like mechanisms explain aspects of learned modular computation (Nanda et al., 2023; Zhong et al., 2023; Gromov, 2023). Our experiments address the complementary question of sample efficiency: how much supervision is needed when the task depends on exact finite-field ratios? Alternative encodings and larger-budget controls connect this question to the existing arithmetic-learning literature.

Neural control with structured execution. Neural algorithmic reasoning and module networks combine learned components with explicit computation (Veličković & Blundell, 2021; Andreas et al., 2016; Shindo et al., 2021). GARN specializes this separation to finite fields: learned weights combine evidence produced by exact table lookups. Unlike STE and Gumbel relaxations (Bengio et al., 2013; Jang et al., 2017; Maddison et al., 2017), it does not learn the field operations through a surrogate gradient. Unlike a fixed algebraic probe, it learns which available interactions contribute to each token's state. We compare these alternatives with and without subgroup templates.

Neural cryptanalysis. Neural distinguishers typically exploit statistical biases in reduced-round ciphers (Gohr, 2019). We instead isolate a specified finite-field relation and control its single-byte marginals. This provides an architectural diagnostic for algebraic learning rather than a cipher-security evaluation. Appendix B gives additional context on neural cryptanalysis, discrete representations, and finite-field solvers.

3 PROBLEM SETUP

3.1 FINITE-FIELD NOTATION

Let $\mathbb{F} = \mathrm{GF}(2^8)$ with AES polynomial arithmetic and primitive element 0×03 for exp/log tables. Write $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$. For $a, b \in \mathbb{F}^*$, the multiplicative ratio is $r = a^{-1} \cdot b$. A cyclic subgroup $H \leq \mathbb{F}^*$ of order $d \mid 255$ is $H = \langle \alpha^{255/d} \rangle$.

3.2 MATCHED-MARGINAL COSET CLASSIFICATION

Each example is a length-$2P$ byte sequence encoding P pairs (a_t, b_t).

- **Positive:** every ratio $a_t^{-1} \cdot b_t$ lies in a fixed target subgroup H.
- **Negative:** every ratio lies in $\mathbb{F}^* \setminus H$.

Both classes sample a_t uniformly in $\mathbb{F}^*$ and set $b_t = a_t \cdot r_t$, so single-byte marginals match up to sampling noise. Consequently, an individual byte carries no label information, and expected byte frequencies agree across classes. Higher-order dependencies remain informative; matching single-byte marginals does not make the full sequence histogram label-independent (Appendix C.1).

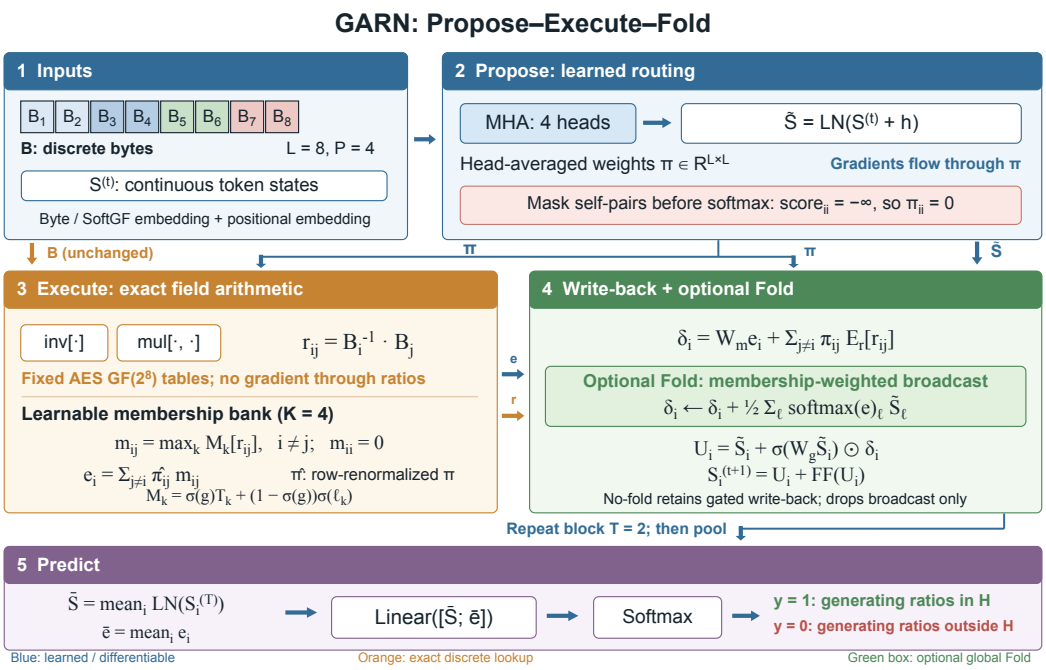


Figure 2: GARN architecture (one block; depth $T{=}2$). **Propose:** multi-head attention produces head-averaged partner weights π, with self-pairs masked. **Execute:** non-differentiable AES $\mathrm{GF}(2^8)$ inverse/multiply tables compute ratios $r_{ij} = B_i^{-1} \cdot B_j$; a membership bank yields m_{ij}, and $e_i = \sum_{j \neq i} \hat{\pi}_{ij} m_{ij}$ with row-renormalized weights $\hat{\pi}$. **Fold (optional):** write membership/ratio evidence into the post-attention state $\tilde{S}$; the no-fold variant retains gated write-back but omits the global broadcast. **Predict:** mean-pool the final layer-normalized states and membership evidence, then classify with a linear head.

Shortcut control. We construct a *shortcut* variant in which positives concentrate 75% of tokens on a single coset support (size $|H|$). This variant introduces class-dependent byte support and serves as a control for frequency bias.

Shuffled partners. We randomly permute token positions of a matched sequence. The generating (a, b) partners are then no longer adjacent, so a fixed-adjacency probe is incomplete and models must use non-adjacent algebraic evidence. This setting tests the use of non-adjacent evidence. Sequence classification and generating-pair recovery are evaluated separately.

Generator (matched). Given order $d \mid 255$, pair count P, and label y, set $H = \langle \alpha^{255/d} \rangle$ and $\overline{H} = \mathbb{F}^* \setminus H$; for each $t \leq P$ sample $r_t \sim \mathrm{Unif}(H)$ if $y{=}1$ else $\mathrm{Unif}(\overline{H})$, sample $a_t \sim \mathrm{Unif}(\mathbb{F}^*)$, and emit $(a_t, a_t \cdot r_t)$. Shuffled variants apply a uniform random permutation to the length-$2P$ token list. The target H is fixed across seeds for a given order; seed variance arises from data draws, split permutations, and model initialization. App. A reports train/test overlap statistics for bytes, ratios, pairs, and full sequences.

4 Method: Galois Algebraic Reasoning Network

GARN maintains a continuous state $S \in \mathbb{R}^{L \times d}$ and a discrete byte view $B \in \{0, \ldots, 255\}^L$. Each block performs Propose $\rightarrow$ Execute $\rightarrow$ Fold/Write (Figure 2).

4.1 Propose

A multi-head attention module contextualizes S. With $H_{\text{heads}}{=}4$, we take the *head-averaged* attention matrix returned by PyTorch MultiheadAttention (`average_attn_weights=True`) as

partner weights $\pi \in \mathbb{R}^{L\times L}$:

$$h,\ \pi\ =\ \mathrm{MHA}_{\mathrm{avg}}(S;\ A_{ii}{:=}{-}\infty), \qquad \tilde{S} = \mathrm{LN}(S+h). \tag{1}$$

Here A denotes the attention scores before softmax. Self-pairs are masked before softmax: $B_i^{-1}B_i{=}1 \in H$ for every subgroup. Gradients flow through π. The returned weights determine how strongly each candidate interaction contributes to the evidence.

4.2 Execute (hard)

For the nonzero task bytes we compute

$$r_{ij} = B_i^{-1} \cdot B_j \in \mathbb{F}^*, \tag{2}$$

with the convention $0^{-1} := 0$ (task generators sample from $\mathbb{F}^*$). No gradient passes through the multiplication/inversion tables. A bank of $K{=}4$ masks $M_k \in [0,1]^{256}$ (default; $K{=}2$ in omit-order ablations) yields

$$m_{ij} = \max_{k\le K} M_k[r_{ij}] \cdot \mathbf{1}[i \neq j]. \tag{3}$$

Template prior (default). Each mask is $M_k = \sigma(g)\, T_k + (1-\sigma(g))\, \sigma(\ell_k)$, where T_k is the indicator of a cyclic subgroup of order in $\{15, 17, 51\}$, $\ell_k \in \mathbb{R}^{256}$ are free logits initialized at 0, and g is a learned gate initialized so $\sigma(g) \approx 0.88$ (favoring templates). For the principal task $|H|{=}17$, the order-17 template is included and accelerates few-shot learning. We also report *template-free* free masks that must discover membership without this prior (Table 3). After zeroing the diagonal of π, Execute uses row-renormalized weights $\hat{\pi}_{ij} = \pi_{ij} / \max(\sum_{j'\neq i} \pi_{ij'}, \varepsilon)$, then $e_i = \sum_{j\neq i} \hat{\pi}_{ij}\, m_{ij}$. Write-back uses the proposal weights π returned by MHA, as in the implementation. The maximum in equation 3 selects the largest mask response for each ratio; gradients update the selected response. Masks have no explicit diversity regularizer.

4.3 Fold / write-back

Let $\tilde{S}$ denote the post-attention, layer-normalized state in block t. One block maps $(S^{(t)}, B)$ to $S^{(t+1)}$ by

$$\delta_i = W_m e_i + \sum_{j\neq i} \pi_{ij}\, E_r[r_{ij}], \tag{4}$$

$$\delta_i \leftarrow \delta_i + \tfrac{1}{2}\Big(\sum_{\ell=1}^{L} \mathrm{softmax}(e)_\ell\, \tilde{S}_\ell\Big) \quad \text{(Fold on only)}, \tag{5}$$

$$S_i^{(t+1)} = \tilde{S}_i + \sigma(W_g \tilde{S}_i) \odot \delta_i; \quad S_i^{(t+1)} \leftarrow S_i^{(t+1)} + \mathrm{FF}(S_i^{(t+1)}), \tag{6}$$

where Fold normalizes over positions ℓ, E_r is a learned embedding of ratio tokens, and FF is a position-wise MLP (applied after the gated residual). Thus e influences later proposals whenever $T{>}1$. The no-fold variant retains the gated update from membership and attention-weighted ratio embeddings, and omits only the global broadcast. On short adjacent tasks this variant is typically stronger.

4.4 Prediction head

After T blocks, let $\bar{S} = \frac{1}{L}\sum_i \mathrm{LN}(S_i^{(T)})$ and $\bar{e} = \frac{1}{L}\sum_i e_i$. The classifier is

$$\hat{y} = \mathrm{softmax}\big(W[\bar{S}; \bar{e}] + b\big) \in \Delta^2, \tag{7}$$

trained with standard cross-entropy on the binary sequence label. Mean pooling matches our homogeneous all-in / all-out labeling. App. E studies min-aggregation and *mixed-fraction* labels where only a subset of pairs lie in H.

Baselines. HardProbe evaluates fixed adjacent pairs with the same hard ratio and a learned mask, and provides an operator reference when partners remain adjacent. RatioFeatTransformer takes bytes together with all off-diagonal exact ratios, giving algebraic access without Propose–Execute routing. Soft baselines replace hard execution by continuous encoders: SoftCosetNet / Transformer (plain or SoftGF embeddings), MLP, STE-Algebraic (Bengio et al., 2013), and Gumbel-Algebraic with temperature annealing (Jang et al., 2017; Maddison et al., 2017). **SoftGF** concatenates a learned byte embedding with fixed algebraic *features* of each token (discrete log and multiplicative inverse as real-valued coordinates). It does *not* grant exact multiply/inverse table lookups or membership labels: soft SoftGF models still lack the hard Execute operator used by HardProbe / RatioFeatTF / GARN. Bitwise and Fourier-log encoders, together with further ablations, are described in Appendix E.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Protocol. The primary task is binary coset classification over AES $\mathrm{GF}(2^8)$ under matched-marginal sampling (Sec. 3): length $L{=}2P$ byte sequences with $P{=}4$ pairs and target subgroup order $|H|{=}17$. We treat matched data as the main evaluation target because labels depend on ratios rather than byte supports; a shortcut generator that concentrates positives on a small support is used only as a hygiene control. A shuffled-partner variant randomly permutes tokens so fixed adjacency no longer reveals generating pairs.

Models. Hard-execution models share exact multiply/inverse tables: HardProbe (fixed adjacent pairs), RatioFeatTransformer (all off-diagonal ratios as tokens), and GARN with optional Fold. Soft baselines replace hard tables by continuous encoders: SoftCosetNet, Transformer, MLP, STE-Algebraic, and Gumbel-Algebraic. Unless noted, networks use width 64, depth 2, and four attention heads. SoftGF embeddings (learned lookup plus discrete-log / inverse *features*, without table execution) are the default input representation for hard models and are optionally supplied to soft Transformers as a structure-aware soft control; App. E shows they do not close the few-shot gap without hard Execute. Further soft encoders and ablations appear in Appendix E.

Training and evaluation. We train with AdamW (learning rate 10^{-3} for sequence models, 10^{-2} for probes; weight decay 10^{-4}), batch size $\min(64, n)$, 25 epochs, and dropout 0.1. We clip the global gradient norm to (1.0) immediately before each AdamW update. A shared pool of $12{,}000$ examples is split into train/val/test of sizes n/1000/1500 under a seed-dependent permutation, so every method sees identical splits. Checkpoints are selected by validation accuracy; we report the corresponding **test** metrics.

Replication. All primary tables average five seeds and report mean$\pm$std. On matched data a rule oracle attains 100%. Appendix A records overlap statistics and the density-matched shuffled generator used only in appendix analyses.

5.2 MAIN RESULTS: MATCHED-MARGINAL SAMPLE EFFICIENCY

Figure 3 and Table 1 show a large separation between soft encoders and models with exact field access. All five soft baselines remain near 50% through $n = 1024$. In contrast, no-fold GARN improves from 86.7% with 32 labels to 99.5% with 256 labels. Its error at 256 labels is 0.5%, compared with 48.7% for the Transformer under the same split and training protocol.

HardProbe reaches 100% at every budget with subgroup templates, providing a reference for known adjacent partners and supplied algebraic structure. GARN approaches this reference while learning its interaction weights. RatioFeatTransformer also benefits from exact ratios, reaching 96.4% at 256 labels. The shared pattern across these architectures identifies exact algebraic access as the central source of sample efficiency. Larger-budget soft results, reported in Appendix E.10, clarify how this gap changes with supervision.

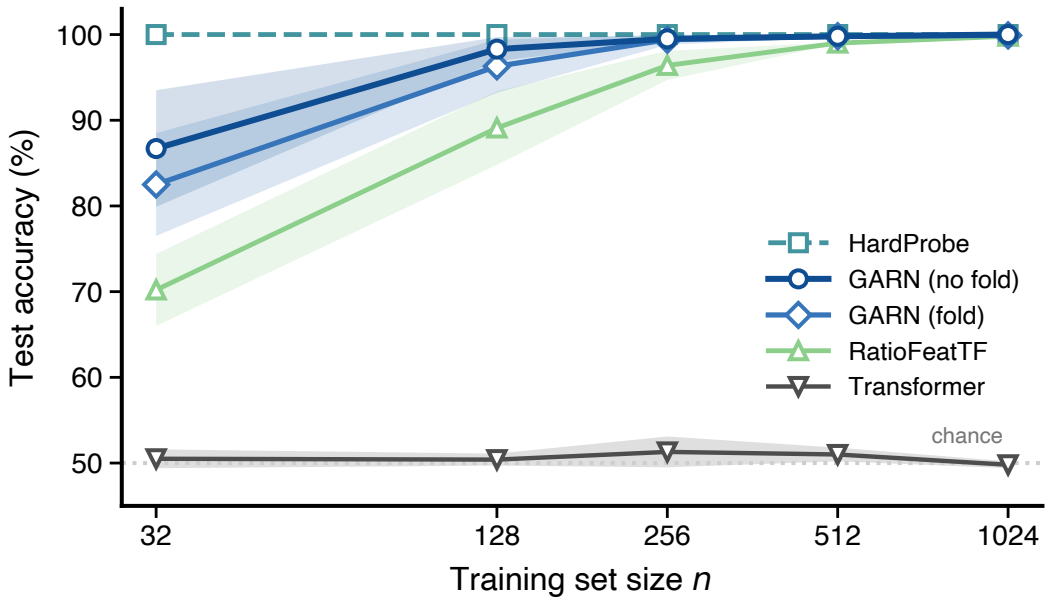


Figure 3: Sample efficiency on the matched-marginal coset task (5 seeds; **test** accuracy after val selection). Hard execution restores few-shot accuracy, while soft models stay near chance under the tested protocol.

Table 1: Matched-marginal coset test accuracy (%; mean±std, 5 seeds). Hard execution is sample-efficient; soft baselines remain near chance under $n \leq 1024$. Best in each column in **bold**.

	Training set size n				
Method	**32**	**128**	**256**	**512**	**1024**
Hard field execution					
HardProbe	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0
GARN (no fold)	86.7±6.8	98.3±1.4	99.5±0.3	99.8±0.2	**100.0**±0.1
GARN (fold)	82.5±6.0	96.3±3.1	99.4±0.6	99.8±0.1	99.9±0.1
RatioFeatTF	70.2±4.2	89.1±4.3	96.4±1.7	99.0±0.1	99.8±0.1
Soft baselines					
SoftCosetNet	49.7±0.6	49.2±0.5	50.1±1.0	49.7±1.3	50.2±0.4
STE-Algebraic	49.8±0.7	49.1±1.0	49.9±1.0	50.5±0.7	49.9±0.8
Gumbel-Algebraic	50.1±0.4	49.6±0.3	50.2±1.1	50.3±0.8	50.2±0.6
Transformer	50.5±1.1	50.4±0.7	51.3±1.8	51.0±0.8	49.8±0.4
MLP	50.1±0.4	50.0±1.3	49.8±0.2	49.6±0.9	49.8±0.7

5.3 SHUFFLED PARTNERS

Shuffling preserves each example's tokens and label but removes the alignment between generating partners and adjacent positions. Fixed-adjacency HardProbe reaches only 67.0% at $n = 1024$ (Figure 4; Table 2). No-fold GARN reaches 79.9%, a gain of 12.9 percentage points, while the soft baselines remain near chance. The comparison shows the value of using non-adjacent algebraic evidence when the original pairing is unavailable.

The sequence objective can exploit more than the generating pairs. Complete all-pair membership aggregators reach 96.9% on the ordinary shuffled data, and density-matched controls bring both these aggregators and GARN close to chance (Appendix E.6). Top-1 attention partner recovery is also near chance despite useful sequence accuracy (Appendix E.9). Thus the observed gain measures

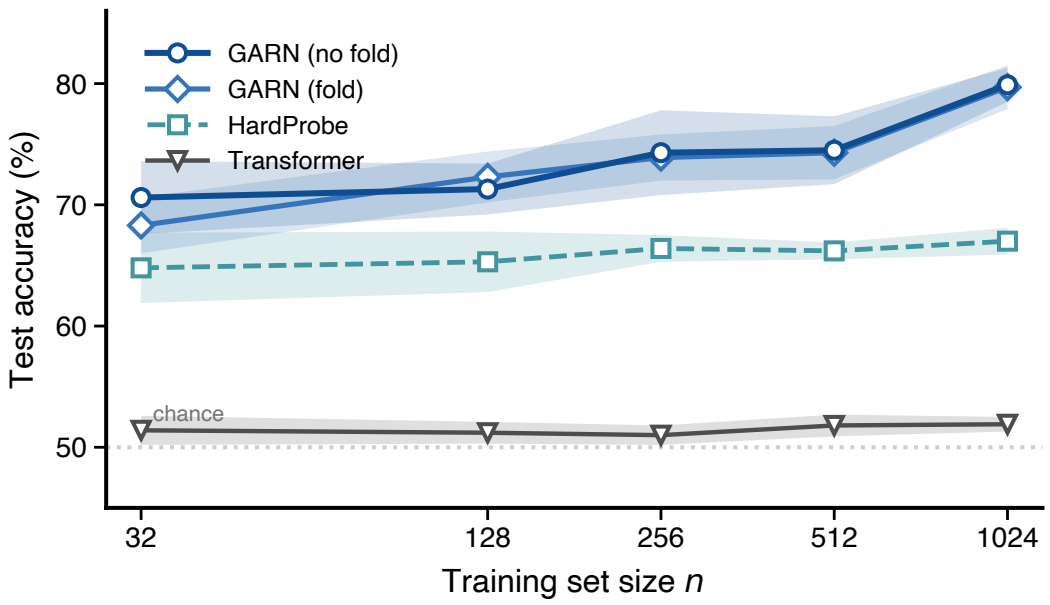


Figure 4: Shuffled partners (5 seeds; **test** after val selection). Propose–Execute outperforms fixed-adjacency HardProbe, while soft models stay near chance.

Table 2: Shuffled-partner test accuracy (%; mean±std, 5 seeds). Learned proposal improves over fixed-adjacency HardProbe; soft models stay near chance. Best in each column in **bold**.

	Training set size n				
Method	**32**	**128**	**256**	**512**	**1024**
Hard field execution					
GARN (no fold)	**70.6**±3.0	71.3±2.1	**74.3**±3.5	**74.5**±2.8	**79.9**±1.4
GARN (fold)	68.3±2.3	**72.3**±2.1	73.9±1.9	74.3±2.2	79.7±1.8
HardProbe	64.8±2.9	65.3±2.5	66.4±1.1	66.2±0.7	67.0±1.1
Soft baselines					
Transformer	51.4±1.2	51.2±0.9	51.0±0.8	51.8±0.9	51.9±0.6
MLP	50.5±0.8	50.6±1.2	49.7±1.1	51.1±1.6	51.4±1.3
SoftCosetNet	51.4±1.2	51.2±0.9	51.0±0.8	51.8±0.9	51.9±0.6

effective aggregation of non-adjacent evidence; it does not establish reconstruction of the generating matching.

5.4 Templates, frozen masks, and hygiene

Subgroup templates provide a strong prior at the smallest budgets, so Table 3 separates their contribution from exact execution. With the order-17 template available, adjacent HardProbe is already perfect at $n = 32$. Without templates, it reaches 62.6% at $n = 256$, whereas free-mask GARN reaches 97.1%. Both reach 100.0% at $n = 1024$. The improvement therefore persists when membership must be learned from labels rather than initialized from the target subgroup.

A frozen oracle membership mask with a trained linear head yields 100% at $n = 32$, consistent with the simple decision rule once the correct features are available. Conversely, random inverse/multiplication tables reduce GARN to $50.4 \pm 1.3\%$, compared with $99.9 \pm 0.1\%$ using correct tables (Appendix E.10). These complementary controls distinguish the benefits of membership priors from those of faithful field arithmetic.

Table 3: Templates vs. free masks on adjacent matched data (test %; mean±std, 5 seeds). Templates accelerate extreme few-shot learning; free-mask GARN still saturates by n=1024. Best in each column in **bold**.

	+ subgroup templates			Free masks		
Method	n=32	**256**	**1024**	n=32	**256**	**1024**
HardProbe	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	49.9±0.8	62.6±13.7	**100.0**±0.0
GARN (no fold)	86.7±6.8	99.5±0.3	**100.0**±0.1	61.7±4.7	97.1±2.8	**100.0**±0.0
RatioFeatTF	70.2±4.2	96.4±1.7	99.8±0.1	70.2±4.2	96.4±1.7	99.8±0.1
Transformer	50.5±1.1	51.3±1.8	49.8±0.4	50.5±1.1	51.3±1.8	49.8±0.4

Robustness of the operator effect. Bitwise and Fourier-log Transformers remain near chance through $n = 1024$; increasing width to 256 and depth to six does not close the gap at $n = 512$. Across subgroup orders $\{3, 5, 15, 17, 51\}$, template-free GARN reaches 97.7–100% at $n = 512$, while Transformers remain near chance. Appendix E details these controls, mixed labels, noise, and longer sequences.

6 DISCUSSION

Operator access and membership priors. Exact ratios expose a relation that token features leave to the learner. Templates additionally supply target-relevant structure; free-mask GARN separates these advantages, reaching 97.1% at $n = 256$ and 100.0% at $n = 1024$. Together with operator corruption, these results identify exact execution as the central sample-efficiency intervention.

What shuffled accuracy measures. At $n = 1024$, all-pair aggregators reach 96.9%, compared with 79.9% for GARN and 67.0% for fixed adjacency (Appendix E.6). Complete aggregation preserves membership counts, whereas row-normalized attention learns a weighted summary, plausibly explaining the gap. Near-chance partner recovery and density-matched results support the narrower interpretation that the gain comes from aggregating non-adjacent evidence. GARN provides a state-update interface, while direct pooling is stronger for the present sequence label.

Stronger arithmetic biases. Fourier and clock mechanisms explain learned modular arithmetic (Nanda et al., 2023; Zhong et al., 2023). Ratios also admit structure: if $a = \alpha^u$, $b = \alpha^v$, and $H = \langle \alpha^{255/d} \rangle$, membership is equivalent to $v - u \equiv 0 \pmod{255/d}$. Fixed Fourier-log features do not enforce this pairwise computation. Their failure leaves stronger modular-difference architectures and curricula as open alternatives; neither was evaluated here. The larger-budget Transformer result ($85.8 \pm 11.9\%$ at $n = 8192$) further supports a sample-efficiency interpretation.

Cost, masks, and transfer. Execution remains dense: $O(TL^2)$ ratio lookups and $O(TKL^2)$ mask responses for depth T, with $O(L^2D)$ ratio-embedding activations per block of width D. Field tables require $O(q^2)$ storage. These costs provide no basis for claiming a runtime advantage over soft models. The free-mask K sweep shows little accuracy sensitivity under identical initialization, without establishing diverse specialization (Appendix E.1). Weak unseen-order transfer ($56.6 \pm 0.6\%$) also separates arithmetic access from transferable membership learning. Larger fields, sparse execution, and stronger interaction objectives remain directions for evaluation.

7 CONCLUSION

GARN separates learned interaction weights from exact finite-field execution. On a diagnostic with matched single-byte marginals, this separation yields near-perfect adjacent classification from a few hundred labels and improves shuffled classification over fixed adjacency. Template-free masks and operator controls show that faithful arithmetic, rather than a target-membership prior alone, drives the gain. The broader lesson is that supplying an executable relation can change the supervision required to learn from it. Finite-field learning provides a precise setting in which to study that choice.

AI USE STATEMENT

In this work, generative AI tools were utilized for grammatical polishing, sentence structure optimization, and academic expression refinement of the manuscript text, and the experimental code of this study was generated with the assistance of Cursor AI. No generative AI tools were employed for the design of research schemes, conduction of core experiments, analysis of original experimental data, and formulation of research conclusions. All figures and illustrations in this manuscript were manually created by the authors using PowerPoint without any generative AI involvement, which effectively avoids AI hallucination problems in visual content production. All other undisclosed AI-assisted tasks are not applicable to this study. Furthermore, large language models (LLMs) were adopted to assist in sorting out literature frameworks and optimizing the logical fluency of manuscript paragraphs. All AI-generated and AI-assisted contents in this work, including the Cursor AI-generated experimental code and polished manuscript text, have been fully reviewed, verified, and revised by multiple authors. Specifically, the authors manually checked all AI-polished text contents to eliminate logical errors, semantic deviations and potential plagiarism risks; the AI-generated experimental code has been jointly examined, tested and confirmed to be completely correct by multiple authors to ensure the accuracy and reproducibility of experimental results; all AI-optimized manuscript frameworks and narrative structures were repeatedly verified and adjusted according to the actual research content and experimental results of this study. The authors take full and ultimate responsibility for all contents of this manuscript, including all texts, research claims, experimental descriptions, code implementations, academic conclusions and visual figures produced or optimized with the assistance of generative AI tools or manually created by the authors.

ETHICS STATEMENT

The study uses synthetic finite-field data and evaluates learning architectures rather than the security of deployed ciphers. It involves no human participants or personal data. Applications to cryptographic or protocol analysis should be limited to authorized research; performance on this diagnostic is not evidence of a practical cryptanalytic attack.

REPRODUCIBILITY STATEMENT

Section 3 specifies the synthetic data construction, Section 4 defines the model, and Section 5.1 describes the evaluation protocol. Appendix A documents implementation choices and split checks; Appendix C gives the assumptions and supporting proofs; and Appendix E records additional controls and ablations. The anonymous supplementary materials provide the model implementation, data generators, experiment runners, archived results, and rerun instructions.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations*, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, learning hierarchical structure. *arXiv preprint arXiv:2305.13673*, 2023.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39–48, 2016.

Zhenzhen Bao et al. On the explanation and enhancement of neural-inspired differential cryptanalysis. IACR Cryptology ePrint Archive 2024/322, 2024.

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*, 2022.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Emanuele Bellini, David Gerault, Anna Hambitzer, and Matteo Rossi. A cipher-agnostic neural training pipeline with automated finding of good input differences. *IACR Transactions on Symmetric Cryptology*, 2023(3):184–212, 2023.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Wilfried Bounsi et al. Transformers meet neural algorithmic reasoners. *arXiv preprint arXiv:2406.09308*, 2024.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

François Charton. Linear algebra with transformers. In *TMLR*, 2022.

Zitong Andrew Chen, Junaid Hasan, Akhil Srinivasan, Hemkesh Bandi, and Jarod Alper. Multiplication beyond groups: Stratified fourier mechanisms in transformer circuits. *arXiv preprint arXiv:2607.07066*, 2026.

Hanseul Cho et al. Arithmetic transformers can length-generalize in both operand length and count. *arXiv preprint arXiv:2410.15787*, 2024.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999, 2016.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The neural data router: Adaptive control flow in transformers improves systematic generalization. In *International Conference on Learning Representations*, 2022.

Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer, 2002.

Gregoire Deletang et al. Neural networks and the chomsky hierarchy. *ICLR*, 2023.

Darshil Doshi, Tianyu He, Aritra Das, and Andrey Gromov. Grokking modular polynomials. *arXiv preprint arXiv:2406.03495*, 2024.

Shaoxiong Duan, Yining Shi, and Wei Xu. From interpolation to extrapolation: Complete length generalization for arithmetic transformers. *arXiv preprint arXiv:2310.11984*, 2024.

Nouha Dziri et al. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, 2023.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems*, 2023.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022.

David Gérault, Thomas Peyrin, and Quan Quan Tan. Survey: Six years of neural differential cryptanalysis. IACR Cryptology ePrint Archive 2024/1300, 2024.

Aron Gohr. Improving attacks on round-reduced Speck32/64 using deep learning. In *Advances in Cryptology – CRYPTO*, pp. 150–179. Springer, 2019.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Andrey Gromov. Grokking modular arithmetic. *arXiv preprint arXiv:2301.02679*, 2023.

Thomas Hader, Daniela Kaufmann, and Laura Kovács. SMT solving over finite field arithmetic. In *LPAR*, pp. 238–256, 2023.

Tianyu Hu, Mingze Liu, Chenwei Weng, et al. Fourier circuits in neural networks and transformers: A case study of modular arithmetic with multiple inputs. *arXiv preprint arXiv:2402.09469*, 2024.

Borja Ibarz, Vitaly Kurin, George Papamakarios, Kyriacos Nikiforou, Mehdi Bennani, Róbert Csordás, Andrew Dudzik, Matej Bošnjak, Alex Vitvitskyi, Yulia Rubanova, Andreea Deac, Beatrice Bevilacqua, Charles Blundell, and Petar Veličković. A generalist neural algorithmic learner. In *Learning on Graphs Conference*, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Samy Jelassi, Stéphane d'Ascoli, Carles Domingo-Enrich, Yuhuai Wu, Yuanzhi Li, and François Charton. Length generalization in arithmetic transformers. *arXiv preprint arXiv:2306.15400*, 2023.

Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition. *arXiv preprint arXiv:2502.00829*, 2025.

Guillaume Lample and François Charton. Deep learning for symbolic mathematics. *ICLR*, 2020.

Nayoung Lee, Kartik Sreenivasan, Jason D Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers. In *International Conference on Learning Representations*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 2022.

Rudolf Lidl and Harald Niederreiter. *Finite Fields*, volume 20. Cambridge University Press, 1997.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations*, 2023.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Andreas Madsen and Alexander Rosenberg Johansen. Neural arithmetic units. In *International Conference on Learning Representations*, 2020.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, 2018.

Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, and Tom Goldstein. Transformers can do arithmetic with the right embeddings. In *Advances in Neural Information Processing Systems*, 2024.

Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Nikhil Zhao, and Sham Kakade. Uncovering a universal abstract algorithm for modular addition in neural networks. In *Advances in Neural Information Processing Systems*, 2025.

Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop at ICLR*, 2022.

Santiago Ontañón, Joshua Ainslie, Vaclav Cvicek, and Zachary Fisher. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. In *ICLR Workshop*, 2022.

Philip Quirke and Fazl Barez. Understanding addition in transformers. *arXiv preprint arXiv:2310.13121*, 2023.

Gleb Rodionov and Liudmila Prokhorenkova. Discrete neural algorithmic reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51851–51865. PMLR, 2025.

Lucas Paul Saldyt and Subbarao Kambhampati. Mind the gap: Quantifying mechanistic gaps in algorithmic reasoning via neural compilation. *arXiv preprint arXiv:2505.18623*, 2025.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *ICLR*, 2019.

Hikaru Shindo, Devendra Singh Dhami, and Kristian Kersting. Neuro-symbolic forward reasoning. *arXiv preprint arXiv:2110.09383*, 2021.

Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking group multiplication with cosets. *arXiv preprint arXiv:2312.06581*, 2024.

Hu Tan, Kuo Gai, and Shihua Zhang. Beyond neural collapse: Task-intrinsic geometry governs neural representations in modular arithmetic. *arXiv preprint arXiv:2606.08985*, 2026.

Andrew Trask, Felix Hill, Scott Reed, John Rae, Chris Dyer, and Phil Blunsom. Neural arithmetic logic units. In *Advances in Neural Information Processing Systems*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7), 2021.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, 2015.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Xingcheng Xu, Zibo Zhao, Haipeng Zhang, and Yanqing Yang. Principled understanding of generalization for generative transformer models in arithmetic reasoning tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4721–4747, 2025.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua Tenenbaum. Neural-symbolic VQA: Disentangling reasoning from vision and linguistic understanding. In *Advances in Neural Information Processing Systems*, 2018.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. In *Journal of Machine Learning Research*, 2024.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Advances in Neural Information Processing Systems*, 2023.

Hattie Zhou et al. What algorithms can transformers learn? a study in length generalization. In *ICLR*, 2024.

APPENDIX CONTENTS

A Additional implementation details 16
B Extended related work 16
B.1 Neural cryptanalysis and algebraic diagnostics 16
B.2 Arithmetic learning, grokking, and coset circuits 16
B.3 Neuro-symbolic reasoning and discrete latents 17
C Theoretical proofs and supporting lemmas 17
C.1 Matched-marginal sampling 17
C.2 Hard execution and oracle routing 18
C.3 Shuffled partners and density cues 18
D Extended discussion of soft failure 19
E Additional experiments and ablations 19
E.1 Mask bank size K (free masks) 19
E.2 Representation-aware soft encoders 19
E.3 Templates vs. free masks; matched features 19
E.4 Transformer capacity and training controls 19
E.5 Multiple subgroup orders 19
E.6 All-pair membership aggregators (shuffled) 19
E.7 Omit-correct-order templates; varying-H pools 20
E.8 Cross-H generalization 20
E.9 Attention partners vs. sequence accuracy 21
E.10 Mixed labels, corrupted tables, larger soft budgets 21
E.11 Noise and longer sequences 21
F Broader impact note 21

The appendix follows the paper structure: implementation notes, extended positioning, supporting lemmas, soft-failure discussion, ablations, and impact.

A ADDITIONAL IMPLEMENTATION DETAILS

Unless stated otherwise, GARN uses SoftGF embeddings (learned lookup + discrete-log / inverse features as real-valued coordinates). SoftGF is an input featurization only: it does not implement multiply/inverse table execution. Hard execution uses a precomputed 256×256 multiplication table and a 256-entry inverse table for AES $\mathrm{GF}(2^8)$. We verified that the exp/log generator 0×03 is primitive (order 255); using 0×02 (order 51) produces incorrect subgroups and invalidates the algebraic claims. Self-pairs are masked in both Propose and Execute. The primary tables report **test** accuracy after selecting the best validation checkpoint; hyperparameters are fixed a priori (Sec. 5.1). On a representative matched pool (n=1024/1000/1500), sequence overlap between train and test is zero by construction of the split permutation. Ratio supports cover all of $\mathbb{F}^*$ in both splits (domain size 255), and exact (a, b) pair tokens recur across splits (shared small domain). Near-perfect few-shot hard-probe results should therefore be read as learning the shared ratio membership predicate rather than memorization of full sequences. Byte-class total variation on the full pool is small (TV ≈ 0.07 in our draw). On standard shuffled data, the mean off-diagonal in-H edge counts are ≈ 11.0 for positives and ≈ 3.2 for negatives out of $L(L-1)$=56 ordered pairs.

Density-matched shuffled generator (appendix analysis). To study aggregate-density cues, we also construct negatives that match a positive's total ordered off-diagonal in-H count (tolerance 0) while keeping planted partners out of H. We greedily rewrite non-partner bytes by $B_j \leftarrow B_i \cdot h$ for $h \in H$ (or randomize if overshooting), rejecting rewrites that would make a planted partner ratio land in H. Matching is per-example and post-permutation. In our pool, mean densities are 11.30 and 10.93 (gap 0.37), with 93% exact pair matches, partner in-H rates of $100\%/0\%$, residual byte TV ≈ 0.11, and fallback $357/6000$.

B EXTENDED RELATED WORK

This appendix expands the brief positioning in Sec. 2. We organize by theme and emphasize how each line of work differs from our matched-marginal few-shot coset diagnostic.

B.1 NEURAL CRYPTANALYSIS AND ALGEBRAIC DIAGNOSTICS

Gohr's neural differential distinguishers (Gohr, 2019) initiated a large literature on learning statistical biases in round-reduced ciphers; subsequent surveys catalog architectures, primitives, and key-recovery pipelines (Gérault et al., 2024). Cipher-agnostic training pipelines and specialized distinguishers (Bellini et al., 2023) improve automation, while interpretability studies relate neural advantages to differential and XOR features beyond classical DDTs (Bao et al., 2024). These works target *implicit* leakage in concrete primitives. Our protocol instead isolates an *explicit* multiplicative predicate on AES $\mathrm{GF}(2^8)$ under matched-marginal sampling, and asks which architectural inductive biases recover it under scarce labels. Symbolic finite-field constraint solvers (Hader et al., 2023) address a harder satisfiability regime; we deliberately use supervised binary classification as a diagnostic rather than a solver substitute. We do not claim cryptanalytic breaks of deployed AES.

B.2 ARITHMETIC LEARNING, GROKKING, AND COSET CIRCUITS

A substantial literature studies when Transformers and MLPs learn arithmetic (Lee et al., 2024; Quirke & Barez, 2023; Cho et al., 2024; McLeish et al., 2024; Trask et al., 2018; Madsen & Johansen, 2020), modular operations and grokking (Power et al., 2022; Nanda et al., 2023; Gromov, 2023; Doshi et al., 2024; Barak et al., 2022), and length/compositional generalization (Jelassi et al., 2023; Dziri et al., 2023; Zhou et al., 2024; Deletang et al., 2023; Duan et al., 2024; Xu et al., 2025; Ontañón et al., 2022; Csordás et al., 2022; Liu et al., 2023). Theory connecting bounded-depth Transformers to finite-field arithmetic and chain-of-thought (Feng et al., 2023; Wei et al., 2022; Nye et al., 2022) clarifies expressivity limits that are distinct from our few-shot sample-efficiency protocol. Analyses of in-context learning (Garg et al., 2022; Akyürek et al., 2023; Zhang et al., 2024) and hierarchical structure (Allen-Zhu & Li, 2023) similarly separate capacity from inductive bias under scarce labels. Mechanistic work shows that modular addition often implements Fourier / "clock" or "pizza" circuits (Zhong et al., 2023; Nanda et al., 2023; Gromov, 2023), that multi-input modular tasks admit Fourier constructions (Hu et al., 2024), and that

networks can discover approximate-coset structure related to CRT-style decompositions (Morwani et al., 2025). Group-composition reverse engineering further highlights coset-based circuits on permutation groups (Stander et al., 2024). Modular multiplication and task-intrinsic geometry (Chen et al., 2026; Tan et al., 2026), trigonometric encodings (Kantamneni & Tegmark, 2025), and classical symbolic-math learning (Saxton et al., 2019; Lample & Charton, 2020; Charton, 2022; Lewkowycz et al., 2022) likewise show that representation and supervision format dominate outcomes. Relative to this body of work, our negative few-shot results are *protocol-relative*: under matched-marginal byte sequences and ≤ 1024 labels, the tested soft configurations do not efficiently acquire $\mathrm{GF}(2^8)$ ratio membership, whereas providing the exact operator does.

B.3 Neuro-symbolic reasoning and discrete latents

Neural algorithmic reasoning (Veličković & Blundell, 2021; Bounsi et al., 2024; Rodionov & Prokhorenkova, 2025; Ibarz et al., 2022) and neural module networks (Andreas et al., 2016) separate learned control from structured execution. Classical interfaces include Neural Turing Machines and pointer networks (Graves et al., 2014; Vinyals et al., 2015). Neuro-symbolic forward reasoning (Shindo et al., 2021), neural-symbolic VQA (Yi et al., 2018), and neural probabilistic logic (Manhaeve et al., 2018) keep symbolic entailment exact while learning perception/control. Discrete NAR (Rodionov & Prokhorenkova, 2025) and neural-compilation gap analyses (Saldyt & Kambhampati, 2025) clarify distinctions among expressivity, training, and access to an executable algorithm. Geometric / relational inductive biases (Cohen & Welling, 2016; Bronstein et al., 2021; Battaglia et al., 2018) encode continuous symmetries; multiplicative structure in $\mathrm{GF}(2^8)^*$ is not obviously captured by Euclidean token similarity. Straight-through estimators (Bengio et al., 2013) and Gumbel-/Concrete relaxations (Jang et al., 2017; Maddison et al., 2017) enable gradients through discrete choices; we evaluate STE and hard Gumbel algebraic baselines and treat their few-shot failure as empirical for the tested configurations, not as a general impossibility theorem. GARN instantiates the control/execution split for finite-field tables: attention routes partners; AES $\mathrm{GF}(2^8)$ ops execute without differentiation through the table.

C Theoretical proofs and supporting lemmas

We record elementary supporting lemmas for the matched-marginal protocol and hard Propose–Execute block. These statements characterize the generator and the information supplied by exact operators.

C.1 Matched-marginal sampling

Recall $\mathbb{F} = \mathrm{GF}(2^8)$, $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$, and a cyclic subgroup $H \leq \mathbb{F}^*$ of order $d \mid 255$. A matched-marginal example with label $y \in \{0, 1\}$ is generated by sampling, for each pair index t,

$$r_t \sim \mathrm{Unif}(H) \text{ if } y{=}1, \qquad r_t \sim \mathrm{Unif}(\mathbb{F}^* \setminus H) \text{ if } y{=}0,$$

then $a_t \sim \mathrm{Unif}(\mathbb{F}^*)$ and $b_t = a_t \cdot r_t$.

Lemma C.1 (Matched single-byte marginals)**.** *Under the matched-marginal generator, for every position in an adjacent encoding and every $c \in \mathbb{F}^*$,*

$$\Pr[a_t = c \mid y] = \Pr[b_t = c \mid y] = \tfrac{1}{|\mathbb{F}^*|},$$

independent of y. Consequently, a single observed byte carries no information about the label, and the expected sequence histogram is identical across classes. The distribution of the full histogram need not be identical.

Proof. By construction a_t is drawn uniformly from $\mathbb{F}^*$ independently of y. Given a_t, $b_t = a_t \cdot r_t$ with r_t uniform on a nonempty subset $S_y \subseteq \mathbb{F}^*$ ($S_1 = H$, $S_0 = \mathbb{F}^* \setminus H$). For fixed c, as a_t ranges uniformly over $\mathbb{F}^*$, so does $a_t \cdot r$ for any fixed nonzero r; averaging over $r \sim \mathrm{Unif}(S_y)$ preserves uniformity. Thus each single-byte marginal is independent of y. Linearity of expectation gives identical expected histograms. This does not establish independence of joint statistics: for example, $a_t = b_t$ occurs with probability $1/|H|$ in a positive generating pair and never in a negative pair, since $1 \in H$. □

Lemma C.2 (Self-pair membership)**.** *For every $a \in \mathbb{F}^*$ and every subgroup $H \leq \mathbb{F}^*$, $a^{-1} \cdot a = 1 \in H$.*

Proof. Immediate from the subgroup axioms. □

Remark C.1. Lemma C.2 motivates masking self-pairs in Propose/Execute: otherwise every token would contribute a spurious in-H self-edge.

C.2 Hard execution and oracle routing

Let $B \in (\mathbb{F}^*)^L$ with $L{=}2P$ encode adjacent pairs (B_{2t-1}, B_{2t}) for $t = 1, \ldots, P$. Write $r_{ij} = B_i^{-1} \cdot B_j$ and let $m^\star_{ij} = \mathbf{1}[r_{ij} \in H] \cdot \mathbf{1}[i \neq j]$ be the oracle membership feature.

Proposition C.1 (Adjacent HardProbe with oracle mask)**.** *On adjacent matched data with homogeneous labels, the feature*

$$\phi = \tfrac{1}{P} \sum_{t=1}^{P} m^\star_{2t-1,2t}$$

satisfies $\phi = 1$ almost surely on positives and $\phi = 0$ almost surely on negatives. Hence a linear threshold on ϕ achieves Bayes error 0.

Proof. By construction, every adjacent generating pair has $r \in H$ iff $y{=}1$. Self-pairs are excluded by the $i \neq j$ indicator. Averaging P deterministic indicators yields $\phi \in \{0, 1\}$ equal to the label. □

Proposition C.2 (Oracle routing recovers HardProbe)**.** *Suppose Propose returns a row-stochastic π that places mass 1 on the true adjacent partner for each left token of a generating pair (and symmetrically), and Execute uses $m^\star$. Then the membership evidence $e_i = \sum_{j \neq i} \pi_{ij} m^\star_{ij}$ coincides with the HardProbe adjacent features up to a fixed permutation of coordinates. In particular, mean-pooled e is a sufficient statistic for the Bayes-optimal classifier of Proposition C.1.*

Proof. Under oracle π, each sum $\sum_j \pi_{ij} m^\star_{ij}$ collapses to the single adjacent partner membership bit. Homogeneous labeling makes the mean of these bits equal to ϕ in Proposition C.1. □

Remark C.2. Propositions C.1–C.2 explain why HardProbe is an operator ceiling on adjacent data and why learned Propose can matter once adjacency is broken. They characterize the sufficient features provided by exact execution and oracle routing.

C.3 Shuffled partners and density cues

Let σ be a uniform random permutation of $\{1, \ldots, L\}$ applied after matched generation, and write $N_H = \sum_{i \neq j} \mathbf{1}[B^{-1}_{\sigma(i)} \cdot B_{\sigma(j)} \in H]$ for the number of ordered off-diagonal in-H ratios.

Lemma C.3 (Label-dependent expected density)**.** *Under matched generation followed by a uniform shuffle, $\mathbb{E}[N_H \mid y{=}1] > \mathbb{E}[N_H \mid y{=}0]$. In particular, aggregate in-H density is an informative (but incomplete) cue for the sequence label.*

Proof. There are $2P$ ordered generating-pair edges. In the positive class both directions lie in H, since H is closed under inversion; in the negative class neither does. Tokens from different generating pairs are independent and uniform on $\mathbb{F}^*$, so each of the remaining $L(L-1) - 2P$ ordered ratios lies in H with probability $|H|/255$. Hence

$$\mathbb{E}[N_H \mid y] = 2Py + \big(L(L-1) - 2P\big)|H|/255.$$

The expected gap is $2P$, and a permutation leaves N_H unchanged. □

Remark C.3. Lemma C.3 motivates the density-matched negative control in App. A/E.6: when negatives are rewritten to match N_H, sequence-level accuracy of density aggregators collapses toward chance, showing that high shuffled accuracy need not certify unique planted-pair recovery.

Remark C.4 (Fixed adjacency after shuffling). After shuffling, fixed adjacent positions need not contain generating partners. The deterministic guarantee of Proposition C.1 therefore no longer applies to that positional probe; its observed accuracy is reported in Table 2.

D EXTENDED DISCUSSION OF SOFT FAILURE

SoftCosetNet and the Transformer share the same encoder backbone in our suite and yield nearly identical curves, so merely renaming attention heads does not create algebraic bias. Representation-aware soft encoders (bitwise features; multiplicative Fourier features of discrete log) also remain near chance at $n \leq 1024$ (Appendix E.2). STE and hard Gumbel-Softmax algebraic variants must jointly learn discrete byte identities and subgroup masks; under our few-shot protocol both stay at chance (49.1–50.5%), so the failure is not specific to a single discrete estimator.

E ADDITIONAL EXPERIMENTS AND ABLATIONS

All numbers below are **test** accuracy (%; mean±std over 3 seeds) unless noted, with model selection on a held-out validation split.

E.1 MASK BANK SIZE K (FREE MASKS)

On adjacent matched data with free masks, template-free GARN shows the same reported accuracy across the tested numbers of membership masks $K \in \{1, 2, 4, 8\}$: all settings give $61.7\pm4.7\%$ at n=32 and $100.0\pm0.0\%$ at n=1024 (5 seeds). Free-mask logits are initialized identically at zero. The $\max_k$ reduction passes each ratio's gradient to a selected maximizing entry, including a selected winner at ties; it does not encourage mask diversity. The equal accuracy across K is therefore compatible with redundant masks. We did not measure specialization or compare diversity regularization and gate initializations. In the template setting, the initial gate favors the supplied templates ($\sigma(g) \approx 0.88$), so its role is distinct from the free-mask sweep.

E.2 REPRESENTATION-AWARE SOFT ENCODERS

Bitwise and Fourier-log Transformers remain near chance on adjacent matched data at $n \leq 1024$ (49.3–50.6%; 5 seeds), matching plain Transformers. Thus the soft failure is not resolved by these representation choices alone under our few-shot protocol (McLeish et al., 2024).

E.3 TEMPLATES VS. FREE MASKS; MATCHED FEATURES

Table 4 shows that cyclic templates are a strong *few-shot prior*, not a necessity at moderate n. Template-free GARN still reaches perfect accuracy by n=1024. Transformers given the same algebraic byte features do *not* rise above chance. A RatioFeatTransformer that also consumes exact ratios becomes competitive, confirming that algebraic *access*, rather than continuous attention alone, drives the soft-to-hard gap.

E.4 TRANSFORMER CAPACITY AND TRAINING CONTROLS

With n=512 fixed, Transformers scaled to depth 6 and width 256 with cosine schedules and optional algebraic embeddings remain at chance (49.9–51.8%), while template-free GARN reaches $99.8\pm0.1\%$. Increasing capacity in these configurations does not resolve the gap at the fixed label budget.

E.5 MULTIPLE SUBGROUP ORDERS

Across $|H| \in \{3, 5, 15, 17, 51\}$ (template-free models, n=512), Transformers stay near 50%. GARN scores 97.7–100%, with larger $|H|$ harder in the few-shot regime (e.g., $|H|$=51: 55.3% at n=64 vs. 96.5% for $|H|$=3).

E.6 ALL-PAIR MEMBERSHIP AGGREGATORS (SHUFFLED)

Table 5 reports non-routing aggregators that consume exact membership scores on shuffled data. These can match or exceed the sequence-level accuracy of GARN, as expected when every off-diagonal ratio is executed and pooled. We therefore present GARN as a *routing interface* for hard

Table 4: Matched coset ablations (test %; mean±std). Templates act as a few-shot prior; free masks still solve the task at moderate n. Algebraic embeddings alone do not lift Transformers above chance. Best in each column in **bold**.

	Training set size n			
Method	32	128	512	1024
Hard execution / algebraic access				
HardProbe + templates	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0
HardProbe free mask	51.8±3.0	52.7±2.5	91.2±9.7	**100.0**±0.0
GARN + templates	89.5±4.6	98.6±0.5	99.9±0.1	99.9±0.1
GARN free mask	69.3±2.1	83.6±4.2	99.9±0.1	**100.0**±0.0
GARN free + plain embed	51.4±1.2	51.0±0.3	86.4±4.1	99.7±0.1
RatioFeatTransformer	75.1±1.1	91.5±1.8	99.1±0.4	99.8±0.1
Soft baselines				
Transformer + alg. embed	50.0±0.5	49.1±1.1	50.4±0.6	50.1±0.4
Transformer plain	50.9±0.4	49.7±1.9	49.9±1.1	50.3±0.4

Table 5: All-pair membership aggregators on shuffled vs. density-matched data (test %; mean±std, 5 seeds). On ordinary shuffled data, complete aggregators can exceed Propose–Execute sequence accuracy; under density matching all methods collapse to chance. Best in each shuffled column in **bold**.

	Shuffled (templates)			Density-matched		
Method	n=32	256	1024	n=32	256	1024
Propose–Execute / fixed adjacency						
GARN (no fold)	70.6±3.0	74.3±3.5	79.9±1.4	49.9±1.1	49.9±1.3	49.9±1.3
HardProbe	64.8±2.9	66.4±1.1	67.0±1.1	50.4±0.7	51.0±0.5	49.6±0.9
All-pair membership aggregators						
AggDensity	78.0±23.2	77.9±17.8	**96.9**±0.2	50.6±1.5	50.6±1.4	50.5±0.7
MembershipSet	81.2±20.1	**96.6**±0.2	**96.9**±0.2	50.5±1.5	51.0±1.6	50.6±0.9
Combinatorial	89.2±7.2	95.5±2.3	**96.9**±0.2	50.0±0.5	50.3±0.8	49.8±0.8
AttnPoolMem	**95.5**±2.6	**96.6**±0.2	**96.9**±0.2	51.2±1.5	51.0±1.2	49.9±0.9
RatioFeatTF	66.1±1.8	78.9±1.2	85.2±1.7	50.3±1.1	49.4±1.2	49.5±1.2

field ops, with lower sequence accuracy than the strongest complete all-pair aggregators on this generator. Under density matching, these aggregators (and GARN) fall to chance. Aggregate in-H density is therefore a strong cue on the ordinary shuffled generator, and sequence accuracy alone should not be read as unique planted-pair recovery. At small n, density-style heads can be unstable (e.g., AggDensity at n=32 gives per-seed $\{97.0, 48.2, 51.0, 96.9, 96.8\}\%$).

E.7 OMIT-CORRECT-ORDER TEMPLATES; VARYING-H POOLS

When the template bank is restricted to orders $\{15, 51\}$ (*templates-only*, no free mixing) on the $|H|$=17 adjacent task, adjacent HardProbe still reaches $86.4\pm7.3\%$ at n=32 via anti-correlated membership on wrong-order subgroups. Free-mask GARN recovers the usual template-free curve ($61.7\% \rightarrow 100\%$). On a mixed-order pool with per-example $|H| \in \{3, 5, 15, 17, 51\}$ and free masks, GARN reaches $96.7\pm0.7\%$ at n=1024 while Transformers remain at chance.

E.8 CROSS-H GENERALIZATION

Training on orders $\{3, 5, 15, 17\}$ and testing on unseen $|H|$=51 yields only $56.6\pm0.6\%$ for multi-bank GARN (seen-order val $\approx 100\%$). Transformers remain at chance on both. Order-specific mask learning and transfer to new orders remain open.

E.9 ATTENTION PARTNERS VS. SEQUENCE ACCURACY

After training template-free GARN on the shuffled sequence label (n=1024), top-1 attention partners match ground-truth partners at $14.2\pm0.2\%$ (chance $1/7$), while sequence test accuracy is $77.0\pm0.7\%$. Shuffled gains over HardProbe therefore reflect useful non-adjacent algebraic evidence under the sequence objective, rather than certified reconstruction of a unique planted matching.

E.10 MIXED LABELS, CORRUPTED TABLES, AND LARGER SOFT BUDGETS

On mixed-fraction labels (majority of pairs in H), template-free GARN reaches $85.3\pm1.5\%$ vs. $48.9\pm1.6\%$ for Transformers (n=512). Replacing multiply/inverse tables by fully random tables drops GARN to $50.4\pm1.3\%$ while the correct-table variant stays at $99.9\pm0.1\%$. Scaling Transformers to n=8192 yields $85.8\pm11.9\%$. Soft models can improve with far more data, supporting a sample-complexity interpretation of the title claim. Log-domain encodings alone remain near chance across these budgets.

E.11 NOISE AND LONGER SEQUENCES

With pair-violation noise $0/12.5/25\%$, template-free GARN at n=512 scores $99.9/90.9/73.6\%$, degrading gracefully while Transformers stay near chance. For $P \in \{4, 8, 16\}$ at n=512, GARN stays $\geq 99.8\%$ and Transformers remain $\approx 50\%$.

F BROADER IMPACT NOTE

Among the positive impacts are better evaluation hygiene for AI-for-crypto papers and tools for detecting algebraic regularities in coding and protocol settings. We discourage overclaiming cryptographic breaks from synthetic coset probes.