In-Context Learning Amplifies a Latent Symbolic Circuit

Melissa Wessel [1]

Abstract

Large language models can learn abstract rules from just a few in-context examples, but how their internal mechanisms activate as examples accumulate is not well understood. We trace a three-stage symbolic reasoning circuit (abstraction, induction, retrieval) across shot counts in three model families and find it is detectable and functional well before the model achieves high accuracy. Per-head causal contribution grows up to $8\times$ from 1- to 10-shot, and cross-shot activation patching raises accuracy from 1% to 56% at 0-shot and 17% to 88% at 1-shot. Function vectors scaled and injected at 0-shot rescue accuracy up to 86%, largely substituting for the induction stage but depending critically on an intact downstream retrieval stage. The infrastructure for abstract rule-following is present in the weights before any demonstrations; in-context examples, function vectors, and related interventions appear to supply input to the same latent circuit.

1. Introduction

Large language models have demonstrated striking capabilities on abstract reasoning tasks, including those requiring generalization over arbitrary tokens (Webb et al., 2023; Musker et al., 2025), but the mechanisms underlying these capabilities are not yet well understood. Recent work has identified a three-stage circuit for symbolic reasoning (Yang et al., 2025b); that work, however, leaves open the question of what role in-context examples play in the circuit's operation. More broadly, the relationship between in-context demonstrations and internal mechanisms remains an active area of study: work on function vectors and concept vectors has characterized how task identity is represented during in-context learning (ICL, Todd et al., 2024; Hendel et al., 2023; Opiełka et al., 2026), while theoretical analyses have modeled ICL as implicit algorithm construction within the forward pass (von Oswald et al., 2023; Akyürek et al., 2024). A priori, several options are possible: new heads could be recruited into the circuit as examples accumulate, entire stages could come online discretely as with induction heads during pretraining (Elhage et al., 2021; Olsson et al., 2022), or heads important at low shot counts could become irrelevant at high shot counts as the model shifts strategy. In this work, we trace the developmental trajectory of the Yang et al. circuit across shot counts and find that it is identifiable and functional well before the model achieves high accuracy, suggesting that even in abstract configurations of arbitrary tokens absent any semantic meaning, examples amplify a latent computation.

We study the circuit identified by Yang et al. (2025b) for algebraic identity rule induction (Marcus et al., 1999), a task in which models must generalize over arbitrary tokens, precluding reliance on memorized semantic associations (Webb et al., 2023; Musker et al., 2025). The model sees N in-context examples of three-token patterns (ABC, delimited by ˆ , separated by line breaks) and must complete the final token of a partial example. Two rules are used: ABA (C = A) and ABB (C = B). A sample 2-shot ABA prompt:

```
iac ˆ ilege ˆ iac
ptest ˆ yi ˆ ptest
ks ˆ ixe ˆ __?
```

The circuit comprises three stages: Symbolic Abstraction (SA) heads convert input tokens into variables, Symbolic Induction (SI) heads perform sequence induction over those variables, and Retrieval (Ret) heads recover the predicted token. Yang et al. (2025b) observed the circuit across several architecture families, however, their analysis focused on the mature circuit at high accuracy, primarily showcasing Llama 3.1-70B (Grattafiori et al., 2024) at 2-shot where accuracy already exceeds 95%. By contrast, smaller models such as Gemma 2-2B (Gemma Team, 2024) require up to 10 shots to reach comparable performance, providing a natural window into how the circuit develops as in-context examples accumulate.

We trace this development in Gemma 2-2B (Gemma Team, 2024), Llama 3.1-8B (Grattafiori et al., 2024) and Qwen 3-4B (Yang et al., 2025a) and find a consistent picture: stable circuit topology with quantitative signal growth. The circuit is observable via causal mediation analysis (CMA) from

[1]Independent Researcher. Correspondence to: Melissa Wessel <melissawessel00@gmail.com>.

Accepted to the Mechanistic Interpretability Workshop at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

just 1-shot, when baseline accuracy for Gemma 2-2B is only 17%, with per-head causal contribution increasing up to 8× between 1-shot and 10-shot. Patching 10-shot activations of SA, SI, and Retrieval heads into 1-shot prompts raises accuracy to 88%. At 0-shot, SA heads cannot be tested as they write to the C positions which do not exist, however the downstream SI and Retrieval stages are functional and writable, with 10-shot activations raising accuracy from 1% to 56%, activating infrastructure that predates any demonstrations. SI stage patches specifically transfer even across prompts with no shared tokens, paralleling prior findings on function vectors (Todd et al., 2024). Function vectors extracted from CMA-significant heads rescue 0-shot accuracy to 86% when injected into the residual stream but collapse to 13% when the downstream Retrieval stage heads are ablated: the FV is causally interchangeable with the SI stage's output (fully on ABB, partially on ABA), contingent on Ret remaining available to read it out.

These results contribute to a mechanistic account of ICL of abstract rules: examples strengthen signal through circuitry that already exists in the weights. ICL has been studied extensively at behavioral (Brown et al., 2020) and theoretical (von Oswald et al., 2023; Akyürek et al., 2024) levels; our results address the mechanistic question of how examples reshape computation in the forward pass, at least for this task family. They also speak to the relationship between ICL and interventions such as function vectors (Todd et al., 2024) and activation steering (Zou et al., 2023; Turner et al., 2024; Liu et al., 2024), suggesting that such interventions may succeed in part because they supply input signal into pre-existing computational pathways.

2. Circuit Amplification

In Gemma 2-2B, 1-shot accuracy is 17% and increases to 93% by 4-shot, reaching 99% by 10-shot. Llama 3.1-8B performs better but is still imperfect at 1-shot (60% ABA), and Qwen 3-4B tracks Gemma more closely (27% ABA, 55% ABB at 1-shot). We evaluate whether the three-stage topology Yang et al. (2025b) identified is observable from 1-shot across all three models, and track how it changes from 1 to 10 shots.

Method. We follow Yang et al. (2025b) and apply their CMA from 0 to 10-shot. Their method uses two contrastive conditions. The abstract condition (varying rule, fixed answer token) identifies SA heads when patched at C positions and SI heads when patched at the query position. The token condition (fixed rule, varying answer token) identifies Retrieval heads at the query position. To quantify topology stability across shot counts we rank heads per stage by mean CMA score and compare rankings using rank-biased overlap (RBO; Webber et al. 2010, p=0.9), a top-weighted [0,1] similarity.

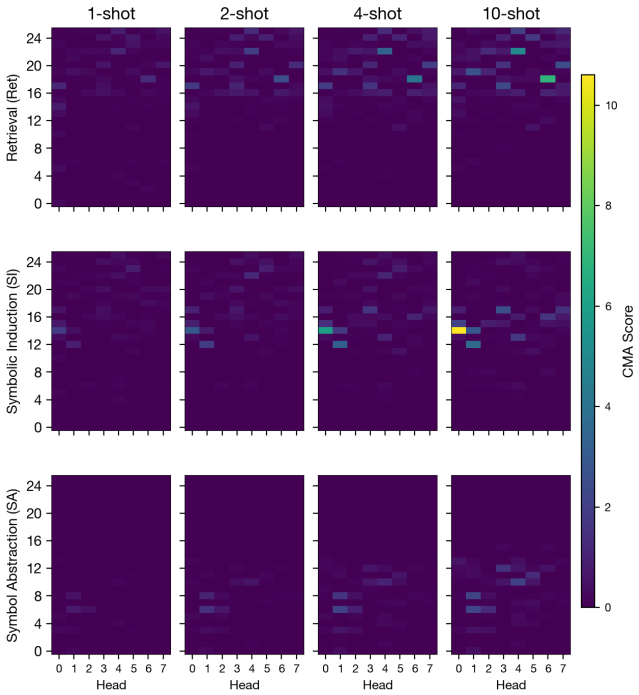
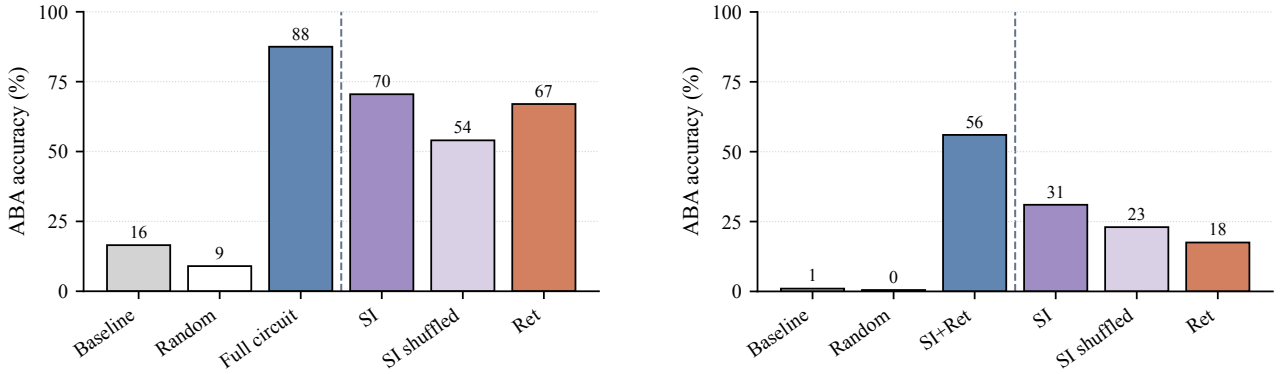


Figure 1. **The Symbolic Reasoning Circuit in Gemma 2-2B at 1 through 10-Shot.** CMA Scores by layer and head by circuit stage across shot counts.

Results. The three-stage circuit is observable from 1-shot in all three models, and changes across shot counts are dominated by signal growth rather than topology change. We describe Gemma 2-2B in detail as it shows the largest accuracy transition, then note how Llama 3.1-8B and Qwen 3-4B compare.

In Gemma, CMA scores grow up to 8× from 1 to 10 shots without major changes in circuit topology (Figure 1). The top-5 heads per stage overlap 4-5/5 across adjacent shot counts for SI and Retrieval (RBO 0.81-0.94; random baseline 0.12, Table 6), and 3-5/5 for SA (RBO 0.66-0.91), consistent with SA's lower overall CMA signal. The persistent SA core (L6H1, L6H2, L8H1, Table 5) is present from 1-shot onward, and no stage exhibits discrete onset. Signal grows within this relatively fixed topology: the dominant SI head L14H0 increases from 0.9 to 7.4 (8.1×) for ABB, and the top Retrieval head L18H6 from 1.3 to 7.0 (5.6×), with amplification largest across the 2-to-4-shot transition where accuracy itself jumps most sharply. The same pattern holds on letter-string analogies (e.g., [a b c] → [a b d], [d e f] → [d e ?, see Appendix A.5).

Llama 3.1-8B shows a strikingly parallel pattern despite its different accuracy regime (60% → 100%): 4/5 persistent core SI heads, 4/5 Retrieval, and 3/5 SA. Qwen 3-4B is



(a) 1-Shot *(b)* 0-Shot

Figure 2. **Cross-Shot Patching Results (ABA).** Accuracy of Gemma 2-2B when patching 10-shot donor activations into 1-shot (a) and 0-shot (b) target prompts. Full results for all models and both rules are in Appendix Figures 7–9.

noisier at top-5 (1/5 SA, 1/5 SI, 3/5 Ret) but recovers at top-10 (5/10 SA, 4/10 SI, 7/10 Ret); inspection of the rankings shows the top-5 noise reflects reranking among the leading heads rather than new heads entering. Stability here is a claim about which heads carry the computation, not exact rank order; the lowest-signal regimes (Qwen at 1-shot, SA broadly) show the most reranking among the leading heads.

3. Cross-Shot Activation Patching

CMA reveals that the circuit is observable at low shot counts, but observation alone cannot establish that it is functional and similar enough to its more mature counterparts for its outputs to be transferrable. To test this, we intervene directly: patching activations from 10-shot forward passes into 1-shot and even 0-shot prompts and measuring the effect on accuracy.

3.1. Patching from 10-shot to 1-shot

For each 1-shot target, we construct a matched pair where the target's single example is the final example of the 10-shot donor (shared tokens, same rule). We patch at stage-specific positions: SA heads at C positions, SI and Ret heads at the query position. We test each stage independently and all stages together, including a random control and a shuffled token control: the donor uses the same abstract rule but entirely different tokens. We evaluate 200 prompts per rule.

Patching all stages from the 10-shot donor rescues ABA accuracy from 17% to 88% (Figure 2a), and ABB from 57% to 93%. Patching Retrieval heads alone yields only 67% (ABA), confirming that the rescue depends on upstream rule computation. Pure answer-token injection would predict the model's native 10-shot ceiling (99%); the gap below that implies upstream rule computation.

3.2. Patching from 10-shot to 0-shot

For 0-shot targets, the prompt contains only the query `A^B^` with no in-context examples. We construct matched triplets: a 10-shot ABA donor, a 10-shot ABB donor, and a 0-shot prompt sharing the same query tokens (Figure 4). Using permutation-significant heads from 10-shot CMA (SI and Ret only, since SA writes to C positions absent at 0-shot), we patch per-head attention output from each donor's query position to the 0-shot forward pass.

At 0-shot, rule-following accuracy is 1% (ABA) and 8% (ABB), yet patching 7 SI and 9 Retrieval heads out of 208 total rescues accuracy to 56% (ABA, Figure 2b) and 73% (ABB). Patching Ret heads alone yields an accuracy of 18% (ABA), again ruling out simple answer injection by the retrieval stage. Critically, the same `A^B^` prompt predicts A when patched from an ABA donor and B from an ABB donor; the injected activations, not the prompt, determine the answer. Patching random non-circuit heads yields $<1\%$ (ABA), confirming the effect is specific to the three-stage circuit.

3.3. Token Invariance of the SI Stage

SI heads appear to carry abstract, largely token-invariant rule information, though the degree of token-invariance varies by model. At 1-shot, matched-token and shuffled-token SI patching rescue Llama 3.1-8B accuracy to nearly identical levels on both rules (ABA: 83% vs 83%; ABB: 83% vs 81%), despite donor and target sharing no tokens. Gemma 2-2B and Qwen 3-4B show only partial transfer: shuffled-token SI patching captures roughly two-thirds (69%) of the matched-token effect on ABA in Gemma and roughly half (52%) in Qwen.

The same pattern holds at 0-shot. In Llama 3.1-8B, shuffled-token SI patching is indistinguishable from matched-token

on both rules (ABA: 36% vs 39%; ABB: 37% vs 38%). In Gemma 2-2B, shuffled patching captures 73% of the matched-token ABA rescue. For Qwen 3-4B we report ABB rather than ABA: the 0-shot ABA condition produces only a 7-point SI-only effect over a 0% baseline, making the matched-vs-shuffled ratio uninformative; on ABB, shuffled patching captures 49% of the matched-token rescue. SI heads thus encode rule information in a largely token-abstract form that is stable across shot counts; the residual matched-prompt advantage in Gemma and Qwen suggests an additional token-specific component layered on the abstract one.

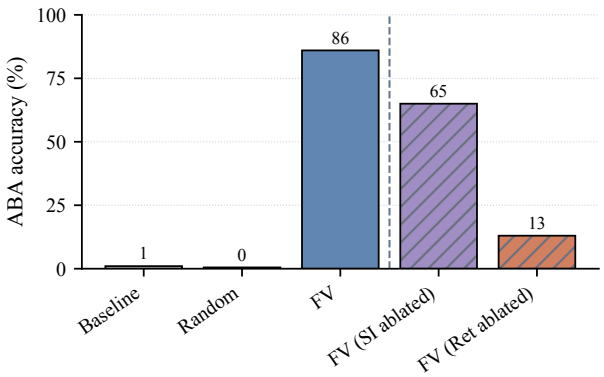


Figure 3. **Function-vector rescue at 0-shot, ABA.**

4. Function Vectors as Circuit Signal

If examples amplify signal through a latent circuit, function vectors (Todd et al., 2024) should be readable as one instance of that signal. Yang et al. (2025b) report that FV-producing heads and SI heads largely overlap (r=0.82) and conjecture FVs implement symbolic induction. We test this causally.

Following Todd et al. (2024), we extract $z\,W_O$ at the last query position from correct 10-shot Gemma 2-2B passes, average across prompts, and sum across the 29 permutation-test-significant heads at 10-shot to obtain one vector per rule. We add $s \cdot \mathrm{FV}_r$ to the residual stream at layer 12, last query position, of held-out 0-shot prompts, then re-run with the 7 SI heads or 9 Ret heads zero-ablated at the query position. We use s=25 and layer 12, where rescue plateaus for both rules (full sweep in Appendix A.2.6).

FV injection rescues 0-shot accuracy from 1% to 86% on ABA (Figure 3) and from 11% to 89% on ABB, with norm-matched random vectors collapsing accuracy to 0.5%, ruling out generic perturbation as the source of rescue. Ret ablation at the query position drops the rescue to 13% (ABA) and 29% (ABB); the rule signal is supplied but cannot be read out. SI ablation is rule-asymmetric: it reduces the ABA rescue to 65% but leaves ABB unchanged at 89%. The FV appears to fully substitute for SI's output on ABB and partially on ABA, while Ret's role as readout cannot be bypassed by injection upstream of it.

5. Discussion

Our results suggest that in-context learning of abstract rules operates predominantly through signal amplification within a latent circuit. The same three-stage topology is identifiable from 1-shot, when accuracy is just 17%, its core heads persist across shot counts, and cross-shot patching confirms the circuit is functional when supplied with sufficient input signal. The FV result reinforces this from another angle: a single static vector roughly substitutes for the SI stage at 0-shot, but only when Ret is available to read it out. On this view, demonstrations are better understood as input to a latent computation than as specifications for one constructed in the forward pass.

The partial token-invariance of SI patches and the success of FV injection both connect to prior work on task, function, and concept vectors (Hendel et al., 2023; Todd et al., 2024; Opiełka et al., 2026), which show that ICL task identity can be captured in transferable activation patterns. Our FV result sharpens the relationship: the vector is not merely correlated with SI heads, as observed by Yang et al., but causally nearly interchangeable with that stage's output, contingent on Ret remaining available to consume it. This frames such transferable representations as readouts of pre-existing computational pathways more so than freestanding mechanisms. The same logic extends naturally to activation steering (Zou et al., 2023), where fixed perturbations reliably shift behavior.

Several limitations apply. Contrastive CMA isolates only rule-discriminating components, potentially missing early-layer infrastructure heads that support both rules and likely explains the residual gap between patched (88%) and native 10-shot (99%) accuracy. The task, while requiring generalization over arbitrary tokens, has a simple relational structure (essentially a 3-column table with a copy operation); the amplification pattern also holds on letter-string analogies (which involve no copying), but whether it generalizes to richer abstract reasoning is open.

A broader implication: if in-context learning of this rule operates by amplifying signal through a pre-existing circuit, then understanding ICL behavior reduces in part to understanding what circuits exist in the weights and what supplies their input. Open questions about what makes a task learnable in-context, why some interventions transfer while others don't, and why context is sometimes used poorly even when present then become questions about latent circuit inventory and accessibility.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Akyürek, E., Wang, B., Kim, Y., and Andreas, J. In-context language learning: Architectures and algorithms. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 787–812. PMLR, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hendel, R., Geva, M., and Globerson, A. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Liu, S., Ye, H., Xing, L., and Zou, J. Y. In-context vectors: Making in context learning more effective and controllable through latent space steering, 2024. Published at the 41st International Conference on Machine Learning (ICML), PMLR 235:32287–32307.

Marcus, G. F., Vijayan, S., Bandi Rao, S., and Vishton, P. M. Rule learning by seven-month-old infants. *Science*, 283 (5398):77–80, 1999. doi: 10.1126/science.283.5398.77.

Musker, S., Duchnowski, A., Millière, R., and Pavlick, E. LLMs as models for analogical reasoning. *Journal of Memory and Language*, 145:104676, 2025. doi: 10.1016/j.jml.2025.104676.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

Opiełka, G., Rosenbusch, H., and Stevenson, C. E. Causality $\neq$ invariance: Function and concept vectors in LLMs. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023.

Webb, T., Holyoak, K. J., and Lu, H. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7:1526–1541, 2023. doi: 10.1038/s41562-023-01659-w.

Webber, W., Moffat, A., and Zobel, J. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems*, 28(4):20:1–20:38, 2010. doi: 10.1145/1852102.1852106.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Yang, Y., Campbell, D., Huang, K., Wang, M., Cohen, J., and Webb, T. Emergent symbolic mechanisms support abstract reasoning in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70515–70549. PMLR, 2025b.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. Representation engineering: A top-down approach to AI transparency. 2023. URL https://arxiv.org/abs/2310.01405.

A. Appendix

A.1. Code and Hardware

All code is written in Python using TransformerLens and the HuggingFace Transformers library. Gemma 2-2B experiments were run on a 36GB Apple Silicon MacBook Pro (MPS); Llama 3.1-8B and Qwen 3-4B experiments were run on a single NVIDIA A40 GPU. All model weights are loaded in the `bfloat16` format. Code and bundled paper results are available at https://github.com/melissawessel/icl-symbolic.

A.2. Experimental Details

A.2.1. Models

Model	Layers	Q heads	KV heads	d_{model}	d_{head}	BOS
Gemma 2-2B (Gemma Team, 2024)	26	8	4	2304	256	yes
Llama 3.1-8B (Grattafiori et al., 2024)	32	32	8	4096	128	yes
Qwen 3-4B (Yang et al., 2025a)	36	32	8	2560	128	no

Table 1. Model architectures. All three use grouped-query attention (GQA). All experiments use `bfloat16` precision and load the models via TransformerLens, which materializes one W_K/W_V per query head so attention-head indexing is uniform across architectures.

A.2.2. Prompts

Following Yang et al. (2025b), tokens are sampled from a 72K-entry English-token vocabulary released with their codebase, restricted at runtime to entries that (a) encode to a single token under the model's tokenizer and (b) do not merge with the separator `^`. Each in-context example occupies six tokens (`A ^ B ^ C \n`) and the query occupies four (`A ^ B ^`). With BOS offset $b \in \{0, 1\}$, example $i \in \{0, \ldots, n-1\}$ has its C position at $b + 6i + 4$ and the final query position lies at $b + 6n + 3$. Total prompt length is $b + 6n + 4$ tokens. Each prompt is validated to tokenize to exactly this length before use.

Accuracy is evaluated on N=500 prompts per rule per shot count (1000 total per shot count). CMA uses N=200 matched context pairs per condition (100 per rule direction). Cross-shot patching uses N=200 matched donor/target pairs per rule. Random seeds are unique per (model, n, rule) so token identity does not co-vary with shot count.

Matched pairs are constructed as follows. For the *abstract* CMA condition, a base ABA prompt $a_1\, b_1\, a_1 \mid \ldots \mid a_q\, b_q\, ?$ is paired with an exp prompt that swaps the first two tokens of every example and the query ($b_1\, a_1\, a_1 \mid \ldots \mid b_q\, a_q\, ?$). Both prompts share their final tokens but differ in which abstract role each token plays; the same construction inverted for ABB. For the *token* condition the rule is held fixed and the query tokens are swapped, isolating the literal-token component. For cross-shot patching, the target's n_t examples are the *last* n_t examples of the matched n_d-shot donor (same tokens, same rule). The 0-shot triplet construction pairs an empty-context query `A^B^` with two 10-shot donors (one ABA, one ABB) sharing the query tokens but with otherwise disjoint demonstrations.

A.2.3. Causal Mediation Analysis

CMA follows Yang et al. (2025b), Algorithm 1. For each context pair (base c_2, exp c_1) and each attention head (ℓ, h), we run the base context forward, cache the head's per-token output `hook_z` at the patch position, and run the exp context with that head's activation overwritten at the patch position. The score is the change in logit difference,

$$\text{CMA}(\ell, h) \;=\; \big[\, \mathcal{L}(c_1^*; a_{\text{causal}}, a_{\text{exp}}) \,\big] - \big[\, \mathcal{L}(c_1; a_{\text{causal}}, a_{\text{exp}}) \,\big],$$

where $\mathcal{L}(c; a_1, a_2) = \text{logit}(a_1 \mid c) - \text{logit}(a_2 \mid c)$, c_1^* denotes the patched exp context, a_{causal} is the answer the patched model should predict if the head carries the rule, and a_{exp} is the answer of the unpatched exp context. Per-pair scores are stored as $[N_{\text{valid}}, n_{\text{layers}}, n_{\text{heads}}]$ and averaged across pairs and across the two rule directions for ranking and significance.

We use two pair-validity filters: `threshold` (keep pairs where both contexts assign $\Pr(\text{answer}) \geq 0.9$, used by Yang et al. on high-accuracy models) and `correct` (keep pairs where the argmax matches the answer). Below 4-shot, Gemma and Qwen rarely clear the 0.9 probability threshold even on prompts they answer correctly, so we use the `correct` filter at all shot counts to keep the methodology comparable across models and shot counts. The number of valid pairs after filtering is reported alongside each result.

Stage	Patch position(s)	CMA condition
Symbolic Abstraction (SA)	all C positions: $\{b + 6i + 4 : 0 \leq i < n\}$	abstract
Symbolic Induction (SI)	final query position: $b + 6n + 3$	abstract
Retrieval (Ret)	final query position: $b + 6n + 3$	token

Table 2. Patch positions per stage. SA writes to the C tokens of the demonstrations (so no SA evaluation is possible at n=0). SI and Retrieval write to the same query position but are distinguished by the contrastive condition.

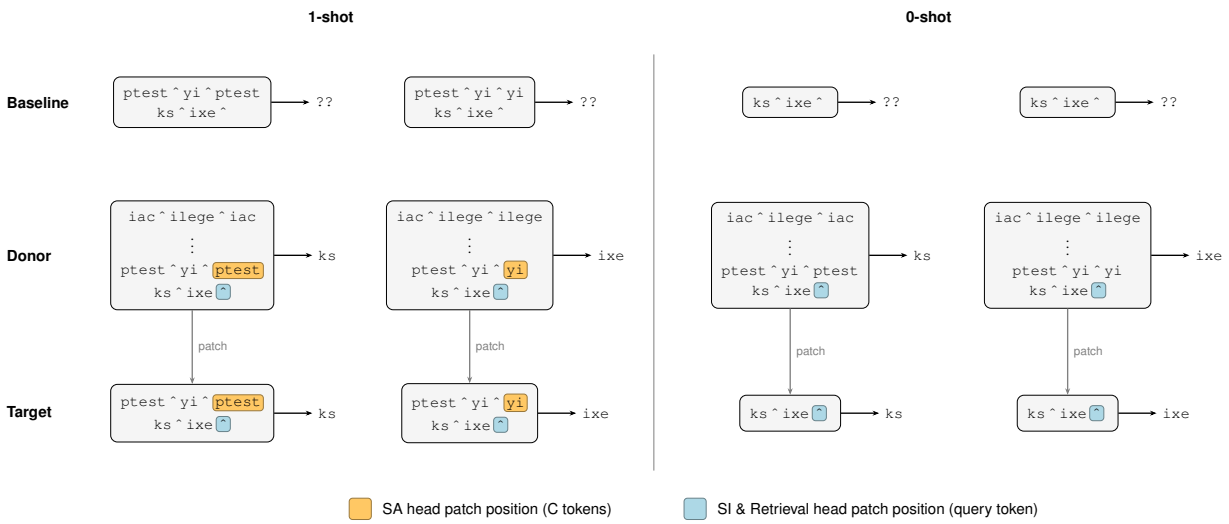


Figure 4. **Cross-shot activation patching.** Activations from 10-shot donors (middle) are patched into 1-shot (left) and 0-shot (right) targets (bottom), steering outputs to the correct rule-following completion. Yellow highlights indicate the SA head patch position (C tokens); blue indicates the SI and Retrieval head patch position (query token).

A.2.4. Activation patching

All patches operate on `hook_z`, the per-query-head attention output before the per-head W_O projection. Patching at this point is per-Q-head rather than per-KV-head, which avoids the GQA complication that arises when patching at W_V/W_K; TransformerLens materializes one W_K, W_V copy per query head so each query head can be patched independently. Patches replace (not add) the activation: for the patched head at the patched position, the donor's cached `hook_z` slice is written into the target's forward pass. All other heads, all other positions, and all MLPs run unmodified. For multi-head patches each head is replaced at its stage-specific position(s).

Two negative controls are used throughout. *Shuffled-token donor:* a donor prompt with the same rule and shot count as the matched donor but a disjoint random token sample. The same heads are patched at the same positions; any rescue that survives must be carried by token-invariant content. *Random non-circuit heads:* a count-matched random sample of attention heads drawn from outside the union of CMA-significant heads, patched at the same positions. The matched-donor rescue must beat both controls for the effect to be attributed to the circuit.

A.2.5. Significance

Per-head significance uses a max-statistic permutation test that controls the family-wise error rate across all (ℓ, h) positions following Yang et al. (2025b), Appendix B.2. For each of 5000 iterations we sign-flip each pair's score independently with probability 0.5 (equivalent to permuting the base/exp assignment under the null), recompute the per-head mean, and record the maximum across (ℓ, h). The significance threshold is the 95th percentile of this null distribution; heads whose observed mean exceeds the threshold are reported as significant at α=0.05.

A.2.6. Function Vectors

The FV is constructed Todd-style: for the union $\mathcal{H}$ of permutation-significant heads at 10-shot (across both rules; $|\mathcal{H}|$=29 for Gemma 2-2B), $\mathrm{FV}_r = \sum_{(\ell,h)\in\mathcal{H}} \mathbb{E}_{c\in\mathcal{C}_r}\left[z_q^{(\ell,h)}(c)\, W_O^{(\ell,h)}\right]$, where $\mathcal{C}_r$ is the set of model-correct 10-shot prompts for

rule r and q is the last query position. Injection adds $s \cdot \mathrm{FV}_r$ to `resid_pre` at layer L, last query position, of held-out 0-shot prompts; ablation zeroes `hook_z` for SI or Ret heads at the query position.

We swept $s \in \{1, 5, 10, 25, 50, 100\}$ and $L \in \{6, 8, 12, 14, 16\}$ (200 0-shot prompts per rule, seed 42). Each experiment draws its own 200-prompt evaluation set, so the 0-shot ABB baseline reported here (11.0%) differs from the cross-shot patching baseline (7.5%, §3.2) by ∼3 pp; this is within the binomial CI half-width (≈ 4 pp at n=200, p≈0.1). Rescue saturates around s=25 across all five layers and collapses at s=100 (e.g. ABA at L=12: 2.0%, 22.0%, 55.0%, 86.0%, 84.5%, 21.5% for $s = 1, 5, 10, 25, 50, 100$). Per-rule peaks are at (L=8, s=50) for ABA (88.0%) and (L=14, s=25) for ABB (92.0%); we report (L=12, s=25), near-peak for both rules (ABA 86.0%, ABB 88.5%), so the ablation factorial uses one (L, s) across rules. The full ablation factorial is in Table 3.

Condition	ABA	ABB
Baseline (no inject)	1.0	11.0
FV inject	86.0	88.5
FV inject + ablate SI	65.0	89.0
FV inject + ablate Ret	13.0	28.5
Random vector (norm-matched)	0.0	0.5

Table 3. FV × per-stage ablation factorial at L=12, s=25. Accuracy (%), n=200 per cell.

A.3. Per-Model CMA Results

A.3.1. Llama 3.1-8B

The same three-stage topology recovered in Gemma is visible in Llama 3.1-8B (Figure 5); persistent cores and RBO are reported in Section A.3.3.

A.3.2. Qwen 3-4B

Qwen 3-4B (Figure 6) shows the same topology with more reranking among the leading heads, especially at low shot counts and in SA, with some apparent diffusion of causal impact across heads at higher shot counts.

A.3.3. Top-K heads, persistent cores, and rank-biased overlap

For each model and stage we average per-pair CMA scores over the 200 context pairs and over the two rule directions (ABA, ABB) to obtain a single mean CMA score per attention head, then rank all heads by descending score. Tables 4–6 summarize the result.

Top-5 heads per stage and shot count. Table 4 lists the top-5 heads at each shot count, with their mean CMA score in parentheses. The pattern is similar across models: the same heads dominate from 1-shot through 10-shot, and changes are dominated by score growth rather than identity.

Persistent cores. The *persistent core* at top-K is the set of heads that appear in the top-K at every shot count $n \in \{1, 2, 4, 10\}$. Table 5 reports the size of the persistent core (and the heads themselves at top-5) for each model and stage. Random-sample expected top-K overlap is $K^2/|H|$ where $|H|$ is the total head count: 0.12 for Gemma top-5, 0.024 for Llama, 0.022 for Qwen.

Rank-Biased Overlap. RBO is a top-weighted similarity measure for ranked lists (Webber et al., 2010) that downweights tail rankings, addressing the fact that ∼90% of heads have near-zero CMA scores whose rank order is dominated by noise. We use the extrapolated form,

$$\mathrm{RBO}_{\mathrm{ext}}(p) \;=\; (1-p)\sum_{d=1}^{D} \frac{X_d}{d}\, p^{\,d-1} \;+\; p^{D}\,\frac{X_D}{D},$$

where X_d is the size of the intersection of the two top-d prefixes and $D = |H|$ is the full ranking depth, with persistence $p = 0.9$ (effective weight ≈ 86% on the top 10 ranks). RBO ranges from 0 (disjoint) to 1 (identical). Table 6 reports RBO across all pairs of shot counts.

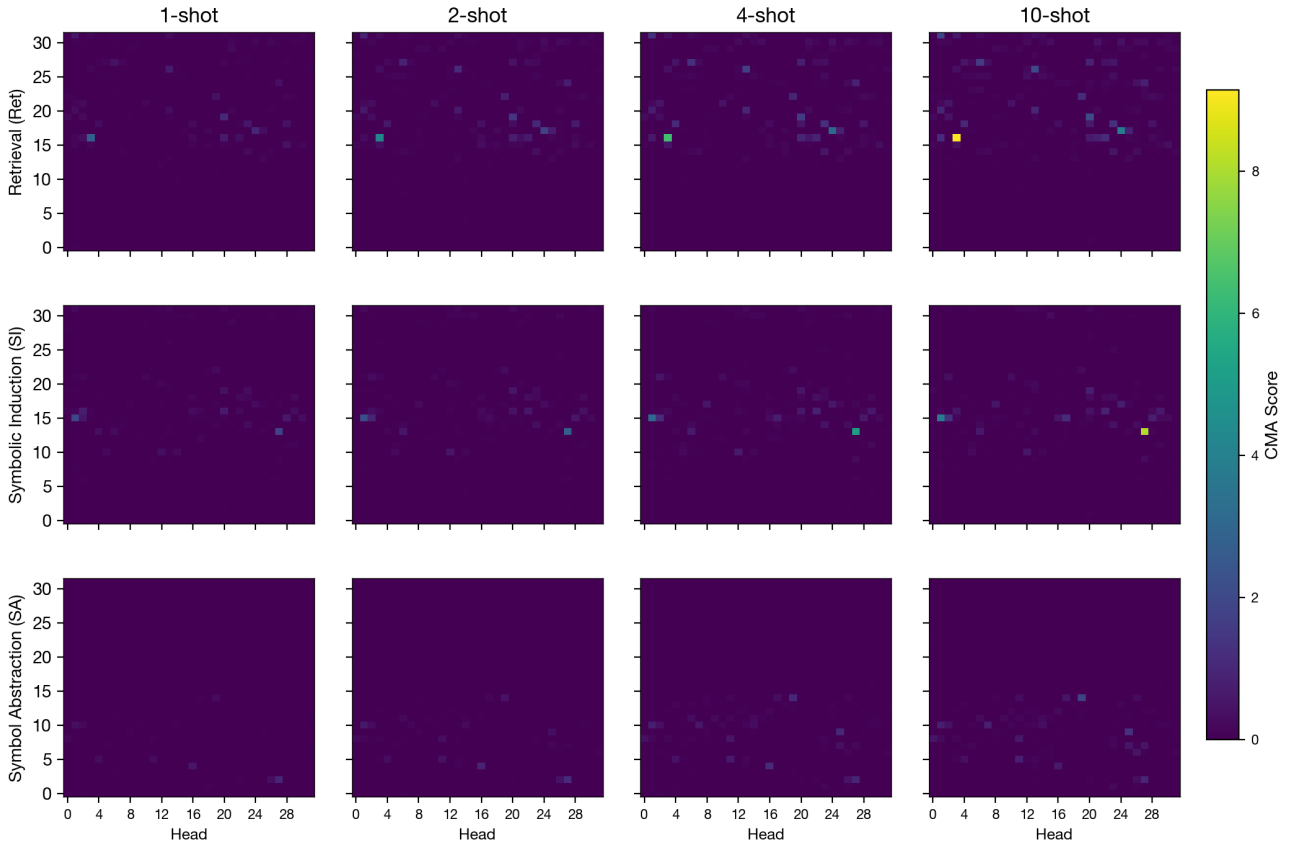


Figure 5. **The Symbolic Reasoning Circuit in Llama 3.1-8B at 1 through 10-Shot.** CMA Scores by layer by head, by circuit stage across shot counts.

The pattern is consistent across the three models. RBO is high among shot counts where accuracy is non-trivial (e.g., $\geq$ 4-shot for Gemma): adjacent-shot RBO sits at 0.81–0.94 for SI/Ret in Gemma and Llama, and at 0.68–0.94 for SI/Ret in Qwen. The lowest RBO values appear at 1-shot for the two models with the weakest 1-shot accuracy (Qwen at 27% ABA, Gemma at 17% ABA), and within those, primarily in the SA stage where the absolute CMA signal is also smallest, consistent with the SA stage being noisier when there are fewer in-context examples to abstract from. Even in those low-signal regimes the persistent-core counts and top-5 overlaps remain well above any random baseline.

Bootstrap resampling over the CMA pairs (1000 iterations, percentile 95% CIs) confirms the RBO point estimates in Table 6 are tight, e.g. Gemma SI $1 \leftrightarrow 10$: 0.72 $[0.65, 0.75]$; Qwen SI $1 \leftrightarrow 10$: 0.45 $[0.40, 0.51]$. A permutation test against independent head rankings (5000 iterations) places every observed RBO at $p < 0.001$.

A.4. Extended Patching Results

See Figure 7, 8, 9, and Table 7.

A.5. Letter-String Analogies

To check that the topology + amplification pattern is not specific to identity rules, we replicate the CMA analysis on the letter-string analogy task of Webb et al. (2023), also used in Yang et al. (2025b). Two rules: *successor* (`[a b c] [a b d]`, last letter $+1$) and *predecessor* (`[b c d] [a c d]`, first letter -1). A prompt is scored correct only when all three answer letters are the argmax. We evaluate 500 prompts per rule per shot count.

Accuracy. Successor is largely solved by 2-shot in both models. Predecessor is harder and mirrors the ABA/ABB asymmetry. Per-letter inspection shows the transformed letter is the bottleneck (Gemma 2-shot predecessor letter 1: 23.6%;

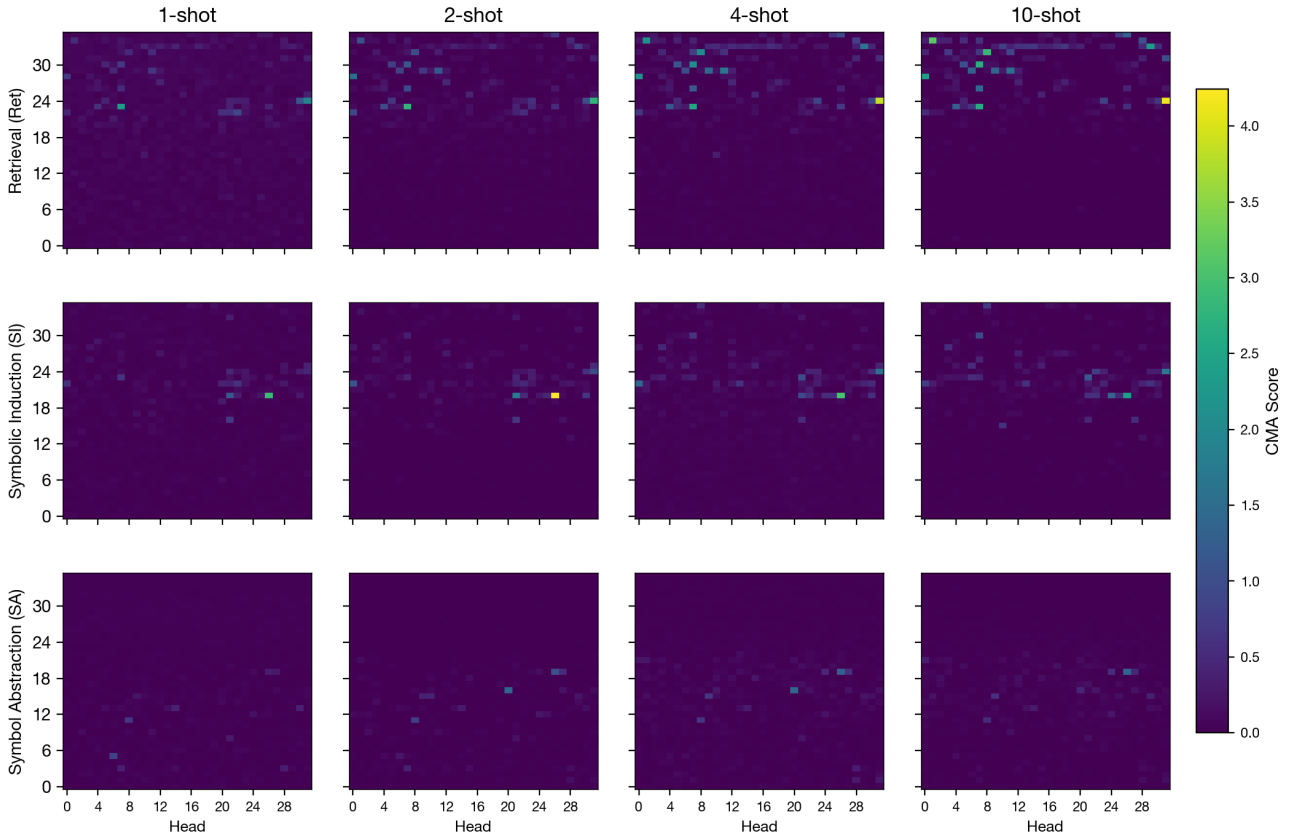


Figure 6. **The Symbolic Reasoning Circuit in Qwen 3-4B at 1 through 10-Shot.** CMA Scores by layer by head, by circuit stage across shot counts.

letter 3, unchanged: 100%).

CMA design. SA and SI contrasts pair successor and predecessor prompts with identical starting letters; Ret contrasts pair same-rule prompts with different query letters. SA is patched at the output letters of in-context examples, SI and Ret at the positions preceding answer letters. Logit shifts are summed over rule-discriminating positions (letters 1 and 3 for SA/SI, all three for Ret). Filtered to correctly answered prompts, 200 pairs per stage per condition.

Results. The same three-stage topology appears here (Figure 10), but cross-task overlap of dominant heads is partial: L14H0 (SI) and L18H6 (Ret) lead in both task families and at every shot count, whereas the SA stage uses different heads across the two tasks. In LSA, the most persistent SA head is L13H0 (top-3 at 1, 2, and 10 shot), with L8H2 emerging as the SA leader by 10-shot ($0.05 \rightarrow 0.59 \rightarrow 2.39$). Amplification replicates on all three stages of the cross-task heads: L14H0 (SI) grows $0.81 \rightarrow 2.20 \rightarrow 4.11$ (1/2/10-shot, $5.1\times$) and L18H6 (Ret) grows $10.6 \rightarrow 11.0 \rightarrow 17.3$ ($1.6\times$) and L13H0 (SA) $0.35 \rightarrow 0.62 \rightarrow 0.78$ ($2.2\times$).

Caveats. Predecessor abstract pairs are sparse at low shot: at 1-shot the predecessor Ret condition has zero valid pairs and is omitted, and at 2-shot only 4 pairs survive. Successor and 10-shot results are unaffected. We did not run LSA CMA on Llama or Qwen, nor extend cross-shot patching to LSA; this section is a generalization check on the CMA topology claim, not a full replication.

Stage	Shot	Top-5 (CMA score)				
Gemma 2-2B						
SA	1	**L6H1** (0.50)	**L8H1** (0.45)	**L6H2** (0.38)	L3H1 (0.14)	L3H0 (0.14)
	2	**L6H1** (1.05)	**L8H1** (0.88)	**L6H2** (0.63)	L10H4 (0.47)	L3H1 (0.22)
	4	**L6H1** (1.73)	**L8H1** (1.33)	L10H4 (1.01)	**L6H2** (0.87)	L12H3 (0.58)
	10	**L6H1** (2.07)	**L8H1** (2.03)	L10H4 (1.72)	L12H3 (1.30)	**L6H2** (1.22)
SI	1	**L14H0** (1.40)	**L12H1** (0.73)	**L15H0** (0.61)	**L14H1** (0.31)	L13H0 (0.25)
	2	**L14H0** (2.73)	**L12H1** (1.58)	**L15H0** (0.90)	**L14H1** (0.76)	L17H3 (0.41)
	4	**L14H0** (5.78)	**L12H1** (2.85)	**L14H1** (1.67)	**L15H0** (1.22)	L17H3 (0.99)
	10	**L14H0** (9.02)	**L12H1** (3.66)	**L14H1** (2.71)	L17H3 (1.83)	**L15H0** (1.50)
Ret	1	**L18H6** (0.82)	**L22H4** (0.70)	L17H0 (0.66)	L25H4 (0.41)	**L20H7** (0.38)
	2	**L18H6** (2.48)	**L22H4** (1.63)	L17H0 (1.04)	**L20H7** (0.96)	L19H1 (0.70)
	4	**L18H6** (3.94)	**L22H4** (2.55)	**L20H7** (1.49)	L17H0 (1.27)	L19H1 (1.25)
	10	**L18H6** (6.73)	**L22H4** (4.86)	L19H1 (2.38)	**L20H7** (2.22)	L25H4 (1.69)
Llama 3.1-8B						
SA	1	**L2H27** (0.63)	L4H16 (0.44)	L5H11 (0.26)	**L14H19** (0.25)	**L10H1** (0.24)
	2	**L2H27** (0.76)	L4H16 (0.58)	**L10H1** (0.52)	**L14H19** (0.42)	L5H11 (0.35)
	4	L9H25 (0.86)	**L14H19** (0.86)	**L2H27** (0.85)	**L10H1** (0.75)	L4H16 (0.66)
	10	**L14H19** (1.30)	L9H25 (1.13)	**L2H27** (0.80)	**L10H1** (0.74)	L5H11 (0.55)
SI	1	**L15H1** (1.81)	**L13H27** (1.47)	**L15H2** (0.43)	**L15H28** (0.41)	L10H12 (0.34)
	2	**L13H27** (2.55)	**L15H1** (2.04)	**L15H2** (0.68)	**L15H28** (0.45)	L13H6 (0.43)
	4	**L13H27** (4.60)	**L15H1** (2.71)	**L15H2** (0.83)	**L15H28** (0.60)	L15H17 (0.54)
	10	**L13H27** (6.39)	**L15H1** (3.03)	**L15H2** (0.96)	L15H17 (0.85)	**L15H28** (0.74)
Ret	1	**L16H3** (2.58)	**L19H20** (0.84)	**L17H24** (0.55)	**L26H13** (0.47)	L31H1 (0.37)
	2	**L16H3** (3.75)	**L17H24** (1.37)	**L19H20** (1.28)	**L26H13** (0.99)	L27H6 (0.79)
	4	**L16H3** (5.42)	**L17H24** (2.52)	**L26H13** (1.61)	**L19H20** (1.42)	L27H6 (1.24)
	10	**L16H3** (7.47)	**L17H24** (2.98)	**L26H13** (1.87)	L31H1 (1.84)	**L19H20** (1.60)
Qwen 3-4B						
SA	1	L5H6 (0.50)	L11H8 (0.39)	L13H14 (0.27)	L3H28 (0.26)	**L19H26** (0.25)
	2	L16H20 (1.05)	**L19H26** (0.93)	L11H8 (0.52)	L19H27 (0.37)	L15H9 (0.34)
	4	**L19H26** (1.23)	L16H20 (0.92)	L15H9 (0.48)	L19H27 (0.46)	L19H24 (0.43)
	10	**L19H26** (1.12)	L19H24 (0.51)	L15H9 (0.41)	L19H27 (0.38)	L16H20 (0.31)
SI	1	**L20H26** (2.20)	L20H21 (0.89)	L16H21 (0.62)	L22H0 (0.35)	L20H22 (0.27)
	2	**L20H26** (3.51)	L20H21 (1.36)	L16H21 (0.65)	L22H0 (0.57)	L20H30 (0.54)
	4	**L20H26** (2.59)	L20H21 (0.86)	L22H0 (0.83)	L24H31 (0.80)	L23H21 (0.75)
	10	**L20H26** (1.67)	L20H24 (1.09)	L24H31 (0.93)	L23H21 (0.82)	L24H22 (0.58)
Ret	1	**L23H7** (2.09)	**L24H31** (1.19)	L29H11 (0.64)	**L28H0** (0.55)	L24H30 (0.52)
	2	**L23H7** (2.71)	**L24H31** (2.18)	**L28H0** (1.35)	L29H11 (1.15)	L34H1 (1.02)
	4	**L24H31** (3.26)	**L23H7** (2.53)	L34H1 (2.12)	**L28H0** (2.04)	L32H8 (1.74)
	10	**L24H31** (3.69)	L34H1 (2.91)	L32H8 (2.48)	**L23H7** (2.44)	**L28H0** (2.23)

Table 4. Top-5 attention heads per stage and shot count, ranked left-to-right by mean CMA score (in parentheses; averaged across pairs and across ABA/ABB rule directions). **Bold** marks persistent-core heads (top-5 at every shot count). Reading down a column shows reranking among the leading heads.

Stage	Top-5 core size	Top-10 core size	Top-5 core members
Gemma 2-2B			
SA	3	6	L6H1, L6H2, L8H1
SI	4	6	L12H1, L14H0, L14H1, L15H0
Ret	3	7	L18H6, L20H7, L22H4
Llama 3.1-8B			
SA	3	5	L2H27, L10H1, L14H19
SI	4	6	L13H27, L15H1, L15H2, L15H28
Ret	4	6	L16H3, L17H24, L19H20, L26H13
Qwen 3-4B			
SA	1	5	L19H26
SI	1	4	L20H26
Ret	3	7	L23H7, L24H31, L28H0

Table 5. Persistent cores: heads in the top-K at every shot count $n \in \{1, 2, 4, 10\}$. Qwen's top-5 is noisier at low shot counts but recovers at top-10.

	1↔2	1↔4	1↔10	2↔4	2↔10	4↔10
Gemma 2-2B						
SA	0.77	0.71	0.66	0.86	0.79	0.91
SI	0.81	0.77	0.72	0.86	0.81	0.91
Ret	0.84	0.79	0.78	0.94	0.86	0.87
Llama 3.1-8B						
SA	0.85	0.58	0.54	0.67	0.62	0.81
SI	0.74	0.70	0.65	0.89	0.84	0.92
Ret	0.85	0.79	0.76	0.90	0.83	0.90
Qwen 3-4B						
SA	0.41	0.28	0.29	0.67	0.55	0.80
SI	0.84	0.68	0.45	0.80	0.55	0.68
Ret	0.85	0.69	0.57	0.77	0.64	0.83

Table 6. Rank-Biased Overlap (RBO, $p = 0.9$, full-depth extrapolated) between per-head CMA rankings at every pair of shot counts. Each ranking averages across the 200 CMA pairs and across the ABA and ABB rule directions. Higher is more similar; for full rankings of unrelated lists RBO would tend to $\approx 1 - p = 0.1$.

Shot	Rule	Baseline	SA-only	SI-only	SI-shuffled	Ret-only	All stages	Random
Gemma 2-2B								
0	ABA	1.0	1.0	31.0	23.0	17.5	**56.0**	0.5
0	ABB	7.5	7.5	41.5	36.5	42.5	**72.5**	8.5
1	ABA	16.5	19.5	70.5	54.0	67.0	**87.5**	9.0
1	ABB	57.0	57.5	84.0	78.0	81.5	**93.0**	47.0
Llama 3.1-8B								
0	ABA	2.5	2.5	35.5	38.5	48.0	**72.0**	5.0
0	ABB	1.0	1.0	37.0	38.0	45.0	**84.5**	3.5
1	ABA	59.5	82.0	83.0	83.0	84.5	**94.5**	66.5
1	ABB	66.5	87.0	82.5	81.0	85.0	**97.5**	72.5
Qwen 3-4B								
0	ABA	0.0	0.0	6.5	1.0	15.5	**44.0**	0.0
0	ABB	2.5	2.5	51.5	26.5	60.0	**85.0**	3.5
1	ABA	27.0	57.0	82.5	56.0	86.0	**95.5**	30.0
1	ABB	55.0	72.5	86.0	68.5	89.5	**97.0**	59.5

Table 7. **Cross-shot patching accuracy (%).** Patching from a 10-shot donor into a 0/1-shot target, by stage. SA-only, SI-only, and Ret-only patch only the heads at that stage's positions; *All stages* patches the union of all three. *SI-shuffled* uses a donor with the same rule but entirely disjoint tokens, patched at SI heads only. *Random* patches a matched-count random selection of non-circuit heads. n=200 prompts per cell. SA-only is identical to baseline at 0-shot because there are no in-context C positions to patch. Bold marks the All-stages column. Mean answer probability p tracks accuracy and is omitted for compactness.

Model	Shot	Succ.	Pred.
Gemma 2-2B	2	89.0%	17.6%
Gemma 2-2B	10	99.8%	76.4%
Llama 3.1-8B	2	98.4%	64.4%
Llama 3.1-8B	10	100%	100%

Table 8. LSA accuracy (all three answer letters correct), 500 prompts per rule.

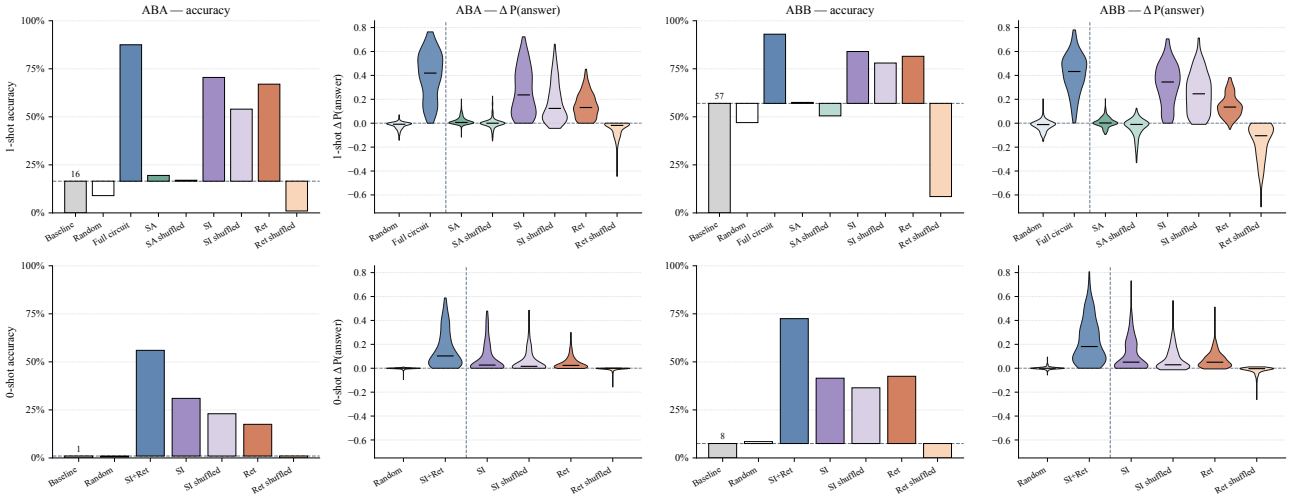


Figure 7. **Cross-shot patching, Gemma 2-2B.** Rows: 1-shot, 0-shot targets. Columns: ABA, ABB. Dashed line: baseline.

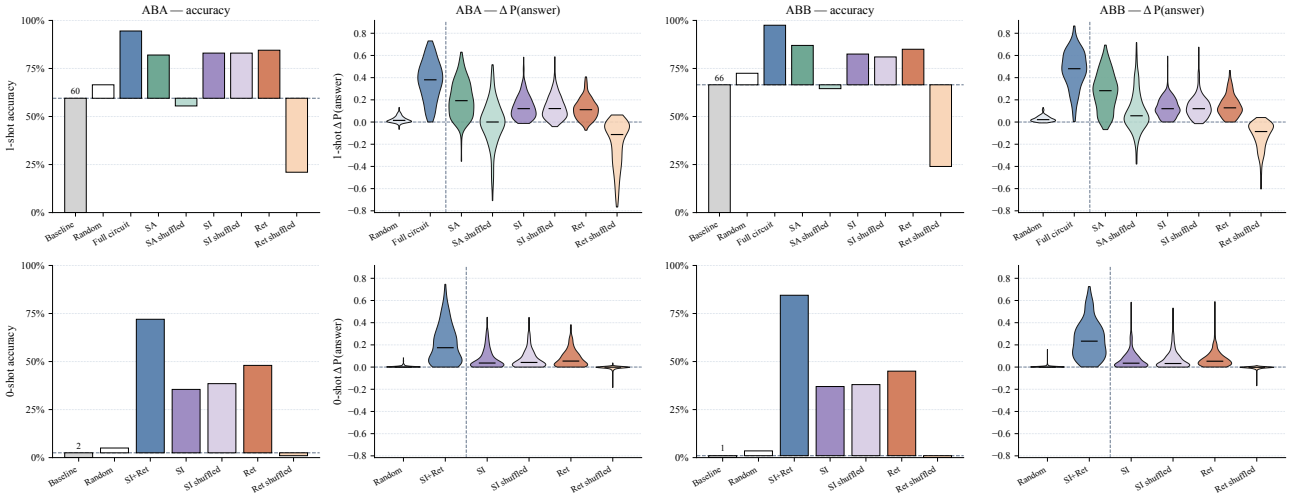


Figure 8. **Cross-shot patching, Llama 3.1-8B.** Rows: 1-shot, 0-shot targets. Columns: ABA, ABB. Dashed line: baseline.

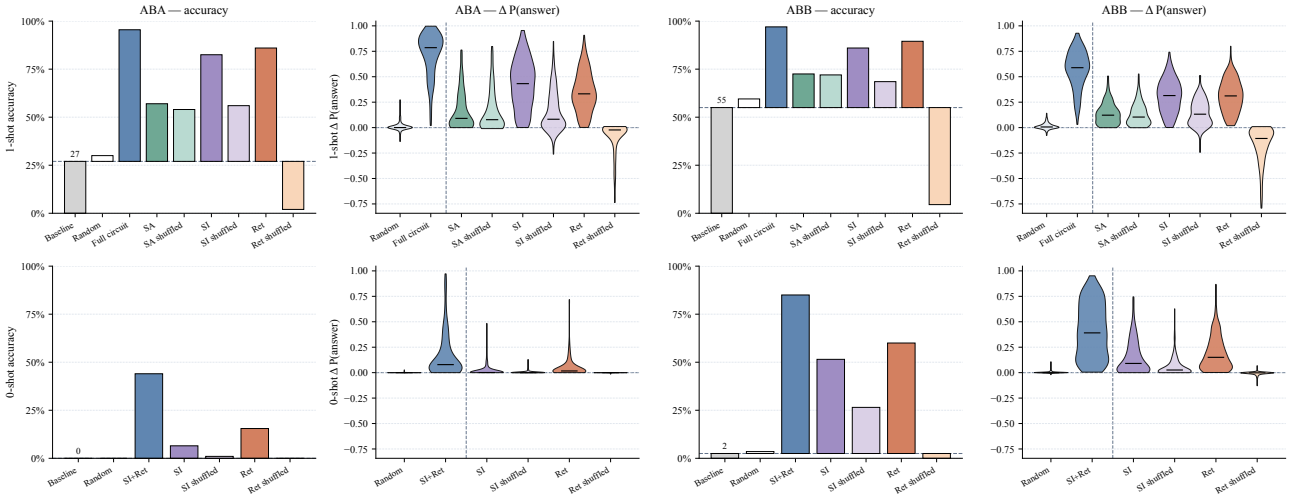


Figure 9. **Cross-shot patching, Qwen 3-4B.** Rows: 1-shot, 0-shot targets. Columns: ABA, ABB. Dashed line: baseline.

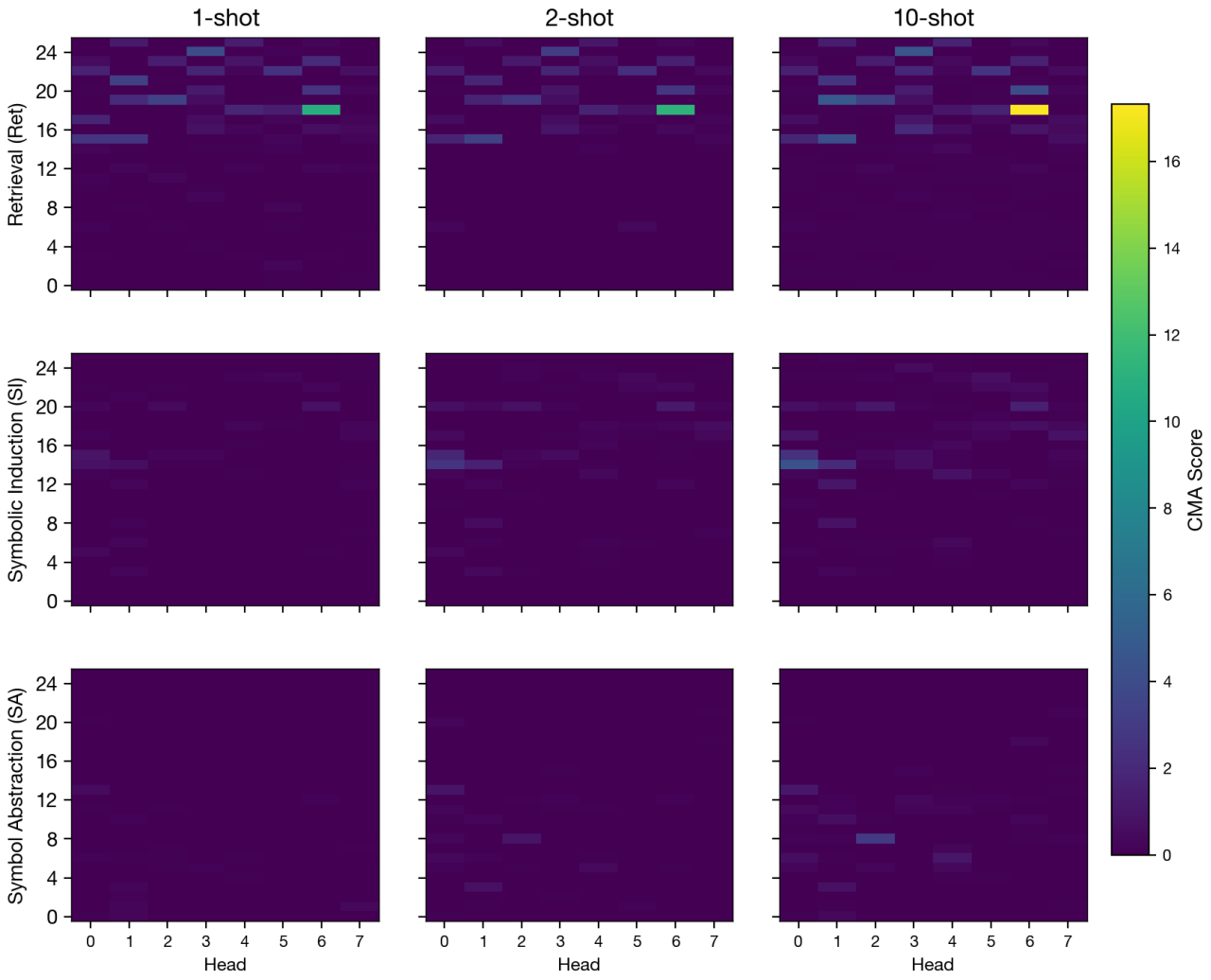


Figure 10. **LSA CMA heatmaps, Gemma 2-2B** at 1, 2, and 10 shot, by stage. Same three-stage topology as ABA/ABB; L14H0 (SI) and L18H6 (Ret) dominate at all shot counts.