PROBING THE EMERGENT GLOBAL WORKSPACE OF REASONING MODELS ON STRUCTURED REASONING TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Global workspace theory has become an influential lens on artificial intelligence. We ask whether a workspace-like channel, a limited-capacity state that later computation repeatedly reads and rewrites, arises on its own in reasoning models that were never given one, and if so, where it lives and what it carries. Recursive reasoning models, which refine a latent state rather than extend a token sequence, offer a revealing case, because the state that would have to be broadcast is explicit and reentrant by construction. We use verifiable, multi-solution mathematical reasoning tasks, where a single intervention can separate *whether* a model produces a valid solution from *which* one it produces, and fit Jacobian lenses at corpus, per-question, and per-solution levels to a recursive latent reasoner and an autoregressive LLM. In the recursive reasoner, a subspace amounting to a fraction of a percent of the latent state is causally necessary across most of the trajectory, the recursion repairs single-step damage so that only sustained removal is fatal, and ablating branch-conditioned directions changes *which* solution is returned while largely sparing *whether* one is. The autoregressive model instead confines constraint enforcement to a narrow band of late decoder layers, where a single edit destroys validity and identity together. In both, the workspace is a functional *role* rather than a dedicated module, a thin channel inside the shared state; the recursive model spreads it over reasoning steps while the autoregressive model compresses it into a few adjacent layers, suggesting why recursive refinement can be more robust than autoregressive decoding. The protocol needs only an accessible internal state and a verifier of explicit constraints, and transfers to other models and tasks.

1 INTRODUCTION

Scientific reasoning is an unusually tractable lens on the reasoning mechanisms of large models: solutions are verifiable, difficulty is controllable, and the constraints a correct answer must satisfy are explicit. Yet while large reasoning models now solve non-trivial mathematical and scientific problems, how they arrive at those solutions internally—and which internal structures are load-bearing— remains largely uncharacterized. Closing this gap is a prerequisite for improving reasoning by design rather than by scale.

We approach the question through global workspace theory (GWT) (Baars, 1988; 2005; Dehaene et al., 1998; Mashour et al., 2020), which holds that flexible cognition arises when specialized modules compete to broadcast information through a shared, capacity-limited workspace. GWT supplies a concrete vocabulary—competition, broadcast, ignition, bottleneck—for asking whether analogous structure emerges in trained models, and it applies to two architectures that reason in very different ways: recursive latent reasoning models such as HRM (Wang et al., 2025), TRM (Jolicoeur-Martineau, 2025), and GRAM (Baek et al., 2026), which iterate in a fixed-dimensional state space, and auto-regressive LLMs, which externalize intermediate computation as tokens.

Global workspace theory has been taken up in artificial intelligence along two lines. In one, the workspace is installed *by design*: Bengio's *consciousness prior* distills a low-dimensional conscious state from a high-dimensional substrate (Bengio, 2017), and (Goyal et al., 2022) make this architectural, routing specialist modules through a shared, bandwidth-limited channel they must compete

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.

to access. In the other, it appears *unbidden*: (Gurnee et al., 2026b) identify a sparse "J-space" in a language model's activations—carrying the representations it can verbalize and control, feeding many downstream computations at under a tenth of internal activity—that no one built in. This tension between an engineered workspace and an emergent one is the axis of our work: we ask not whether a workspace can be imposed, but under what conditions recursive computation gives rise to one on its own.

As a probe we use *multi-solution constraint-satisfaction problems*. These instances are cheap to generate at controlled difficulty, admit a deterministic verifier, and typically have many valid solutions—so we can ask not only whether a model succeeds, but which solution branch it commits to and at what point in its computation.

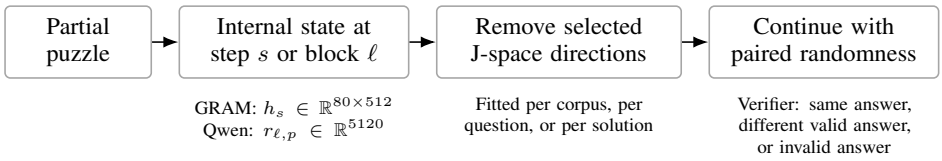


Figure 1: Overall schematic of our J-space analysis. We remove Jacobian-lens directions from an internal state, continue the computation with paired randomness, and verify the final answer, separating a change of solution from a loss of validity.

Our main contributions are as follows: (i) **A verifiable instrument for mechanistic analysis.** We build a testbed of multi-solution constraint-satisfaction problems, n-queens completion and a new cubic-graph edge-coloring collection, whose deterministic verifier classifies any intervention outcome as the same solution, a different valid solution, or an invalid answer. On this testbed we introduce a J-space framework that fits Jacobian lenses at corpus, per-question, and per-solution conditioning levels, so that changes in answer identity can be separated from changes in validity. (ii) **How a recursive reasoner carries a solution.** Applying the framework to GRAM, we show that constraint satisfaction depends on a subspace occupying a fraction of a percent of the latent state, that the recursion repairs single-step damage so only sustained removal is fatal, and that the branch-local subspace maintains the model's commitment to one solution separately from its ability to produce any valid one: removal redirects the answer far more often than it invalidates it. (iii) **Where an autoregressive reasoner enforces constraints.** Applying the same instrument to Qwen3-32B, we localize constraint sensitivity to a narrow band of late decoder layers that neither per-question nor per-solution conditioning moves, and where removal invalidates the answer rather than redirecting it. (iv) **An architectural contrast.** The same workspace role is enforced diffusely over reasoning steps in the recursive model but compressed into a few adjacent layers in the autoregressive one, suggesting a mechanistic account of why recursive refinement can be more robust than autoregressive decoding and a concrete target for reasoning-model design.

2 Related Work

Final-answer accuracy is an incomplete measure of reasoning: transformers degrade as compositional depth grows (Dziri et al., 2023), stronger models still fail systematically on long, constraint-heavy problems (Song et al., 2026), and supervising the search process itself helps more than supervising only the final plan (Lehnert et al., 2024). Verifiable benchmarks in formal proof and combinatorial optimization show the same gap, including fluent solutions that still violate constraints (Liu et al., 2025; Sun et al., 2025; Jiang et al., 2026). These works evaluate the external output; we turn verification inward. On multi-solution instances, a deterministic verifier sorts every intervention outcome into the same solution, a different valid solution, or an invalid answer, revealing which parts of a model's state maintain the constraints and the identity of a solution.

Most reasoning methods for language models work through a textual trace, whether a single chain (Wei et al., 2022), many aggregated chains (Wang et al., 2023), or a searched tree (Yao et al., 2023). A different line reasons by repeatedly refining a latent state. Universal Transformers apply a shared block recurrently over depth (Dehghani et al., 2019), and looped transformers show that iterating a block can emulate programmable computation and improve reasoning with the number of loops (Giannou et al., 2023; Saunshi et al., 2025). Recent recursive reasoners push this further: HRM

couples slow and fast recurrent modules (Wang et al., 2025), TRM shows that one small network applied recursively can outperform it on puzzle benchmarks (Jolicoeur-Martineau, 2025), and GRAM makes the recursion stochastic so that one input yields several distinct solutions (Baek et al., 2026). These models reach strong accuracy on constraint-satisfaction puzzles, but what their latent state carries remains largely unexamined. Because that state is explicit and re-entrant, an edit at one step must be propagated, or repaired, by every later step; we use GRAM's sampled branches to ask whether an intervention changes *which* solution is found or *whether* one is found at all, and contrast it with an autoregressive reasoner whose intermediate computation lives in a token trace.

Global workspace theory holds that specialized processors compete for access to a limited-capacity workspace whose contents are broadcast widely (Baars, 1988; 2005), with a neuronal formulation linking this broadcast to long-range cortical dynamics (Dehaene et al., 1998; Dehaene & Naccache, 2001; Dehaene & Changeux, 2011; Mashour et al., 2020). In AI the theory has mostly served as a blueprint: shanahan2006cognitive combined it with internal simulation, bengio2017consciousness proposed a low-dimensional conscious state as an inductive bias, vanrullen2021deep outlined a deep-learning implementation, and goyal2022coordination coordinated neural modules through a shared, capacity-limited workspace. The complementary direction reads workspace structure off trained models: gurnee2026verbalizable use Jacobian lenses to argue that verbalizable representations in language models form a global workspace. We adopt their lens but change both the measurement axis, from transformer depth to recurrent refinement steps, and the test: rather than asking what is verbalizable, we ask which low-dimensional span must be maintained for a sampled answer to survive.

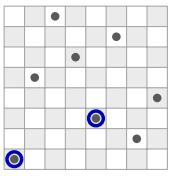
3 METHOD

3.1 TASKS AND VERIFIER-DEFINED ANSWERS

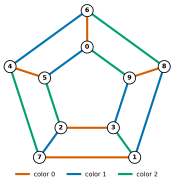
For an input x, let $\mathcal{S}(x)$ be the set of certificates accepted by its verifier. A prediction is valid if its parsed certificate belongs to $\mathcal{S}(x)$. Exact agreement with one reference answer is unnecessary.

N-queens completion. Following Baek et al. (2026), an input is a partial $n \times n$ board. The output must place n queens, preserve every supplied queen, and have exactly one queen in each row and column with no shared diagonals. We use $n = 8$ for all mechanistic experiments and $n \in \{8, 10\}$ for the language-model benchmark. Different valid completions of the same partial board count as different answer identities.

Cubic 3-edge-coloring. We construct a separate benchmark of colorable, connected, simple, bridgeless cubic graphs on 16 vertices. A certificate assigns one of three colors to each edge; the three edges incident to every vertex must have distinct colors. Equivalently, each color class is a perfect matching. We identify equivalent certificates under graph automorphisms and color permutations before measuring solution coverage. Figure 2 illustrates both tasks, and Figure 6 summarizes the graph collection. Construction and verification details appear in Appendix A.



(a) A valid 8-queens completion



(b) A valid edge coloring

Figure 2: Certificates make correctness explicit. (a) Circled queens are supplied clues; the completed board must satisfy all row, column, and diagonal constraints. (b) The pentagonal prism illustrates edge coloring; evaluation uses 16-vertex graphs.

3.2 STOCHASTIC RECURSIVE COMPUTATION

We build on GRAM (Generative Recursive Reasoning Model) (Baek et al., 2026), which reasons by iteratively refining a hierarchical latent state $z = (h, l)$. Unlike the deterministic recursion of HRM,

GRAM makes its high-level updates *stochastic*, inducing a distribution over trajectories that recovers multiple valid solutions. Let $e_x \in \mathbb{R}^{L\times D}$ encode the input, where L is the number of latent positions (board positions plus registers) and D is the hidden width. GRAM maintains high-level and low-level states $h, l \in \mathbb{R}^{L\times D}$. Following the hierarchical GRAM formulation, our implementation uses the transition

$$\begin{aligned} l_{t,0} &= l_{t-1}, & l_{t,k} &= f_L(l_{t,k-1} + h_{t-1} + e_x), \quad k = 1, \dots, K, \\ l_t &= l_{t,K}, & u_t &= f_H(h_{t-1} + l_t), \\ \epsilon_t &= \mu_\theta(u_t) + \sigma_\theta(u_t) \odot \xi_t, & h_t &= u_t + \epsilon_t, \qquad \xi_t \sim \mathcal{N}(0, I). \end{aligned} \tag{1}$$

Here t indexes transitions, K is the number of low-level refinements, f_L, f_H are learned update networks, and $\odot$ denotes elementwise multiplication. Within one transition, the low-level component l is refined K times with h held fixed, after which h undergoes a single stochastic update $h_t = u_t + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(\mu_\theta(u_t), \sigma_\theta^2(u_t)I)$; stochasticity enters only at the high level. A supervision step is a sequence of T transitions whose terminal state initializes the next, and GRAM chains N_{sup} such steps. The shared decoder is applied to h at the end of every supervision step, giving deep supervision along the trajectory; it reads only the content positions of h, and l is carried forward but never decoded. Thus h is the reportable latent state while l supports faster internal refinement.

During training, a target-conditioned posterior supplies ϵ_t and is regularized toward the prior in Eq. 1. Inference uses only the prior. A shared decoder reads the content positions of h_s at each supervision step; it does not directly read l_s or the registers. The final certificate is decoded after a fixed number of steps. GRAM generates no autoregressive reasoning tokens; its computational budget is the number of recurrent updates.

3.3 Jacobians at three conditioning levels

Let $z_x \in \mathbb{R}^R$ contain the final task logits. For an 8-queens board, $R = 64 \times 2 = 128$, with one EMPTY and one QUEEN logit per cell. At step s, define

$$J_{x,s} = \frac{\partial z_x}{\partial \operatorname{vec}(h_s)} \in \mathbb{R}^{R\times LD}, \qquad \bar{J}_s = \frac{1}{M}\sum_{i=1}^{M} J_{x_i,s}, \tag{2}$$

where M is the number of puzzles used to fit the corpus lens and vec flattens the state. The derivative includes all subsequent recurrent updates, including the state dependence of the prior. We do not apply the gradient truncation used for training.

The singular value decomposition $J = U\Sigma V^\top$ yields the input directions with the largest local effects on final logits. We collect its first k right singular vectors as rows of $B \in \mathbb{R}^{k\times LD}$, so $BB^\top = I_k$. The resulting *J-space* is a local sensitivity subspace; a finite intervention is needed to determine its behavioral effect. The chosen width k is an intervention budget, not an inferred intrinsic dimension.

We factorize three different operators. The *corpus* lens factorizes $\bar{J}_s$, giving one basis for all inputs. The *per-question* lens factorizes each $J_{x,s}$. Both use the deterministic reference trajectory $\epsilon_t = \mu_\theta(u_t)$. The *per-solution* lens factorizes $J_{x,b,s}$ along a sampled branch b, with its stored standard-normal draws $\{\xi_t^{(b)}\}$ fixed. Two trajectories that produce the same mathematical solution can still have different Jacobians; "per-solution" therefore does not mean one basis per enumerated solution.

3.4 Removal, replay, and solution transfer

For a flattened state a, basis B, and start-step corpus mean state $\bar{a}$, mean-filled removal replaces

$$a \leftarrow a - B^\top B(a - \bar{a}). \tag{3}$$

A single-step intervention applies this edit once. A persistent intervention starting at s applies the same start-step basis B_s and mean $\bar{a}_s$ after every supervision step $m = s, \dots, 15$, before readout. Thus the start index also determines the number of edits. This experiment measures the effect of sustained removal, not the independent importance of each step.

We pair baseline and intervention runs using the same ξ_t at every transition. After an intervention, μ_θ and σ_θ may change with the state, so the realized ϵ_t need not remain equal. Pairing fixes the underlying

random inputs while allowing the downstream consequences of the edit to unfold. Rank-matched random bases provide GRAM controls where evaluated.

For baseline-valid branches of a multi-solution puzzle, we record three mutually exclusive outcomes: the same solution, a different valid solution, or an invalid output. The first two sum to any-solution validity.

3.5 Layer interventions in Qwen

For Qwen3-32B, ℓ indexes decoder blocks rather than recurrent steps. Following Gurnee et al. (2026a), we fit a task-conditioned transport lens

$$J_\ell^{\mathrm{Q}} = \mathbb{E}_{c,p<q}\left[\frac{\partial r_{63,q}}{\partial r_{\ell,p}}\right] \in \mathbb{R}^{d\times d}, \tag{4}$$

where c is a fitting transcript, p, q are source and future target positions, $r_{\ell,p} \in \mathbb{R}^d$ is the residual state after block ℓ, and block 63 is the target. Rows of $W_U J_\ell^{\mathrm{Q}}$, with unembedding matrix W_U, define token directions in the source state. At each processed position, including prompt prefill, we exclude the clean model's ten highest-probability output tokens, select the ten remaining tokens with highest lens scores, and remove the residual projection onto their lens-vector span. Excluding token identities does not guarantee preservation of their vector coordinates. These two lens constructions share a derivative-based approach but are not quantitatively interchangeable.

For the Qwen conditioning study, the *corpus* lens pools accepted training-side responses and is shared across evaluation inputs. A *per-question* lens pools separate accepted reference responses for one puzzle and is reused across that puzzle's evaluation draws; these references are not the branches being scored. A *per-solution* lens is fitted to the recorded, verifier-valid baseline response for one draw, then evaluated by freely generating the edited continuation with paired sampling randomness. We classify that continuation as the same solution, another valid solution, or invalid. As in GRAM, "per-solution" refers to a sampled path rather than an enumerated mathematical solution. The Qwen operator remains position-dependent even with a shared corpus lens, so these levels describe how the Jacobian is fitted, not a fixed set of directions removed at every token.

4 Experimental Setup

4.1 J-space estimation and interventions

The corpus lens averages all the evaluation-puzzle Jacobians. All ablations draw from the 773-puzzle test split: the corpus sweep removes a rank-64 span on 128 puzzles with four draws each, the per-puzzle sweep removes rank 16 on 64 puzzles with four draws, and the per-solution sweep removes rank 16 on 32 multi-solution puzzles with eight draws. The per-solution sweep retains 30 puzzles and 239 baseline-valid branches after requiring at least two observed valid solutions per puzzle; we average branches within puzzles, then weight puzzles equally. Appendix B.2 gives the full experimental setup.

Before any intervention, we verify that our step-by-step replay of GRAM, which lets us edit the state between supervision steps and reuse the same noise draws, reproduces the unmodified model's output logits exactly. All GRAM removal experiments edit the high-level state h_s at the end of a supervision step, repeat the edit after every later step through step 15, and replace the removed component with its corpus mean (Eq. 3).

4.2 Task-conditioned Qwen experiment

The corpus-level experiment fits one task-conditioned lens on 100 verifier-accepted training-side Qwen3-32B responses and evaluates 100 held-out 8-queens boards once each. It uses 128-token fitting windows, target block $L63$, thinking enabled, a 16,384-token generation cap, temperature 0.6, top-$p = 0.95$, and top-$k = 20$. Its coarse decoder sweep and exploratory late-layer follow-up are historical results, separate from the conditioned study below; Appendix D gives more experimental details.

The conditioned study evaluates per-question lenses on 60 puzzles with four draws each, fitting each lens from four independent, verifier-correct and uncapped reference responses. Its per-solution arm uses the original 32 multi-solution puzzles selected for the GRAM branch study, with eight baseline draws each; only valid, uncapped baseline branches can supply a trajectory lens, and solution-identity summaries additionally require at least two distinct observed valid answers per puzzle. Both arms target $L63$ and test layers $L48, L52, L56, L60, L62$, these late-layer anchors feasible within the available compute budget. We also experimented on corse-grained full-range layers, with results in Appendix D.

5 Results

5.1 Model evaluation on verifiable tasks

We first evaluate how well current models solve these challenging multi-solution mathematical problems whose answers can be verified exactly. Both tasks, n-queens completion and cubic graph edge coloring, require the model to output a complete certificate, which a deterministic verifier checks against every constraint. Because each instance admits multiple valid certificates, we report two metrics: single-draw accuracy, the fraction of draws that return a valid certificate, and coverage, the fraction of distinct valid solutions recovered across repeated draws. We compare the recursive latent reasoner GRAM with the autoregressive Qwen3-32B, evaluated with and without thinking, and with frontier language models (GPT-5.5 and Claude Opus 4.8). The full experimental protocols are in Appendix A. Results are reported in Figure 3.

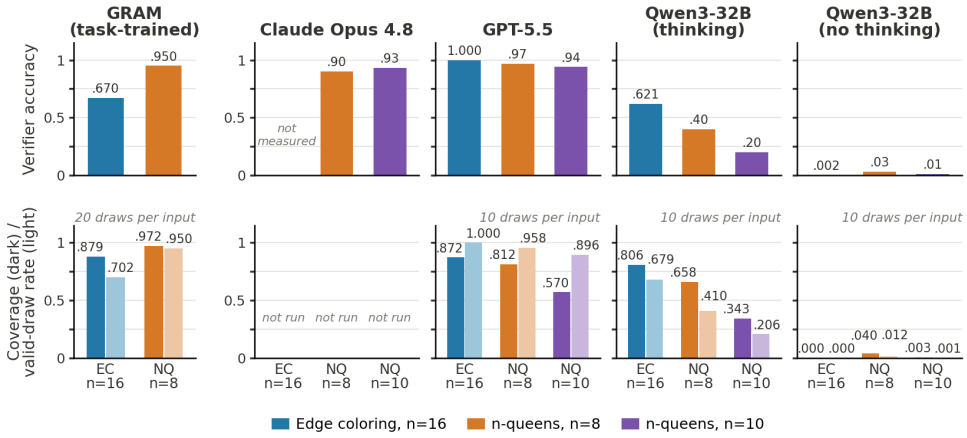


Figure 3: Model evaluation results. Left: task-trained GRAM; coverage samples the entire test set 20 times per input (773 8-queens puzzles or 578 edge-coloring graphs). Right: zero-shot autoregressive language models.

GRAM solves both tasks reliably without any reasoning trace. On 8-queens it returns a valid completion in 95.0% of single draws and finds at least one valid completion for every puzzle within 20 draws, whereas Qwen3-32B reaches only 40% even with thinking. On edge coloring, GRAM also exceeds Qwen in both single-draw accuracy (67.0% vs. 62.1%) and coverage of distinct solutions (87.9% vs. 80.6%). Frontier models such as GPT-5.5 remain strongest, yet even a perfect valid-draw rate does not guarantee recovery of all distinct solutions. Since both GRAM and Qwen return valid certificates on many inputs, and each input admits several valid answers, these tasks let us test whether an intervention preserves a particular answer separately from task success.

5.2 GRAM: validity and answer identity respond differently

This section asks how the recursive model carries a solution through its sixteen refinement steps. Is constraint satisfaction held in a compact subspace of the 40,960-dimensional latent state, is that subspace re-established by the recursion when disturbed, and does it also fix which of several valid

answers the model returns? The three conditioning levels of Section 3.3 probe these questions at increasing specificity, from one basis shared by all puzzles to one basis per sampled trajectory (Figure 4).

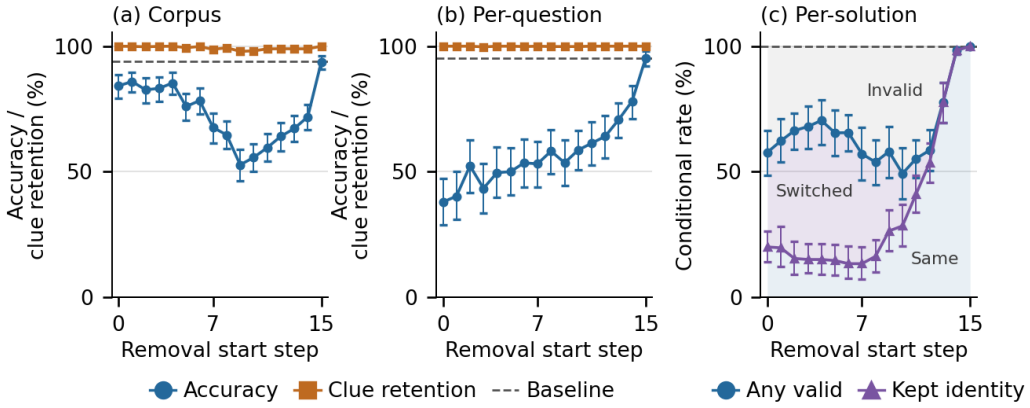


Figure 4: Persistent high-level-state ablation of the GRAM model, from each supervision step through step 15. Left: removal of the corpus J-space. Middle: removal of each puzzle's per-question J-space. Right: removal of each sampled trajectory's own per-solution J-space. Left and middle show accuracy and clue retention over all draws. The right panel shows rates conditional on branches whose baseline answer is valid: bands partition them into same-solution, switched-valid, and invalid outcomes, and the lines give the any-valid and kept-identity rates. Error bars are 95% puzzle-cluster bootstrap intervals; dashed lines are the paired no-intervention baselines.

The recursion repairs local damage, so we remove persistently. A single-step removal barely moves validity: corpus removal at one step leaves it between 92.8% and 94.1% against a 93.9% baseline (Table 4), even though the same edit on sampled trajectories already switches 22.1% of branches to another valid solution. This is the first mechanistic observation: because the encoded input e_x re-enters every transition in Eq. 1, the steps after an edit recompute the removed component, so the recursion behaves as an error-correcting process rather than a feed-forward pipeline. To measure what the state must carry, we therefore reapply the removal from the start step through step 15, accepting that a start step fixes both the onset and the number of edits.

Corpus: a thin shared channel carries the constraint state. Against a paired baseline of 93.9%, persistently removing the corpus rank-64 span lowers validity at most onset steps, bottoming out at 52.7% for onset at step 9, while clue retention stays above 97.9% and the rank-matched random control remains between 93.4% and 94.3% throughout (Figure 4a). Mechanistically, a subspace shared across all puzzles, occupying under 0.2% of the state, carries information the model needs to satisfy the row, column, and diagonal constraints, and this information is separate from the copied clues, which survive the edit. The onset profile shows how the recursion uses its steps: early onsets leave the most steps in which to rebuild the channel and hurt least, middle-to-late onsets remove it when the model is still resolving constraints, and the return to baseline at step 15 follows from editing only the final state. Constraint resolution is thus spread over the recursion and is not complete until late.

Per-question: each puzzle's own channel is far narrower and far more necessary. Because the signed corpus average cancels across puzzles, the shared basis need not contain the directions that matter most for any one puzzle. Removing each puzzle's own rank-16 span reduces validity from 95.3% to 37.9% at step 0, 43.4% at step 3, 53.1% at step 7, and 78.1% at step 14, while clue retention stays at or above 99.7% (Figure 4b). A 16-dimensional span, 0.04% of the state, that the recursion cannot rebuild under sustained removal indicates that the constraint state is puzzle-specific: the model does not solve every board through one fixed set of coordinates, but assembles a low-dimensional working representation for each input and refines it across most of the trajectory. Width and population differ between panels, so the curves do not estimate a minimum rank, and the

sharpness is a property of this run: a checkpoint from a second training recipe falls only from 92.2% to 85.2% at step 3 (Appendix B.2).

Per-solution: the channel maintains a commitment rather than selecting it. Fitting directions on each branch's own sampled trajectory separates two functions that the previous panels conflate: holding onto a particular answer, and being able to produce any valid one. Over 32 multi-solution puzzles with eight matched-noise draws each, outcomes partition into *same solution*, *switched to another valid solution*, and *invalid* (Figure 4c). Early removal destroys identity far more than validity: from step 0, 57.7% of branches remain valid but only 20.0% return the same solution; from step 4, 70.4% and 15.0%. Conditional on validity, 79–81% of branches switch across steps 2–7, whereas the matched random control retains the original solution in at least 99.2% of cases. This reveals a second mechanism inside the first. The branch-local subspace holds the model's current commitment to one solution; remove it and the recursion, still able to satisfy the constraints, re-converges to a different valid board. The two channels also close at different times: switching is zero from step 13 onward while invalidation persists until step 14, so the identity of the answer is fixed a few steps before its validity is finalized. The directions that *select* a solution are different from these: the leading branch-difference component transfers identity between branches (Appendix **??**), and the J-space directions maintain whichever choice has been made.

Taken together, the three levels describe a recursive reasoner that (i) keeps its constraint state in a thin, partly shared, largely puzzle-specific channel of the latent state; (ii) actively repairs that channel across steps, so only sustained disruption is fatal; (iii) commits to one of several valid answers early, holds that commitment in a separate branch-local subspace, and finalizes validity a few steps later; and (iv) can be redirected to a different valid solution without losing the ability to solve the task. This is the functional signature of a global workspace, a limited-capacity state that later computation repeatedly reads and re-writes, realized inside an ordinary latent state with no dedicated module in the architecture.

5.3 Qwen: late-layer sensitivity at three conditioning levels

Applying the same instrument to an autoregressive model asks whether answer validity and answer identity can also be separated when the intervention axis is decoder depth rather than recursion step. The two lenses are not quantitatively interchangeable, so what transfers is the question and the outcome vocabulary, not a shared scale, and each Qwen cohort is read against its own baseline. Results are summarized in Figure 5, with the full-depth sweep in Appendix D.

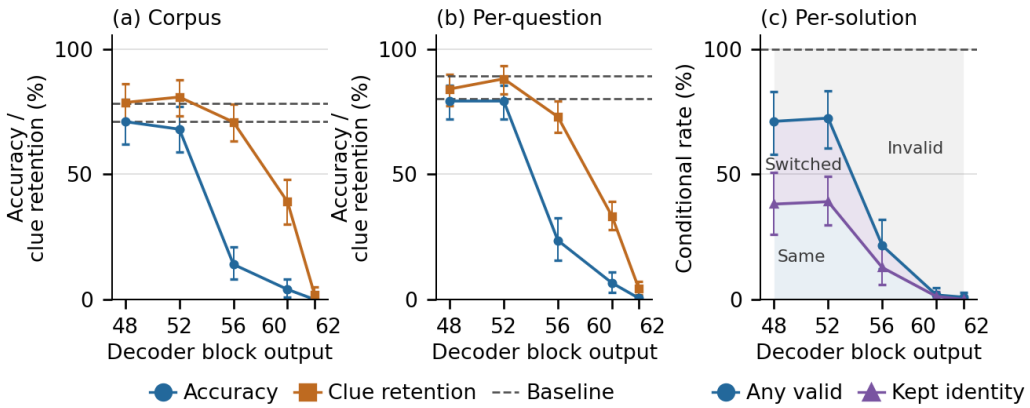


Figure 5: Qwen3-32B removal at three conditioning levels, laid out as Figure 4. Each lens removes $k = 10$ directions at every processed position of one decoder block; the horizontal axis gives that block, and blocks are edited one at a time, not cumulatively. Left: corpus lens. Middle: per-question lens. Right: per-solution lens, with rates conditional on branches whose baseline answer is valid, as in Figure 4. Error bars are 95% puzzle-cluster bootstrap intervals; dashed lines are the no-intervention baselines on the same inputs. No random control is shown.

Corpus lens: enforcement is confined to a narrow late band. Removing the lens-selected directions from a single block leaves validity at baseline through the first four fifths of the decoder, up to and including $L52$, and then collapses it within the next four blocks (Figure 5a). The collapse at $L56$ is selective: clue retention declines only modestly, truncation does not rise, and a norm-matched control at the same layer is unaffected. The model still copies the clues and still produces a board; what it loses is the ability to make that board satisfy the constraints. Edits at the last two tested blocks behave differently, producing truncated or malformed outputs, and so reflect disruption of output production rather than of constraint reasoning. The onset of constraint sensitivity therefore lies between $L52$ and $L56$, and only the $L56$ effect admits the selective interpretation.

Per-question lens: conditioning on the puzzle does not move the boundary. A lens fitted to each puzzle's own reference solutions finds the same transition at the same layers as the shared lens, with validity intact at $L52$, sharply reduced at $L56$, and near zero thereafter, and with the same signature of clue retention falling far less than validity (Figure 5b). Whereas per-puzzle conditioning strengthened the GRAM effect, in Qwen it changes neither the location nor the magnitude of the sensitive band. The band appears to be a property of the model rather than of the particular puzzle.

Per-solution lens: the late band invalidates rather than redirects. Fitting a lens to each valid baseline response and classifying the edited continuation as the same solution, a different valid solution, or invalid gives the sharpest contrast with the recursive model (Figure 5c). At $L52$, outcomes are indistinguishable from what re-sampling alone produces, so in the absence of a random control the switching observed there cannot be attributed to the removed directions. At $L56$, the majority of branches become invalid and only a small minority switch to another valid board. Where GRAM, deprived of its branch-local subspace, re-converges to a different valid solution, Qwen does not recover a valid answer at all: the edit that separates identity from validity in the recursive model damages both together in the autoregressive one.

Both architectures carry constraint enforcement in a compact, causally necessary subspace, but they place it differently. In GRAM the necessary subspace is spread across most of the recursive trajectory, and the recursion repairs local damage, so only sustained removal is fatal and a disrupted branch often lands on a different valid solution. In Qwen3-32B the same role is compressed into a few adjacent decoder blocks, $L52$–$L56$, where a single edit is fatal and the disrupted trajectory rarely recovers a valid answer. *The same workspace role is enforced diffusely over reasoning steps in the recursive model but concentrated in a narrow band of adjacent layers in the autoregressive one.* The boundary marks sensitivity to this edit, not the site of all constraint computation, which may occur earlier; the Qwen control matches perturbation norm but not rank, and the conditioned lenses have no random control, so the contrast is qualitative rather than a matched comparison.

6 Discussion and Conclusions

We asked under what conditions a global workspace arises in a reasoning model that was never given one. In both architectures studied, it arises as a *role* rather than a module: a thin, causally necessary channel inside an ordinary latent state that later computation repeatedly reads and rewrites. Where the two differ is in how that role is realized. The recursive reasoner spreads it across its trajectory, keeps the channel that maintains a commitment to one solution separate from the channel that keeps the solution valid, and actively repairs local damage to either, so that a disrupted trajectory often re-converges to a different valid answer. The autoregressive reasoner compresses the same role into a few adjacent decoder layers, where a single edit destroys validity and identity together and the trajectory rarely recovers. This contrast suggests a mechanistic account of why recursive refinement can reason more effectively than autoregressive decoding, namely that a re-entrant state keeps alternative valid solutions reachable under perturbation, and it points to a concrete target for future reasoning-model design. The account remains a hypothesis: the effects come from one GRAM checkpoint and an autoregressive control that matches norm but not rank, and unequal budgets preclude a performance ranking of the two architectures.

The methodology is not specific to combinatorial problems: any domain with clear verification and explicit constraints can serve as an instrument for mechanistic analysis, making verifiable reasoning tasks a general route to understanding, and ultimately steering, how large models reason.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experimental design, implement methods, clean and reformat datasets, and support qualitative and thematic data analysis. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, or interpret results; the formulation of mathematical claims, the provision of critical ingredients for proofs, the writing of proofs, and translation are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures and to edit the paper for readability. We have reviewed all AI-assisted work: AI-generated code was verified and tested for correctness by the authors, and AI-assisted figures and text were checked against the underlying experimental records. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, Cambridge, 1988.

Bernard J. Baars. Global workspace theory of consciousness: Toward a cognitive neuroscience of human experience. *Progress in Brain Research*, 150:45–53, 2005.

Junyeob Baek, Mingyu Jo, Minsu Kim, Mengye Ren, Yoshua Bengio, and Sungjin Ahn. Generative recursive reasoning, 2026. URL https://arxiv.org/abs/2605.19376.

Yoshua Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

Stanislas Dehaene and Jean-Pierre Changeux. Experimental and theoretical approaches to conscious processing. *Neuron*, 70(2):200–227, 2011.

Stanislas Dehaene and Lionel Naccache. Towards a cognitive neuroscience of consciousness: Basic evidence and a workspace framework. *Cognition*, 79(1–2):1–37, 2001.

Stanislas Dehaene, Michel Kerszberg, and Jean-Pierre Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, 95(24): 14529–14534, 1998.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*, 2023.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning (ICML)*, 2023.

Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. Coordination among neural modules through a shared global workspace. In *International Conference on Learning Representations (ICLR)*, 2022.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026a.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models, 2026b. URL https://arxiv.org/abs/2607.15495.

Xia Jiang, Jing Chen, Cong Zhang, Jie Gao, Chengpeng Hu, Chenhao Zhang, Yaoxin Wu, and Yingqian Zhang. Reasoning in a combinatorial and constrained world: Benchmarking LLMs on natural-language combinatorial optimization. *arXiv preprint arXiv:2602.02188*, 2026.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*, 2025.

Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul McVay, Michael Rabbat, and Yuandong Tian. Beyond A*: Better planning with transformers via search dynamics bootstrapping. *arXiv preprint arXiv:2402.14083*, 2024.

Junqi Liu, Xiaohan Lin, Jonas Bayer, Yael Dillies, Weijie Jiang, Xiaodan Liang, Roman Soletskyi, Haiming Wang, Yunzhou Xie, Beibei Xiong, Zhengfeng Yang, Jujian Zhang, Lihong Zhi, Jia Li, and Zhengying Liu. CombiBench: Benchmarking LLM capability for combinatorial mathematics. *arXiv preprint arXiv:2505.03171*, 2025.

George A. Mashour, Pieter Roelfsema, Jean-Pierre Changeux, and Stanislas Dehaene. Conscious processing and the global neuronal workspace hypothesis. *Neuron*, 105(5):776–798, 2020.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations (ICLR)*, 2025.

Peiyang Song, Pengrui Han, and Noah Goodman. Large language model reasoning failures. *arXiv preprint arXiv:2602.06176*, 2026.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36*, 2023.

A Experimental Details and Evaluation Protocols

A.1 GRAM system and data

Primary GRAM interventions use the selected raw-weight checkpoint of one seed-42 run of our community implementation. It follows GRAM's hierarchical recurrence and n-queens data recipe (Baek et al., 2026), but is not an exact reproduction. The model has 16 supervision steps and a 40,960-dimensional recurrent state; Appendix A.4 gives the architecture and checkpoint identity.

Training uses AdamW and a balanced KL objective. Appendix A.4 gives the optimizer, KL, EMA, and checkpoint settings, including those of the separately trained edge-coloring model and of a 10-queens model that appears only in the behavioral comparison.

Removing five, six, or seven queens from complete 8-queens boards produces 5,148 unique partial inputs. The 85:15 input-level split contains 4,375 training and 773 evaluation puzzles, including 259 evaluation puzzles with multiple completions. This evaluation split participated in historical checkpoint selection and lens estimation; it is not an untouched final test set. Reported effects and intervals are conditional on the selected checkpoint.

A.2 Dataset construction and verification

N-queens. All 92 complete 8-queens placements are enumerated. Removing five, six, or seven queens produces 5,148 unique partial boards; an input-level split gives 4,375 training and 773 evaluation inputs, which have 1,203 completions, at most 18 per input. Multiple target completions may correspond to one input. GRAM uses a fixed board representation and predicts all board positions. Language models return coordinate pairs or the prescribed row-to-column representation, which is parsed before verification. Full validity requires in-bounds coordinates, one queen per row and column, no diagonal conflict, and preservation of the input queens. *Clue retention* is the fraction of supplied queens retained, averaged over puzzles and draws; it is not the fraction of puzzles on which every clue survives. Unparseable language-model outputs receive zero clue retention.

Cubic edge coloring. Enumeration yields 3,874 non-isomorphic connected bridgeless cubic graphs on 16 vertices. We remove the 26 without a proper 3-edge-coloring, leaving 3,848 colorable graphs. The graph-level 85:15 split contains 3,270 training and 578 evaluation graphs. Each input presents an edge list; `edge_colors` must contain one of $\{0, 1, 2\}$ for every listed edge. The verifier checks that all three colors occur at each vertex. Vertex labels and edge order are randomized before presentation. Canonicalization under automorphisms and color permutations is used for identity and coverage, rather than for deciding whether a coloring is valid. The evaluation split contains 2,101 essential colorings. This task differs from GRAM's published random-graph vertex-coloring experiments (Baek et al., 2026).

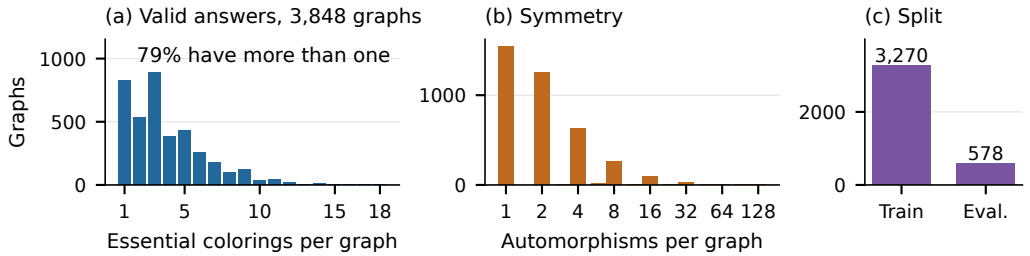


Figure 6: The 16-vertex edge-coloring collection, counted over unique colorable graphs. (a) Most graphs admit several essential colorings, which motivates measuring coverage under repeated sampling. (b) Order of the automorphism group; 1 means no symmetry. (c) The 85:15 split assigns 3,270 graphs to training and 578 to evaluation, with no isomorphism class shared across them.

Input overlap in the language-model benchmark. The family-specific n-queens item sets are not paired. The Qwen and GPT-5.5 sets overlap in 16 of 100 inputs at $n = 8$ and 8 of 100 at

$n = 10$. Their aggregate accuracies must not be interpreted as within-puzzle treatment contrasts. The 76-instance Claude $n = 10$ result was reconstructed from saved logs; its original responses, input identifiers, completion-token counts, and latency are unavailable. We retain the aggregate point with this limitation rather than filling those fields by inference.

A.3 Accuracy, repeated sampling, and computational cost

Validity and pass rates. Single-draw validity is the fraction of decoded certificates accepted by the verifier. For GRAM's 8-queens trajectory evaluation, the reported pass@1 is estimated from 20 independent model draws per puzzle. The archived GRAM edge-coloring single-draw score uses one designated draw. Pass@20 asks whether at least one of 20 draws is valid. It is an oracle availability metric, not the accuracy of a model-selected answer. The 8-queens epoch-300 pass@1 and coverage are 0.94987 and 0.97174 under the retained 20-draw protocol; older one-draw summaries use a different evaluation protocol and are not mixed into Figure 3.

Essential-solution coverage. For puzzle or graph j, let $\mathcal{S}_j$ be its valid solution identities and $\mathcal{D}_j^{(B)} \subseteq \mathcal{S}_j$ the identities discovered in B draws. We report

$$\text{coverage}_B = \frac{\sum_j |\mathcal{D}_j^{(B)}|}{\sum_j |\mathcal{S}_j|}. \tag{5}$$

For language-model edge coloring, $B = 10$ on the first 100 evaluation graphs in dataset order. That subset has 227 essential colorings, at most eight per graph, and 42 single-solution graphs. GRAM's $B = 20$ coverage uses each task's full evaluation split. These are micro-averages over solutions; they differ from pass@B, which averages whether each input has any success. Valid-draw rate averages the correctness of individual generations.

Generation budgets. For edge coloring, Qwen uses a 32,768-token cap, temperature 0.6, top-$p = 0.95$, top-$k = 20$, and thinking enabled; GPT-5.5 uses an 8,192-token cap. Table 1 gives n-queens budgets and costs. The saved Qwen no-thinking $n = 8$ run shows a 4,096-token cap despite a declared 8,192 cap in the summary configuration; the table uses the observed cap. Not all sampling arguments were preserved for the archived n-queens benchmark. These runs are separate from the 16,384-token Qwen mechanistic experiment.

Board	Model / mode	Items	Cap	Mean total tokens	Mean seconds
8×8	Claude Opus 4.8	100	4,096	2,505	27.2
	GPT-5.5	100	4,096	1,532	23.9
	Qwen3-32B / thinking	100	8,192	7,764	368.4
	Qwen3-32B / no thinking	100	4,096	756	25.3
10×10	Claude Opus 4.8	76	16,384	6,390	—
	GPT-5.5	100	16,384	3,605	50.1
	Qwen3-32B / thinking	100	32,768	22,068	2,024.9
	Qwen3-32B / no thinking	100	32,768	645	18.0

Table 1: Archived language-model benchmark budgets and efficiency. Total tokens include prompt and completion; the cap applies to completion only. Provider and hardware differences preclude a controlled latency comparison. Dashes denote unavailable measurements. Model names follow the archived endpoint labels. GRAM has no corresponding autoregressive token count.

A.4 GRAM training and implementation

The architecture uses $N_{\text{sup}} = 16$, $T = 3$, $K = 4$, $D = 512$, eight attention heads, and two attention–SwiGLU blocks in each update network. Its $L = 80$ positions include 16 learned registers and give 40,960 state coordinates. The recurrence uses additive input injection, rotary position encodings, and fixed Gaussian initial recurrent states. The posterior is conditioned on target information at each latent position. The raw weights are used for both the main interventions and the retained training-trajectory evaluation.

Optimization uses AdamW with initial learning rate 10^{-4}, weight decay 1, gradient clipping at norm 1, and batch size 768. The learning rate follows cosine decay to zero over 3,000 epochs. EMA is tracked at 0.9999 but is not used for the analyzed checkpoint.

Training uses a supervised output loss, balanced posterior-to-prior KL, a halt-head binary cross-entropy loss, and an LPRM-head mean squared error loss. The latter two have unit weights in the retained run. For posterior q_ϕ, prior p_θ, and stop-gradient operator sg, the balanced KL term is

$$\mathcal{L}_{\mathrm{KL}} = \alpha\, D_{\mathrm{KL}}(\mathrm{sg}(q_\phi)\|p_\theta) + (1-\alpha) D_{\mathrm{KL}}(q_\phi\|\,\mathrm{sg}(p_\theta)), \qquad \alpha = 0.8. \tag{6}$$

It is summed over hidden dimensions and all 80 latent positions, then weighted by $\beta = 0.07$, equivalent to 5.6 if positions are averaged; free bits $\lambda = 0$. The main analysis runs all 16 supervision steps and does not use the learned halt head to stop early or the LPRM head to select a prediction. Training truncates gradients to the final transition of each supervision step; the analysis reconstructs the full differentiable continuation instead.

The edge-coloring evaluator uses a separately trained seed-42 model on the graph training split, with graph relabeling augmentation and AdamW (initial learning rate 10^{-4}, weight decay 1, gradient clipping at norm 1, global batch size 768). Its cosine schedule spans 1,500 epochs; the raw-weight checkpoint selected on validation is from epoch 650. Its effective KL coefficient is 33 under the convention that averages over latent positions. The retained evaluation uses 20 draws on 578 graphs. Its 67.0% single-draw validity and 87.9% coverage must not be conflated with the 95.0%/97.2% values of the n-queens model. No across-training-seed error bars are claimed for either model.

B Supplementary GRAM Analyses

B.1 Signed consistency and unsigned overlap

For output row r, let $J_{x_i,s,r} \in \mathbb{R}^{LD}$ denote the gradient of that logit with respect to the step-s state. The row-balanced directional-consistency statistic is

$$\rho_s = \frac{1}{R}\sum_{r=1}^{R} \frac{\left\| M^{-1}\sum_{i=1}^{M} J_{x_i,s,r} \right\|_2^2}{M^{-1}\sum_{i=1}^{M} \|J_{x_i,s,r}\|_2^2}. \tag{7}$$

Here M is the number of fitted puzzles and $R = 128$ is the number of semantic output rows. The $1/M$ comparison assumes uncorrelated signed gradients of comparable scale; it is a reference scaling, not a universal null for every shared representation. Table 2 reports the archived measurements. They lie close to $1/M$, supporting cancellation of signed effects in the corpus average. This statistic cannot establish pairwise orthogonality: gradients can cancel even when they occupy the same subspace.

Fitting puzzles M	$1/M$	ρ_3	ρ_7
64	0.015625	0.015625	0.015633
128	0.0078125	0.007815	0.007782
256	0.00390625	0.003908	0.003894

Table 2: Signed consistency for the epoch-300 high-level state. Values are rounded as retained in the diagnostic records; no sampling interval is available for this table.

For orthonormal row bases $B_1, B_2 \in \mathbb{R}^{k\times LD}$, unsigned overlap is $\|B_1 B_2^\top\|_F^2/k$. The reported cross-puzzle top-16 overlap is approximately 0.02–0.06, compared with the isotropic reference $16/40{,}960 \simeq 0.00039$. Thus cancellation of the mean coexists with nonzero unsigned overlap. The causal top-16 span captures about 1.5–1.9% of activation variance at steps 3 and 7, versus roughly 70% for rank-16 PCA. Sensitivity-selected directions therefore need not coincide with directions of largest activation variance. Neither comparison by itself tests broadcast or uniquely identifies a shared workspace.

B.2 Comparison of ablation designs

Table 3 records the different estimands. All all-step sweeps edit the terminal high-level state before decoding, hold the selected start-step basis and reference mean fixed, and repeat the edit through step

15. There are 16 applications at start step 0 and one at start step 15. Directions are not refitted to the intervened trajectory after each edit.

	Corpus	Per-puzzle	Per-solution
Operator	Mean of puzzle Jacobians	One Jacobian per puzzle	One Jacobian per sampled branch
Reference	Deterministic mean path	Deterministic mean path	Stored sampled path
Width k	64	16	16
Puzzles $\times$ draws	128×4	64×4	32×8 before filtering
Baseline validity	0.9395	0.9531	0.9961 before filtering
Primary outcome	Any valid solution	Any valid solution	Same / switched / invalid
Random control	Rank 64, all steps	Rank 16, selected steps in separate runs	Rank 16, all steps

Table 3: All-step GRAM experiments. Width, population, reference path, and endpoint differ, so between-panel comparisons do not isolate any one design choice.

Trajectory eligibility and weighting. Let $\mathcal{B}_p$ be the eight sampled branches for puzzle p, and y_p^b its baseline solution identity or $\perp$ for invalidity. A branch is eligible when $y_p^b \neq \perp$ and the puzzle's baseline draws include at least two distinct valid identities. Write $\mathcal{E}_p$ for the eligible branches and $\mathcal{P}^*$ for puzzles with a nonempty eligible set. For an outcome O, the reported rate is

$$\widehat{r}(O) = \frac{1}{|\mathcal{P}^*|} \sum_{p \in \mathcal{P}^*} \frac{1}{|\mathcal{E}_p|} \sum_{b \in \mathcal{E}_p} \mathbf{1}\{O(p, b)\}. \tag{8}$$

There are 30 eligible puzzles and 239 branches. The baseline same-solution rate is one by construction on this conditional set. Puzzles receive equal weight; their branches reflect the model's own sampling frequencies rather than uniform enumeration of mathematical solutions. Puzzle-cluster bootstrap resampling preserves this hierarchy.

Control summaries. The corpus rank-64 random control ranges from 0.9336 to 0.9434 validity across all start steps. The trajectory rank-16 random control's same-solution retention never falls below 0.9917. These quantities are retained even though the standalone random-control plots have been removed. The per-puzzle all-step artifact does not contain a full all-step random arm; its selected-step controls must not be presented as a complete matched curve.

Single-step and persistent removal. Table 4 uses a preliminary lens artifact distinct from the all-step artifact in Figure 4. Within this table, checkpoint, lens, puzzle population, draws, fill rule, and noise are matched across scopes. Thus it isolates the effect of repeating an edit, but its points should not be combined with the later all-step sweep.

	Corpus $k = 64$		Corpus $k = 16$	
Start step	Single-step	Persistent	Single-step	Persistent
0	0.9395	0.8027	0.9277	0.8672
3	0.9336	0.8203	0.9375	0.8691
7	0.9316	0.6602	0.9395	0.8770
11	0.9375	0.5859	0.9414	0.9063
15	0.9375	—	0.9395	—

Table 4: Validity in the preliminary scope comparison; baseline 0.9395. Dashes denote unreported persistent cells. A transient null does not imply that repeated removal is harmless.

The encoded input is reintroduced at each transition, making reconstruction after perturbation possible. Cross-puzzle whole-state replacement provides supporting behavioral evidence: valid completions are much more common after an early replacement than after a final-step replacement. Neither experiment directly observes the reconstruction of a specific removed feature. Accordingly, we describe recovery as an interpretation of the behavior, not as an independently identified algorithm.

Single-step per-solution removal. A single-step sweep on the same 30 eligible puzzles and 239 branches as Figure 4c gives, at start step 0, same-solution 77.4%, switched-valid 22.1%, and invalid 0.4%; the matched random control gives same-solution 98.8%. Any-solution validity never falls below 99.1% at any start step. The switched-valid rate declines to 6.3% at step 3 and is at most 1.7% from step 6 onward. This sweep uses the same eligibility rule, weighting, and stored noise as the persistent sweep.

A second training run. The preliminary per-puzzle artifact also evaluated a checkpoint from a separately trained register-16 run with a different recipe (`v12r`, epoch 250) under the same persistent mean-filled removal, with four draws per puzzle and 256 paired draws in total. Its validity falls from 92.2% to 85.2% from step 3 and to 86.7% from step 7, with clue retention 100%. The sharp effect of the epoch-300 `v13` checkpoint on the same protocol (95.3% to 43.4% from step 3) is therefore a property of the analyzed run, not a general property of the architecture.

C Exploratory Measurements Across Training

This appendix retains the training analysis separately from the main intervention claims. A fixed panel is evaluated at 60 checkpoints, every 50 epochs from epoch 50 to epoch 3,000, across all 16 supervision steps. Geometry uses 16 mean-path puzzles and eight puzzles with two stochastic draws each. Behavioral measurements use the complete 773-puzzle evaluation split with 20 draws per checkpoint. All checkpoints belong to the same training run.

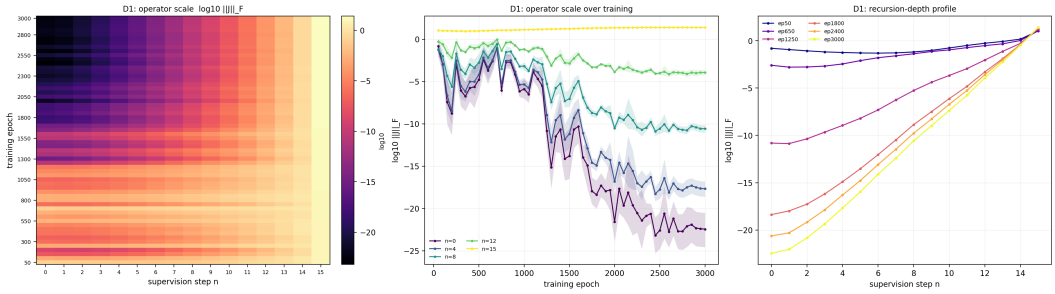


Figure 7: Exploratory operator scale across one GRAM training run. Left: $\log_{10} \|J_{x,b,s}\|_F$ over epoch and supervision step; right: selected steps with puzzle-cluster uncertainty intervals. The vertical marker identifies the epoch-300 checkpoint used in the main interventions. This figure measures derivatives, not the effect of removing early computation.

The scale of the early-step state-to-output Jacobian decreases markedly late in training. The archived step-0 log-norm is approximately -6.1 at epoch 300 and -22.5 near epoch 3,000, while the readout-boundary operator remains much larger. Very small local derivatives can arise in saturated or contractive regions and do not establish that an early state is dispensable under finite intervention. Near numerical limits, normalized spectral summaries also require care; we do not infer a stable intrinsic rank from these measurements.

Under the retained evaluation protocol, mean single-draw validity falls from 0.94987 at epoch 300 to 0.69715 at epoch 3,000. Pass@20 is 1.000 at both endpoints, although it dips slightly below one at some intermediate checkpoints. Repeated sampling thus retains high access to valid completions despite a lower valid-draw rate. This does not establish miscalibration, an argmax-selection failure, or a specific change in the prior's modes, because those hypotheses require additional measurements.

The panel contains no interventions at later checkpoints, no independent training-seed replication, and a small geometry population. Checkpoints are strongly dependent observations. Accordingly, this appendix is a descriptive record of one training trajectory and does not extend the epoch-300 causal effects to all training stages.

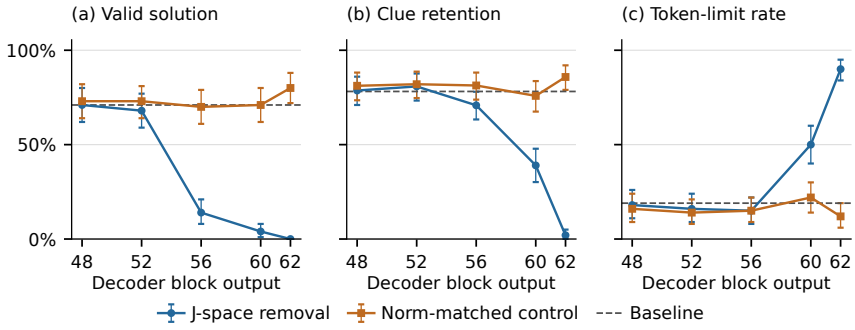


Figure 8: Archived corpus-lens Qwen pilot: one draw on each of 100 puzzles. Validity, mean clue retention, and token-limit rate distinguish constraint failure from output disruption. At $L56$, validity falls much more than clue retention, without increased truncation. At $L60$ and $L62$, multiple outcomes deteriorate. Error bars show 95% paired puzzle-bootstrap intervals. Controls match perturbation norm, not subspace rank. This figure contains no per-question or per-solution results.

D Qwen Protocol and Additional Layer Results

The archived corpus-level task-conditioned lens follows the transport construction of Gurnee et al. (2026a). Block labels denote residual states after zero-based decoder blocks, so $L63$ is the final target boundary and $L62$ is the last admissible source. A sparse list of fitted source layers is not a continuous workspace band. Fits use verifier-accepted 8-queens reasoning transcripts from training-split boards; no generic-corpus result is substituted for this task-conditioned experiment. Its older broad sweep is retained here as historical context; the new conditioning study is restricted to $L52, L56, L60, L62$.

At each processed position, including prompt prefill and subsequent decoding, the lens transports the residual state into the final-layer basis. The clean model's ten highest-probability output token identities are excluded from selection, and ten high-scoring remaining rows of $W_U J_\ell^{\mathrm{Q}}$ define the removed span. This exclusion does not orthogonalize the edit against those tokens' directions. The edit subtracts the projection without GRAM's mean-fill offset. The comparison arm uses a single isotropic direction scaled to match the removed norm at that site. This control addresses perturbation magnitude, but does not control for rank or every property of the selected span.

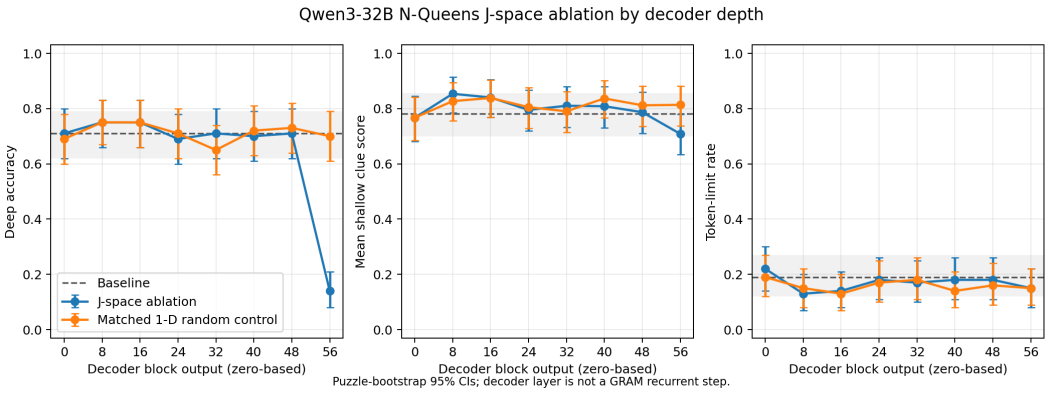


Figure 9: Archived coarse corpus-lens Qwen sweep. Each condition edits one of $L0, L8, \ldots, L56$ during prompt prefill and subsequent decoding; evaluation uses 100 puzzles and the same one-draw protocol as Figure 8. The historical follow-up added $L52, L60, L62$ after observing the effect at $L56$. These older source layers are not the planned four-layer conditioning sweep.

The paired validity difference between J-space and the norm-matched control is -5 percentage points at $L52$, with 95% interval $[-15, +5]$; the corresponding differences are -67 at $L60$, interval

Condition	Validity	Mean clue retention	Token-limit rate
Baseline	0.71	0.7817	0.19
J-space $L52$	0.68	0.8083	0.16
J-space $L56$	0.14	0.7083	0.15
J-space $L60$	0.04	0.3900	0.50
J-space $L62$	0.00	0.0200	0.90

Table 5: Selected Qwen outcomes. Clue retention includes zeros for unparseable outputs, so its late-layer decline combines missing certificates with failure to preserve clues.

$[-76, -58]$, and -80 at $L62$, interval $[-88, -72]$. The archived follow-up reports Holm-adjusted bootstrap p-values 0.374, 0.0003, and 0.0003 for these three new layers. Those corrections concern that follow-up family, not every comparison across the entire research program.

We avoid inferring intact fluency or intact search from verifier statistics alone. At $L56$, the validity loss is much greater than the clue-retention loss, and truncation is not more frequent. At $L60$ and $L62$, missing and corrupted outputs become substantial, so low validity no longer supports a narrowly selective interpretation. The near-output lens also changes with the source–target separation; no identity-map limit is assumed for the cross-position Jacobian in Eq. 4.

In the archived pilot, each puzzle contributes one generated response per condition. Shared resampling indices preserve pairing in the bootstrap, but do not create independent estimates of decoding noise. An independent preregistered layer sweep, rank-matched controls, and repeated generations would strengthen localization and selectivity claims.

A second task: fitting and testing on edge coloring. The experiments above fit and test on n-queens. A further run repeats the whole procedure on edge coloring: the lens is fitted on verifier-accepted edge-coloring transcripts, and the same rank-10 output-protected removal is applied at one decoder block on 100 held-out graphs, under the same 16,384-token cap. Its no-intervention baseline is 0.54 validity with a 0.32 token-limit rate. Both differ from the n-queens baseline, so each task is read against its own baseline and the two are never differenced.

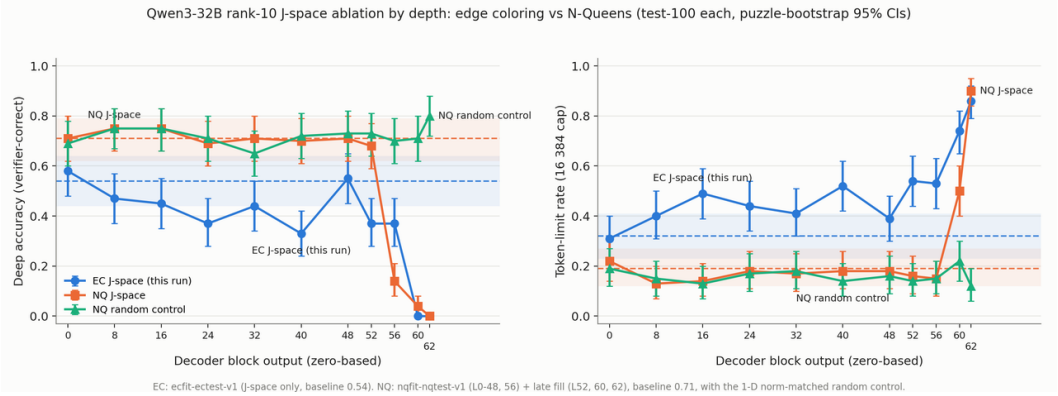


Figure 10: Rank-10 removal on 100 held-out items per task, with 95% puzzle-bootstrap intervals. Dashed baselines mark 0.54 edge-coloring and 0.71 n-queens validity. Each task has its own lens and item set. Only the n-queens arm has a one-dimensional norm-matched control; read each arm against its own baseline.

On n-queens, validity stays near baseline through $L52$ and then collapses. Edge-coloring validity declines at most tested blocks (0.47 at $L8$, 0.33 at $L40$, and 0.37 at $L56$, from a 0.54 baseline), while its token-limit rate rises from 0.32 to 0.39–0.54. Both tasks collapse at $L60$–$L62$. Because the edge-coloring arm lacks a control and truncation rises, this run does not identify a selective, distributed constraint channel for that task.

The n-queens lens does not transfer to edge coloring. An off-diagonal pilot applies the n-queens lens to edge coloring on 16 graphs, beside same-task lenses and their own baselines (Figure 11).

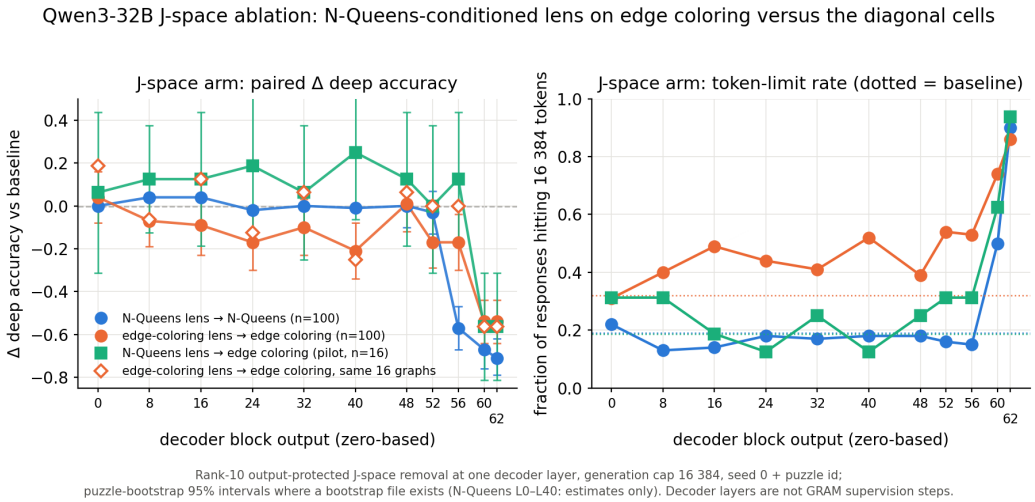


Figure 11: Cross-task pilot on 16 edge-coloring graphs, alongside two same-task arms. Left: paired validity change from each arm's baseline; right: token-limit rate. Open markers show the edge-coloring lens on the same 16 graphs, exposing population variability.

Up to $L56$, transfer shows no validity loss; its paired difference is $+0.13$ at $L56$, versus -0.57 for the same lens on n-queens. This 16-graph pilot cannot rule out a moderate effect (95% interval $[-0.19, +0.44]$). At $L60$–$L62$, every arm collapses. The results are consistent with task-conditioned directions, but do not prove that the task subspaces are disjoint.