WHEN CONTEXTUAL SKILLS CAN BE WEIGHTED: THE COMPILABILITY AND REACHABILITY OF IN-CONTEXT TO IN-WEIGHT LEARNING FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models can rapidly acquire new task capabilities from instructions, rules, and examples provided in context, but repeatedly invoked skills incur persistent computation and context overhead. We study how such context-induced capabilities can be compiled into reusable parameter updates so that the model preserves the desired behavior after the skill is removed. We show that successful compilation depends not only on optimization, but also on whether finite calibration data identify the target behavior, whether the target behavior is realizable within the model's parameterization, and how well limited update capacity can approximate it. We derive a characterization for each of these conditions: identifiability, structural realizability, and low-rank approximation. Based on these results, we develop a screening procedure that evaluates candidate calibration sets, update layers, and ranks before expensive behavioral optimization. In a fully enumerated, theoretically supported configuration space, the procedure reaches 95% of the best configuration in that space while using $4.76\times$ less candidate-training GPU time than random search. These results show that our theory can both explain the structural limits of skill-to-weight compilation and directly reduce its practical search cost.

1 INTRODUCTION

As large language models take on long-horizon, repeated agentic tasks, textual skills have become a common way to inject capabilities: task rules, tool instructions, or workflows placed directly in context give a model specific behavior without training (Anthropic, 2025; Agent Skills, 2025). The main benefit is immediacy, since a skill can be added, replaced, or removed at any time. Its cost, however, recurs on every call: the same tokens re-enter the model with each query, consuming context budget and compute. A parameter update makes the opposite trade, a one-time adaptation cost followed by reuse (Ouyang et al., 2022); the two modes are termed in-context and in-weight learning (Chan et al., 2022). For a frequently invoked skill, the natural question is: can we process the skill once, remove it from the runtime context, and consolidate its capability into weights?

Recent work makes this possibility increasingly concrete. Transformers can implement gradient-descent-like learning in the forward pass (Akyürek et al., 2023; von Oswald et al., 2023); the effect of context on attention, or on a full Transformer block, can be expressed as an implicit or explicit parameter update (Chen et al., 2024; Dherin et al., 2025; Goldwaser et al., 2025); and recent methods fit reusable static patches from calibration examples, turning a transient prompt into a persistent parameter object (Dherin et al., 2025; Mazzawi et al., 2025). Together, these results show that context-induced computation can be transferred into model parameters. Related work reduces context or encodes behavior in parameters in other ways, through prompt compression (Lester et al., 2021; Mu et al., 2023), context distillation (Snell et al., 2022), task and function vectors (Ilharco et al., 2023; Hendel et al., 2023; Todd et al., 2024), and model editing (Meng et al., 2022; Mitchell et al., 2022). However, a practical deployment must still decide which of the many possible compilations of a given skill should be constructed, each specified by a calibration set, an update layer, and a rank.

The choice of compilations comes at a cost. Under a fixed training recipe, candidates can currently be distinguished only by training and evaluating each one. Search rather than training therefore

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

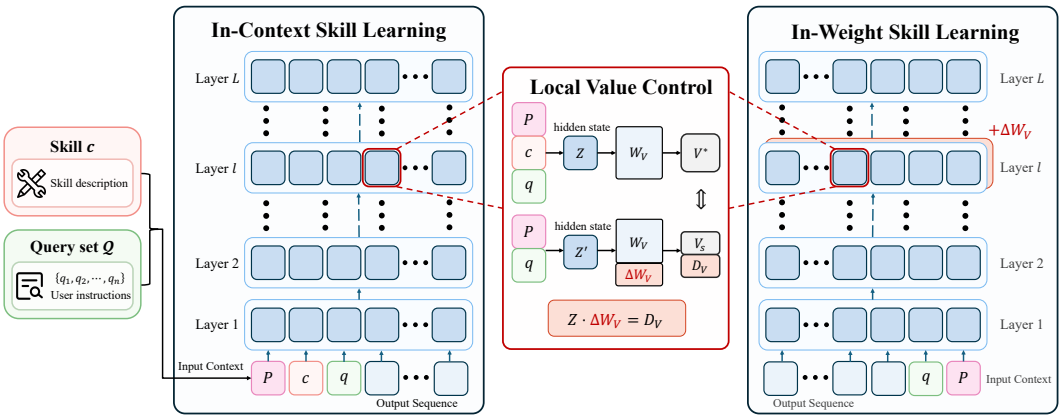


Figure 1: **Compiling a skill into weights through local value control.** *Left:* In in-context skill learning, the skill c is placed in the input context together with the persistent context P and a query q from the query set $\mathcal{Q}$. *Right:* In in-weight skill learning, ΔW_V is merged into layer ℓ, and the model reproduces the skill's effect without c in its context. *Centre:* At layer ℓ, a non-skill position produces the value V^* when the skill is present and V_s when it is absent; the gap between them is the required value displacement D_V. A weight update ΔW_V is learned so that $Z\,\Delta W_V = D_V$.

dominates the compilation cost: enumerating the candidates costs many times more than training the single configuration that is ultimately deployed. Whether a candidate is viable depends on three distinct factors: whether finite calibration data identify the target behavior, whether the parameters shared across attention heads can represent the behaviour, and how accurately a given rank budget can approximate the behaviour.

We begin by characterizing these factors through the frozen model's structure and responses to skill tokens, without relying on task-loss optimization. We define the compilation target as the attention-value control induced by the skill tokens and computed directly from each head's attention. Relative to this target, each factor becomes a property of the frozen model, the skill, and the calibration set, with two quantities requiring little additional computation: one is obtained from the tokenizer and the other from a single activation capture. This yields a ranking over candidate compilations $m = (\mathcal{Q}, \ell, r)$ before any adapter is trained. Behavioral optimization can then go first to the candidates favored by model structure, rather than allocating rank or placement during training (Hu et al., 2022; Zhang et al., 2023). In a fully enumerated space of 90 configurations, repeated over eighteen skill seeds, this ranking identifies configurations whose task score reaches 95% of the best task score in the space, while reducing the candidate-training GPU time by $4.76\times$ relative to random search, or $1.86\times$ including the cost of screening. We further find that the same structural signals remain predictive across model scales, skill families, and architectures.

Overall, our contributions are summarized as follows:

- **A structural account of skill compilation.** We cast skill compilation as a shared-control problem over attention values and derive three factors, each computable before training, that govern whether a candidate compilation can recover the target behavior.
- **A training-free ranking of candidate compilations.** We turn these factors into a screening criterion that orders candidate calibration sets, update layers, and ranks before training.
- **Evidence on real activations and a search cost reduction.** We show that the structural quantities predict compilation quality on measured activations and that the ranking reduces search cost and remains useful across model scales, skill families, and architectures.

2 Compiling a Skill into Weights

A fixed Skill compilation should answer three questions: whether finite calibration data identify the required behavior, whether the architecture admits the corresponding shared control, and how accurately a finite-rank update can represent it. We answer these questions relative to a common control target. Sections 2.1 and 2.2 derive the control that disappears with the Skill and express its

restoration as a shared parameter-update problem. Sections 2.3 to 2.5 characterize the three limits. Each is fixed before any adapter is trained, which makes a preflight possible.

2.1 The Skill-Induced Control

We define task-level Skill compilation as a map $\Delta_c = G(\theta, c, \mathcal{Q}_{\text{cal}})$. Given a frozen model θ, a Skill c, and a calibration pool $\mathcal{Q}_{\text{cal}}$, the map produces a fixed parameter update before deployment. The Skill specifies a deterministic interpreter I_c. The goal is for $f_{\theta+\Delta_c}$ to reproduce on unseen queries the capability induced by c, measured against I_c. At deployment, the input contains neither the Skill nor the calibration examples, and the same Δ_c is shared across all queries.

To characterize the behavior that the compiled update must restore, we analyze a single attention head during Skill-conditioned execution. We represent the input as $P \oplus C \oplus q$, where P denotes the persistent context, C denotes the removable block of Skill tokens, and q denotes the query. After C is removed, the remaining positions constitute the retained set S.

Let $Z := N(X)$ be the normalized residual stream, $V = ZW_V$ the value states, and A the causal attention matrix, so that the head output is $H = AV$. For a retained position t, we split its attention row between the Skill positions C and the retained positions S. Let $m_t = \sum_{j \in C} A_{tj}$ be the total attention mass on the Skill, $\bar{c}_t$ be the attention-weighted average value read from those positions and $\widetilde{B}_{t,:}$ be the attention weights on S renormalized to sum to one, we have

$$h_t^c = (1 - m_t)\widetilde{B}_{t,:}V_S + m_t\bar{c}_t.$$

Collecting the per-position decompositions into a single matrix expression yields, we have

$$H_S^c = (I - M)\widetilde{B}V_S + M\bar{C}, \qquad M := \text{diag}(m_t), \tag{1}$$

where $\bar{C}$ stacks the Skill-value readouts $\bar{c}_t$. By construction, $\widetilde{B}$ has unit row sums. Causality gives $m_t = 0$ on persistent rows and $m_t < 1$ on the remaining retained rows, so $I - M$ is invertible.

2.2 The Shared Update Relation

We consider a matched-state counterfactual in which the retained states Z_S from the Skill-conditioned execution remain fixed, while the Skill positions are removed and the attention routing of this head is recomputed over the remaining positions. The resulting routing matrix is denoted by B^0. This construction defines a *matched-state counterfactual*: the states entering the attention site remain fixed, and only the routing induced by Skill removal changes.

Under this routing, let $V^\star$ denote the values required to recover the original Skill-conditioned head output. Since causal attention makes B^0 lower triangular with strictly positive diagonal,

$$B^0V^\star = H_S^c \qquad \Longrightarrow \qquad V^\star = (B^0)^{-1}H_S^c.$$

The required value displacement is therefore $D_V := V^\star - V_S$. It combines the two effects isolated in Eq. 1, the contribution read directly from the Skill tokens and the change in retained-token routing after those tokens are removed. We now ask whether one fixed parameter update can produce this displacement. Before the update, the retained values are $V_S = Z_SW_V$. After applying ΔW_V,

$$V_S' = Z_S(W_V + \Delta W_V) = V_S + Z_S\Delta W_V.$$

Matching the required values $V^\star$ therefore requires $Z_S\Delta W_V = D_V$. Stacking the retained positions over all calibration queries gives

$$\mathbf{Z}\,\Delta W_V = \mathbf{D}_V, \tag{2}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times d}$, $\mathbf{D}_V \in \mathbb{R}^{n \times d_v}$, and $\Delta W_V \in \mathbb{R}^{d \times d_v}$. Equation 2 is the shared-update problem underlying our analysis. Exact shared control is possible if and only if every column of $\mathbf{D}_V$ lies in $\text{col}(\mathbf{Z})$. We formalize this condition and provide its proof in Appendix E.

The control target serves only structural analysis, while the deployed adapter is optimized for the task objective. The matched-state counterfactual is exact at the entrance layer but only a local approximation at deeper layers, where retained states have evolved without the skill. Appendix L quantifies this approximation gap. Because B^0 may be ill-conditioned despite being invertible, we evaluate all reported quantities through least-squares residuals without explicitly computing $(B^0)^{-1}$.

With the shared-update relation in place, three limits remain: whether the calibration states determine the shared map, whether the architecture permits the required shared control, and how accurately a rank-r update can represent it. We study these three limits from Sections 2.3 to 2.5 in turn .

2.3 Finite Calibration Fixes What Is Identifiable

Under fixed positional and attention-support conventions, the normalized state at a position in a causal decoder is determined by its causal prefix. Thus, two calibration positions sharing the same prefix contribute identical rows to $\mathbf{Z}$. After collapsing duplicate rows, at most N_{pre} distinct row vectors remain, where N_{pre} is the number of distinct causal prefixes in the calibration set. Since $\mathbf{Z}$ has d columns, its row-space dimension is bounded by both N_{pre} and d.

Proposition 1 (Causal-prefix bound on calibration rank) *For the stacked calibration states $\mathbf{Z} \in \mathbb{R}^{n\times d}$,*

$$\mathrm{rank}(\mathbf{Z}) \le \min\{N_{\mathrm{pre}}, d\}. \tag{3}$$

The bound is attained in the non-degenerate case.

This bound explains why insufficient calibration appears first as *ambiguity*, rather than fitting error. When $\mathrm{rank}(\mathbf{Z}) < d$, different parameter updates can agree on every observed calibration state while differing along directions that the calibration set never probes. Adding another occurrence of an existing prefix only repeats a row of $\mathbf{Z}$ and cannot remove this ambiguity; only new causal prefixes can increase the available rank.

Appendix K gives the non-degeneracy conditions and the full argument. Because N_{pre} is determined entirely by the tokenized calibration set, it is available before any forward pass or adapter training and serves as the calibration-side signal in Section 3.

2.4 Parameter Sharing Fixes What Is Realizable

The target constructed in Section 2.2 is head-local. For each head h, the Skill-removed routing B_h^0 and the original Skill-conditioned output H_h^c determine a required value matrix

$$B_h^0 V_h^\star = H_h^c, \qquad V_h^\star = (B_h^0)^{-1} H_h^c.$$

Grouped-query attention imposes an additional constraint: all query heads in a KV group $\mathcal{H}_g$ share one value projection, so exact realization requires a single value matrix shared across the group $V_g^\star$. At the activation level, this is possible if and only if

$$V_h^\star = V_{h'}^\star, \qquad h, h' \in \mathcal{H}_g. \tag{4}$$

When the head-local targets agree, their common value provides an exact activation-level solution for the group. Realizing that common value through an actual parameter update additionally requires the shared-update relation in Eq. 2. We give the joint activation- and parameter-level conditions in Appendix E.5.

This condition exposes an architectural limitation distinct from the calibration limitation above. When heads sharing a value projection demand different $V_h^\star$, any shared update must compromise between them. This leaves an error floor that cannot be removed simply by adding more calibration data. We measure this compatibility gap on real model activations in Section 4.2.

2.5 Rank Fixes What Is Representable

Even when the target is compatible with the shared parameterization, a rank-r adapter cannot realize an arbitrary shared map. For a fixed calibration problem, we write the best rank-r control error by

$$E_r := \min_{\mathrm{rank}(\Delta W_V)\le r} \|\mathbf{Z}\Delta W_V - \mathbf{D}_V\|_F .$$

Writing $P_{\mathbf{Z}}$ for the orthogonal projector onto $\mathrm{col}(\mathbf{Z})$. Every realizable control $\mathbf{Z}\Delta W_V$ lies in this subspace, whereas the target decomposes orthogonally as

$$\mathbf{D}_V = P_{\mathbf{Z}}\mathbf{D}_V + (I - P_{\mathbf{Z}})\mathbf{D}_V.$$

Orthogonality then gives,

$$\|\mathbf{Z}\Delta W_V - \mathbf{D}_V\|_F^2 = \|\mathbf{Z}\Delta W_V - P_{\mathbf{Z}}\mathbf{D}_V\|_F^2 + \underbrace{\|(I - P_{\mathbf{Z}})\mathbf{D}_V\|_F^2}_{e_{\text{img}}^2}.$$

The second term is an unavoidable compatibility floor. For the first term, $\operatorname{rank}(\mathbf{Z}\Delta W_V) \le r$, so the best attainable approximation is the best rank-r approximation to $P_{\mathbf{Z}}\mathbf{D}_V$. By the Eckart–Young–Mirsky theorem, the residual is exactly the spectral tail beyond the leading r singular directions.

Proposition 2 (Optimal rank-r control error) *Under the constraint* $\operatorname{rank}(\Delta W_V) \le r$,

$$E_r^2 = e_{\text{img}}^2 + \sum_{j>r} \sigma_j(P_{\mathbf{Z}}\mathbf{D}_V)^2 . \tag{5}$$

The two terms capture distinct limitations. The compatibility term e_{img} is independent of rank, while the spectral tail is the part that can be reduced by increasing the rank budget. Thus, for fixed $\mathbf{Z}$ and $\mathbf{D}_V$, additional rank stops buying approximation capacity once the remaining spectral energy is negligible. Changing the calibration set instead changes $\mathbf{Z}$, $\mathbf{D}_V$, and hence the control spectrum itself; it defines a new approximation problem rather than relaxing the rank constraint of the old one.

Appendix E gives the full proof and constructs an update that attains the bound. Finally, because $\operatorname{rank}(\mathbf{Z}\Delta W_V) \le \min\{\operatorname{rank}(\mathbf{Z}), \operatorname{rank}(\Delta W_V)\}$, the controllable rank is bounded by $\min\{\operatorname{rank}(\mathbf{Z}), r\}$. If either calibration rank or adapter rank is the bottleneck, increasing the other alone cannot enlarge the controllable subspace. Section 3 uses these two quantities to prioritize candidate calibration sets and ranks before training.

3 Preflight: Ordering Compilations Before Training

The three structural results support different components of the algorithm. Equation 3 provides the basis for Stage I by distinguishing calibration sets before any forward pass, while Equation 5 provides the basis for Stage II by predicting how approximation capacity varies across layers and ranks. Eq. 4 assesses a compatibility imposed by parameter sharing, and we retain it as a structural diagnostic.

A candidate compilation is a triple $m = (\mathcal{Q}, \ell, r) \in \mathcal{M} = \mathfrak{Q} \times \mathcal{L} \times \mathcal{R}$, comprising a calibration subset, an update layer, and a rank. We fix the parameter route to the value projection throughout. Accordingly, ℓ indexes the Transformer layer receiving the update, while r denotes its rank. Evaluating a candidate entails training the corresponding adapter and decoding on a held-out test set. Preflight instead orders all candidates in $\mathcal{M}$ using a key computed before any adapter is trained,

$$K(\mathcal{Q}, \ell, r) = \big(\tau_{\text{cal}}(\mathcal{Q}), \tau_{\text{rank}}(\ell, r), \widehat{E}^{\text{probe}}_{\ell,r}, \xi\big),$$

where the calibration term τ_{cal} orders a binary verdict on the calibration planes, while τ_{rank} and $\widehat{E}^{\text{probe}}$ order the (ℓ, r) cells within each plane. The term ξ breaks exact ties within each Skill seed. Alternative projection routes and multilayer adapters are evaluated separately in Appendix V.

The key is computed in two stages. In Stage I, we draw 1000 random calibration subsets of equal size and mark a subset as DEFICIENT if its causal-prefix complexity falls below 0.90 times the median across subsets. This tokenizer-only test determines $\tau_{\text{cal}}(\mathcal{Q})$ and orders the calibration planes, with deficient subsets placed after sufficient ones. In Stage II, we evaluate Eq. 5 once for each (ℓ, r) cell using a fixed probe stratification of 64 modal-length examples, independent of the candidate calibration subset. This procedure yields $\tau_{\text{rank}}(\ell, r)$ and $\widehat{E}^{\text{probe}}_{\ell,r}$, which order the layer–rank cells within each calibration plane. A rank is considered dominated if a smaller rank achieves a predicted error within a factor of 1.05 while the remaining cells are ordered by predicted error.

The capacity estimates used for Stage II ordering follow two implementation conventions. First, the control spectrum is computed separately for each KV group, and the groupwise errors are summed. Independent rank-r updates across groups can therefore yield a combined rank of up to $4r$, making $\widehat{E}^{\text{probe}}$ an *optimistic* estimate of the error achievable by a true combined rank-r adapter. Second, the joint problem retains one design matrix per group rather than reducing all groups to a single SVD (Appendix Q). The tie-breaking term ξ resolves exact ties from a fixed random draw, and the ordering is computed separately within each Skill seed.

Algorithm 1 Preflight ordering.

Require: frozen model θ, Skill c, calibration pool, menu $\mathcal{M}$
Ensure: priority order π over $\mathcal{M}$
1: **for** each calibration subset $\mathcal{Q}$ **do**
2: $\quad \tau_{\mathrm{cal}}(\mathcal{Q}) \leftarrow$ SUFFICIENT/DEFICIENT ▷ Stage I, tokenizer only
3: **end for**
4: **for** each (layer, rank) cell u **do**
5: $\quad (\tau_{\mathrm{rank}}, \widehat{E}^{\mathrm{probe}})(u) \leftarrow$ Eq. 5 ▷ Stage II, one capture + SVD
6: **end for**
7: **return** SORT($\mathcal{M}$, key K) ascending

The two terms act on different axes and can compose. τ_{cal} is constant in (ℓ, r) and separates the calibration planes: it moves every candidate built on an under-informative calibration set behind every candidate built on a sufficient one, without needing to know where the update will go. $\widehat{E}^{\mathrm{probe}}$ is constant in $\mathcal{Q}$ and orders within a plane: for each layer it says where along the rank axis the spectrum has flattened, so ranks that cannot buy additional approximation capacity are visited last. Both are available before candidate training: Stage I is computed from the candidate calibration set, whereas Stage II is computed from one fixed activation probe.

Realizability as a structural diagnostic. Equation 4 tests whether the virtual values associated with different heads are compatible with parameter sharing. It is therefore a feasibility test: it asks whether a shared projection can realize the required control, rather than which calibration set, layer, or rank should be prioritized. We consequently exclude it from the ordering defined by Eq. 3. The effect of incorporating its residual into the ordering is evaluated in Section 4.2.

Cost. The preflight overhead is modest. Stage I is purely token-based and requires no model evaluation. Stage II requires a single activation capture at the candidate layers, followed by an SVD of the resulting control matrix. For the menu considered here, the two stages together cost roughly three candidate-training runs, compared with the ninety runs required for exhaustive evaluation. Section 4.3 accounts for this overhead when quantifying the net advantage of preflight.

4 EXPERIMENTS

Our experiments connect the structural analysis to the deployed compilation problem. We establish that a Skill can be removed from the prompt while its behavior is retained in weights, test whether the quantities derived in Section 2 are measurable in frozen models, and then use those measurements to prioritize candidate compilations before training.

We conduct experiments on seven controlled Skill families spanning procedural policies, rule transformations, and retrieval tasks. We use a conditional tool-use policy for exhaustive compilation, runtime, and search experiments, and a copy-and-transform task for an independent prospective replication. The remaining families probe the boundary cases of the structural analysis. Unless otherwise stated, behavioral experiments use Gemma-2-2B (Team et al., 2024). Appendix P and Appendix M provide the task definitions and the full experimental protocol, respectively. Exact behavioral Skill instances and the public-Skill inventory are provided in Appendix W.

4.1 COMPILATION PRESERVES CAPABILITY AND REMOVES RECURRING CONTEXT

We first ask whether compilation produces a valid deployment endpoint: the Skill should be absent from the prompt while the capability it specifies remains available. We evaluate the merged fp32 model with free-running greedy decoding and exact match on 128 sealed examples. The reference target is produced by the deterministic Skill interpreter I_c, rather than by the Skill-conditioned model itself. Table 1 therefore compares three ways of reaching the same target behavior: the base model, the base model with the Skill supplied in context, and the compiled model with the Skill removed.

Across the three sealed seeds, the compiled model reaches a mean exact match of 0.802, compared with 0.523 when the Skill is supplied in context and 0 for the base model. The compiled score can

Table 1: **Compilation preserves held-out behavior while removing recurring context cost.** (a) Exact match on three sealed Skill seeds on behaviour endpoint. (b) Runtime cost with the Skill supplied in context versus compiled weights .

Condition	S1	S2	S3	Mean
Base	0.000	0.000	0.000	0.000
Skill in context	0.367	0.469	0.734	0.523
Compiled	**0.789**	**0.844**	**0.773**	**0.802**

Metric	Skill	Compiled
Prompt tokens	263.9	23.9
Batch-1 latency (ms)	33.0	31.7
Batch-64 queries/s	109	**1132**
Peak memory (GB)	22.7	**6.5**

exceed the in-context score because both are evaluated against the interpreter-defined target: the Skill-conditioned model is a deployment baseline, not the teacher that supplies the training targets.

Compilation also removes the recurring prompt cost of the Skill, reducing the average prompt length from 263.9 to 23.9 tokens. The shorter context has little effect on isolated batch-1 latency, where model execution dominates the saved prefill work, but becomes substantial at higher utilization. At batch 64, prefill throughput increases from 109 to 1132 queries/s and peak memory falls from 22.7 to 6.5 GB.

Using the measured median end-to-end compilation cost, including preflight, the batch-64 throughput saving offsets this one-time cost after approximately 6,400 individual queries. This break-even point is specific to the measured batch-64 regime; at batch 1, the latency saving is too small to provide a comparable practical amortization.

4.2 The structural quantities are measurable before training

Having established the behavioral endpoint, we next test whether the quantities derived in Section 2 can actually be read from a frozen model before any candidate is trained. Figure 2 evaluates the three structural predictions on a six-model ladder spanning 2B to 70B parameters across three model families.

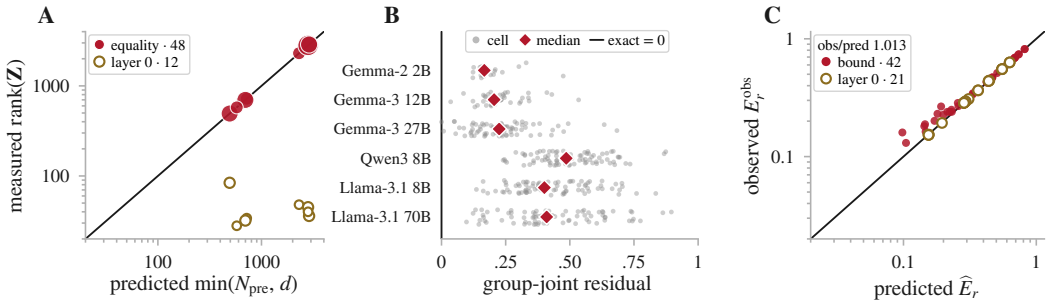


Figure 2: **The three structural predictions, measured on frozen models before any adapter exists.** (A) Eq. 3 across the six-model ladder; marker area is cells per point, and all twelve misses are layer 0, where the design matrix is degenerate. (B) The group-joint residual of Eq. 4 on every testable cell: the group-shared exact-control condition is violated on every testable model. (C) Eq. 5, no free parameters; the 21 layer-0 points, separately diagnosed as not capacity-bound, are drawn open rather than pooled into the median.

We first test the calibration prediction. Across the six-model ladder, Eq. 3 holds on all 60 testable cells and is tight on 48. On the cells where the prefix-based and nominal-row predictions differ, causal-prefix complexity tracks the transition in 8 of 12 cases, while nominal row count does so in none. This is the information Stage I obtains directly from the tokenized inputs.

A second effect comes from parameter sharing. The group-shared exact-control condition in Eq. 4 is violated throughout the ladder, with median residuals ranging from 0.167 to 0.486. The architecture-dependent realizability floor is therefore measurable before training. Its magnitude provides a direct pre-training measure of how strongly parameter sharing constrains the target control.

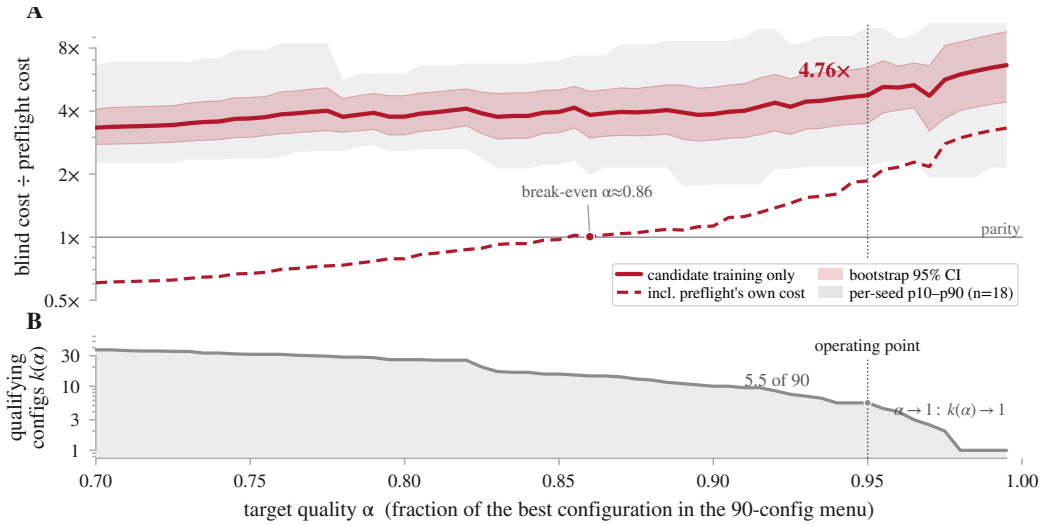


Figure 3: **Preflight reaches high-quality configurations after substantially less candidate training, and the advantage grows as the target tightens.** (A) Paired cost ratio over 18 Skill seeds (gray: per-seed p10–p90; red: bootstrap 95% CI on the geometric mean). The dashed curve charges preflight's own computation in full and crosses parity at $\alpha \approx 0.86$, below which the measured cost of blind search is lower than the cost of preflight. (B) The qualifying set $k(\alpha)$ shrinks with α, which is why the advantage grows and why $\alpha \to 1$ becomes an argmax search.

Finally, the rank-limited prediction is quantitatively accurate where capacity is the active constraint. Across the 42 capacity-bound points, the median ratio between observed and predicted residual is 1.013. More importantly for preflight, the predicted errors preserve the ordering of configurations by their observed errors, with Spearman correlations between 0.85 and 0.98. Together, these measurements show that the information used to construct the ordering is already present before optimization begins.

The same structural picture is not confined to our controlled generators. We additionally measure fourteen independently authored Skills from Super-NaturalInstructions (Wang et al., 2022) and five from BIG-bench (Srivastava et al., 2022). The answer-axis zero-collision condition used by Stage I holds for every Skill, while prompt-prefix duplication varies primarily with corpus construction rather than with whether a Skill is controlled or publicly authored. Appendix N reports the full structural distributions, and Appendix W lists the exact task identities and reproduces the behavioral Skill instances used in the main experiments.

4.3 Preflight reduces configuration-search cost

The preceding section established that the quantities introduced in Section 2 are measurable. We next examine whether ordering candidate compilations according to these quantities can reduce the cost of identifying a high-quality compilation.

We exhaustively train a 90-configuration menu spanning calibration subsets, layers, and ranks across 18 Skill seeds, yielding 1,656 trained candidate runs. Because every candidate is observed, any visiting order can be replayed exactly rather than estimated. For a target quality $\alpha S^\star$, we measure the GPU time required to reach the first candidate meeting that threshold and use random visiting orders as the blind-search baseline. The resulting search costs across target qualities are shown in Fig. 3.

At the 95% target-quality level, preflight reaches a qualifying candidate with 4.76× less candidate-training GPU time than blind search, with a 95% confidence interval of 3.49–6.46×. Since the one-time activation capture and SVD cost 0.0071 GPU-h per Skill, our preflight search method reduces the end-to-end advantage to 1.86×. The structural analysis reaches break-even once the target quality exceeds approximately $\alpha \approx 0.86$.

The exhaustive grid establishes the effect under complete replay, but the ordering is intended to be chosen before candidate training begins. We therefore freeze the priority order in advance and

repeat the evaluation on prospective seeds, a second Skill family, and two additional 8B architectures, Qwen3-8B (Yang et al., 2025) and Llama-3.1-8B (Grattafiori et al., 2024).

Table 2: **The ordering persists under prospective evaluation across Skill families and model architectures.** Every priority order was fixed before that setting's adapters were trained.

Setting	Model	Menu	Seeds	Cand.-training (95%)	Reversed
Primary family	Gemma-2-2B	90	15	$5.01\times$	$0/15$
Second family	Gemma-2-2B	30	10	$5.82\times$	$0/10$
Second family	Qwen3-8B	30	3	$4.87\times$	$0/3$
Second family	Llama-3.1-8B	30	3	$1.74\times$	$1/3$

Every ordering in Table 2 was frozen before its first candidate trained. The reduction persists across the primary study, a second Skill family, and two additional 8B architectures, requiring $1.74\times$ to $5.82\times$ less candidate-training GPU time. Across these settings, the spectral capacity term is the most stable component; Appendix U gives the component-wise breakdown. The controlled menu isolates calibration, layer, and rank at a fixed value-projection route, while Appendix V shows that compilation itself also extends to other projections and multilayer updates.

5 Conclusion and Discussion

We cast the compilation of a textual Skill into weights as a shared-control problem over attention values, and derived three structural limits on whether a candidate compilation can recover the Skill's behavior: identifiability from finite calibration data, realizability under grouped-query parameter sharing, and low-rank approximation. All three are properties of the frozen model, the Skill and the calibration set, and all three are measurable on frozen models from 2B to 70B. Two of them order candidate compilations before any adapter is trained. On a fully enumerated 90-configuration menu over 18 Skill seeds, the resulting preflight reaches 95% of the best configuration with $4.76\times$ less candidate-training GPU time than random search, or $1.86\times$ once its own computation is charged.

Calibration set, update layer and rank are normally treated as optimization hyperparameters, discovered by running the optimization (Hu et al., 2022; Zhang et al., 2023). Our results show that they leave measurable signatures in the frozen model before any of it runs: identifiability can be read from the tokenizer alone, and capacity from a single activation capture. Realizability is the exception, and an instructive one. The grouped-query condition is violated on every model we test, yet ordering candidates by the size of that violation roughly doubles search cost, because the layer with the smallest group-shared residual contains the best configuration on no seed. The derivation identifies which structural limits exist, but only measurement shows which of them is useful for ordering. Structural fidelity and cheap compilation are different objectives.

5.1 Limitations

Our analysis assumes fixed positional and attention-support conventions and uses a matched-state counterfactual to define the local control induced by Skill removal. This construction is exact at the entrance layer and becomes approximate at deeper layers as the retained hidden states change with context. The current theory also focuses on shared value projections, leaving more general parameter-sharing patterns and coupled multilayer updates for future analysis. Empirically, we validate the structural quantities across multiple model scales, controlled Skill families, and publicly authored Skills, while the behavioral preflight study remains centered on a predefined single-site value-projection search space. Although richer projection routes and multilayer updates also support effective compilation, ranking such coupled configurations before training remains open. Finally, preflight measures structural control difficulty in the frozen model, whereas the deployed adapter is optimized for task behavior, so the connection between structural ordering and final behavioral performance is currently supported primarily by empirical evidence.

AI USE STATEMENT

We used generative AI tools for language polishing, translation, broad literature search, and assistance with selected parts of the code. The theoretical framework, mathematical claims, research hypotheses, proofs, and interpretation of the results were developed by the authors without generative AI assistance. AI was also not used to generate, label, clean, or reformat the datasets used in this work. Literature suggested during the search process was checked by the authors before citation, and any AI-assisted code was reviewed and tested against the intended implementation. We also manually reviewed and revised all AI-polished text before submission. The authors take full responsibility for the final manuscript and the accompanying research artifacts.

REFERENCES

Agent Skills. Agent skills: Specification and documentation. Open specification, 2025. https://agentskills.io.

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

Anthropic. Agent skills. Claude Platform Documentation, 2025. https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, 2022.

Brian K Chen, Tianyang Hu, Hui Jin, Hwee Kuan Lee, and Kenji Kawaguchi. Exact conversion of in-context learning to model weights in linearized-attention transformers. In *International Conference on Machine Learning*, 2024.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. Learning without training: The implicit dynamics of in-context learning. *arXiv preprint arXiv:2507.16003*, 2025.

Adrian Goldwaser, Michael Munn, Javier Gonzalvo, and Benoit Dherin. Equivalence of context and parameter updates in modern transformer blocks. *arXiv preprint arXiv:2511.17864*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Hanna Mazzawi, Benoit Dherin, Michael Munn, Adrian Goldwaser, Michael Wunder, and Javier Gonzalvo. Transmuting prompts into weights. *arXiv preprint arXiv:2510.08734*, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations*, 2024.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 2023.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 5085–5109, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

A Related Work

Context-to-weight transfer. Transformers can implement gradient-descent-like learning in the forward pass (Akyürek et al., 2023; von Oswald et al., 2023), a context's action on attention or on a full block can be written as an implicit or explicit parameter update (Chen et al., 2024; Dherin et al., 2025; Goldwaser et al., 2025), and recent methods fit reusable static patches from calibration examples (Dherin et al., 2025; Mazzawi et al., 2025). These results establish that the transfer exists. Which of the many possible compilations of a given Skill to attempt is a separate question, and it is the cost this paper targets.

Neighbouring goals. Compression shortens the context but keeps it in the input (Lester et al., 2021; Mu et al., 2023); context distillation trains a model that no longer needs it (Snell et al., 2022); task and function vectors ask what a task's representation is and how it composes (Ilharco et al., 2023; Hendel et al., 2023; Todd et al., 2024); editing writes individual associations into weights under locality constraints (Meng et al., 2022; Mitchell et al., 2022). We remove the Skill entirely, supervise from a deterministic interpreter, and ask a cost question about the resulting artifact: given that a Skill will be compiled, which configuration should be tried first.

Adaptive parameter-efficient methods. Low-rank adapters make an update cheap (Hu et al., 2022) and adaptive variants allocate rank or placement by observing training (Zhang et al., 2023). The quantities used here come from the calibration set and one activation capture before any adapter exists, which is what lets them rank candidates rather than adjust a run in progress.

B Task Definition and Compilation Protocol

B.1 Task Readout and Functional Risk

Let $\mathcal{M}_\theta(x)$ denote the output of the frozen language model on an input sequence x, and let ρ be the fixed task-specific readout used for evaluation. We write

$$f_\theta(x) = \rho(\mathcal{M}_\theta(x)) \,. \tag{6}$$

The Skill c specifies a deterministic interpreter I_c, and the compilation target for a query $q \sim \mathcal{P}_c$ is that interpreter's output

$$y_c(q) = I_c(q). \tag{7}$$

A compiled parameter update Δ is evaluated by

$$\mathcal{R}_c(\Delta) = \mathbb{E}_{q\sim\mathcal{P}_c}\left[\ell(f_{\theta+\Delta}(q), I_c(q))\right], \tag{8}$$

where ℓ is the task loss used for the corresponding evaluation. The Skill-conditioned execution $f_\theta(c \oplus q)$ is therefore *not* the functional target: it is a runtime reference trajectory, and the source of the structural activations from which the control of Appendix E is read. Compiling against I_c rather than against $f_\theta(c \oplus q)$ is what keeps teacher competence from bounding the compiled model, and the deployment endpoint measures the gap directly (Appendix T). The architecture-level results in Section 2.5 construct sufficient control targets whose exact realization makes the corresponding model states, and hence the fixed readout, agree.

B.2 Compilation Protocol

For each Skill c, let

$$\mathcal{Q}_{\mathrm{cal}} = \{q_1, \ldots, q_n\}$$

be the calibration queries available during compilation, and let $\mathcal{Q}_{\mathrm{test}}$ denote held-out deployment queries drawn from the same task distribution $\mathcal{P}_c$. The compiler receives

$$(\theta, c, \mathcal{Q}_{\mathrm{cal}}) \longmapsto \Delta_c = G(\theta, c, \mathcal{Q}_{\mathrm{cal}}). \tag{9}$$

On $\mathcal{Q}_{\mathrm{cal}}$, the compiler may evaluate both

$$f_\theta(c \oplus q) \qquad \text{and} \qquad f_\theta(q),$$

and may access their intermediate activations. For the layerwise control targets in Section 2.5, we use a matched-state counterfactual. At layer ℓ, the retained-token state is taken from the Skill-conditioned trajectory, $X^c_{\ell,S}$, while the removable tokens C are excluded from the attention support and the retained tokens use the deployment positional convention. We denote the resulting retained-only attention operator by $B^0_\ell(q)$. Under the layerwise construction of Appendix I, this is exactly the attention problem presented to layer ℓ once all preceding retained states have been matched.

Once Δ_c is produced, it is fixed. Deployment evaluates

$$f_{\theta+\Delta_c}(q), \qquad q \in \mathcal{Q}_{\text{test}}, \tag{10}$$

without providing either c or $\mathcal{Q}_{\text{cal}}$, and without recomputing Δ_c from the deployment query. Thus the same update is shared across all deployment queries. The Skill-only protocol $G(\theta, c)$ is the special case in which no calibration queries are used.

Calibration queries used to construct or select the compiler are disjoint from the held-out queries used for final evaluation.

B.3 Adapter Conventions

For a parameter block $W_\mu \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, we analyze its effective parameter change ΔW_μ. A rank-r adapter denotes

$$\text{rank}(\Delta W_\mu) \le r. \tag{11}$$

A LoRA parameterization

$$\Delta W_\mu = B_\mu A_\mu, \qquad B_\mu \in \mathbb{R}^{d_{\text{in}} \times r}, \quad A_\mu \in \mathbb{R}^{r \times d_{\text{out}}}, \tag{12}$$

parameterizes exactly the family of matrices with rank at most r, up to the non-uniqueness of the factorization.

For a multilayer compiler, Δ denotes the collection of all modified blocks,

$$\Delta = \{\Delta W_{\mu,\ell}\}_{(\mu,\ell)\in\mathcal{I}},$$

while route-specific statements fix all blocks outside the stated route.

When LoRA is attached to a combined projection, its rank is the rank of the full combined update. In particular, for

$$W_V \in \mathbb{R}^{d \times H_{kv} d_v},$$

a rank-r adapter constrains the combined matrix ΔW_V, rather than the maximum rank of its individual head or KV-group slices. The corresponding GQA consistency conditions and combined-projection control ranks are derived in Appendix E.

C Attention Algebra

Consider a pre-norm attention layer with

$$X \in \mathbb{R}^{T \times d}, \qquad Z = N(X),$$

and one attention head

$$Q = ZW_Q, \qquad K = ZW_K, \qquad V = ZW_V.$$

For scaled dot-product attention,

$$A = \text{softmax}_{\text{row}}\left(\frac{QK^\top}{\sqrt{d_k}} + B_{\text{pos}} + M_{\text{mask}}\right), \qquad H = AV.$$

Let $W_O^{(h)} \in \mathbb{R}^{d_v \times d}$ denote the slice of the output projection corresponding to this head, so that

$$O_h = H_h W_O^{(h)}.$$

C.1 HEAD-LOCAL PARAMETER ROUTES

Output projection. With H fixed, a finite update to W_O gives

$$\Delta O = H \Delta W_O. \tag{13}$$

If

$$\Delta W_O = \sum_{j=1}^{r} u_j v_j^\top,$$

then

$$\Delta o_t = \sum_{j=1}^{r} \langle h_t, u_j \rangle v_j^\top.$$

Thus a rank-r output update uses at most r fixed residual write directions.

Value projection. If only W_V changes, Q, K, and A remain fixed at this attention site, and

$$\Delta O = AZ \Delta W_V W_O. \tag{14}$$

For

$$\Delta W_V = \sum_{j=1}^{r} u_j v_j^\top,$$

$$\Delta o_t = \sum_{j=1}^{r} \left(\sum_s A_{ts} \langle z_s, u_j \rangle \right) (v_j^\top W_O),$$

and therefore

$$\Delta o_t^V \in \mathrm{row}(W_O).$$

Query and key projections. At a fixed state, the score changes induced by W_Q and W_K are

$$\delta S_Q = \frac{Z \Delta W_Q K^\top}{\sqrt{d_k}}, \tag{15}$$

$$\delta S_K = \frac{Q (Z \Delta W_K)^\top}{\sqrt{d_k}}. \tag{16}$$

For rank-r updates,

$$\Delta W_Q = \sum_{j=1}^{r} u_j v_j^\top, \qquad \Delta W_K = \sum_{j=1}^{r} \tilde{u}_j \tilde{v}_j^\top,$$

their entries satisfy

$$\delta s_{ts}^Q = \frac{1}{\sqrt{d_k}} \sum_{j=1}^{r} \langle z_t, u_j \rangle \langle k_s, v_j \rangle, \tag{17}$$

$$\delta s_{ts}^K = \frac{1}{\sqrt{d_k}} \sum_{j=1}^{r} \langle q_t, \tilde{v}_j \rangle \langle z_s, \tilde{u}_j \rangle. \tag{18}$$

For an attention row a_t on support Ω_t, the softmax Jacobian is

$$C_t = \mathrm{Diag}(a_t) - a_t a_t^\top.$$

When all logits on Ω_t are finite,

$$\mathrm{rank}\, C_t = |\Omega_t| - 1.$$

Indeed, with $D_t = \mathrm{Diag}(a_t)$,

$$C_t = D_t^{1/2} \left(I - \sqrt{a_t} \sqrt{a_t}^\top \right) D_t^{1/2},$$

where D_t is invertible on Ω_t and the middle factor is the orthogonal projector onto the complement of $\sqrt{a_t}$.

Moreover,

$$C_t V_{\Omega_t} = \mathrm{Diag}(a_t) \left(V_{\Omega_t} - \mathbf{1} \bar{v}_t \right), \qquad \bar{v}_t = a_t^\top V_{\Omega_t}.$$

Hence the first-order Q/K output space satisfies

$$\dim \mathcal{U}_t^{Q/K} \le \min \left\{ |\Omega_t| - 1,\ \mathrm{rank}(V_{\Omega_t} - \mathbf{1} \bar{v}_t),\ \mathrm{rank}\, W_O \right\}. \tag{19}$$

C.2 Architecture-restricted low-rank lifting

Consider

$$Y = L\Delta R,$$

with

$$a = \operatorname{rank} L, \qquad b = \operatorname{rank} R, \qquad r' = \min\{r, a, b\}.$$

Proposition. The image of rank-r updates is exactly

$$\{Y : \operatorname{col}(Y) \subseteq \operatorname{col}(L),\ \operatorname{row}(Y) \subseteq \operatorname{row}(R),\ \operatorname{rank} Y \le r'\}. \tag{20}$$

Proof. Every $L\Delta R$ satisfies the three conditions. Conversely, for any Y satisfying them, choose

$$\Delta = L^{\dagger} Y R^{\dagger}.$$

Then

$$L\Delta R = LL^{\dagger} Y R^{\dagger} R = Y,$$

and

$$\operatorname{rank} \Delta \le \operatorname{rank} Y \le r'.$$

□

The corresponding rank-r' stratum has dimension

$$r'(a + b - r'). \tag{21}$$

This follows from the factorization $UV^{\top}$ with $U \in \mathbb{R}^{a \times r'}$ and $V \in \mathbb{R}^{b \times r'}$, modulo the r'^2-dimensional change of basis.

For the head-local V route,

$$L_V = \operatorname{stack}_q(A_q Z_q), \qquad R_V = W_O,$$

so

$$D_{V,r} = r'(a_V + b_V - r'), \qquad a_V = \operatorname{rank} L_V, \quad b_V = \operatorname{rank} W_O.$$

For the Q score route, with

$$a_Z = \dim \operatorname{span}\{z_{q,t}\}, \qquad a_K = \dim \operatorname{span}\{k_{q,s}\},$$

the score family satisfies

$$D^{\text{score}}_{Q,r} \le r'(a_Z + a_K - r'), \qquad r' = \min\{r, a_Z, a_K\},$$

before the additional contraction induced by the softmax tangent and the centered-value map.

D Context Decomposition

Let P denote persistent tokens, C the removable Skill tokens, and Q the deployment query tokens. Define

$$S = P \cup Q.$$

D.1 Causal-prefix independence

Proposition. For every Transformer layer ℓ,

$$X_{\ell, P\cup C}(P \oplus C \oplus q) = X_{\ell, P\cup C}(P \oplus C), \qquad q \in \mathcal{Q}, \tag{22}$$

provided the decoder is causal, normalization and MLP sublayers are token-wise, and the computation of the prefix positions does not depend on future sequence-level state.

Proof. The prefix embeddings are fixed at layer 0. Assume the claim at layer ℓ. By causality, every attention row indexed by $P \cup C$ can attend only to preceding positions in $P \cup C$, so its attention inputs and outputs are independent of Q. Residual updates, normalization, and the MLP act token-wise and preserve this independence. Induction over layers gives Eq. equation 22. □

Consequently,

$$(K_{\ell,C}, V_{\ell,C})$$

is query-independent at every layer.

D.2 EXACT $(C/P/Q)$ DECOMPOSITION

Restrict the context-conditioned attention output to query rows:

$$H_Q^c = A_{QS}^c V_S + A_{QC}^c V_C.$$

Define

$$m_t = \sum_{j \in C} A_{tj}^c, \qquad M = \mathrm{Diag}(m_t).$$

For $m_t > 0$, define

$$R_{tj} = \frac{A_{tj}^c}{m_t}, \qquad R\mathbf{1} = \mathbf{1}, \qquad \bar{C} = RV_C.$$

Rows of R for which $m_t = 0$ may be chosen arbitrarily. Define

$$\widetilde{B} = (I - M)^{-1} A_{QS}^c.$$

Whenever each query row contains at least one retained key, $m_t < 1$ and the inverse is well defined. Then

$$\widetilde{B}\mathbf{1} = \mathbf{1}.$$

Proposition.

$$\boxed{H_Q^c = (I - M)\widetilde{B}V_S + M\bar{C}.} \tag{23}$$

Proof. By construction,

$$A_{QS}^c = (I - M)\widetilde{B}, \qquad A_{QC}^c V_C = M\bar{C}.$$

Substitution into the expression for H_Q^c proves the claim. □

For persistent prefix tokens P that precede the removable Skill block C, causality gives

$$A_{PC}^c = 0, \qquad H_P^c = H_P^0.$$

Accordingly, in the layerwise construction below we use the retained-token target

$$H_S^c := \begin{bmatrix} H_P^0 \\ H_Q^c \end{bmatrix}, \qquad S = P \cup Q. \tag{24}$$

Thus the nontrivial context-removal target lies on the query rows, while persistent prefix rows are carried through unchanged.

D.3 SAME-STATE QUERY-ROUTING INVARIANCE

Suppose corresponding query tokens in the context-conditioned and compiled executions are shifted by a common positional offset δ. Write the pre-softmax score as

$$\ell_{ts} = \Psi(q_t, k_s, p_t, p_s, \eta).$$

Assume that corresponding pre-attention query states and Q/K parameters agree, and that for corresponding query pairs

$$\Psi(q, k, p + \delta, s + \delta, \eta_c) = \Psi(q, k, p, s, \eta_0), \tag{25}$$

with translation-equivariant query-query hard support.

Proposition. For any two corresponding allowed query keys s_1, s_2,

$$\frac{A_{t,s_1}^c}{A_{t,s_2}^c} = \frac{A_{t-\delta,s_1-\delta}^0}{A_{t-\delta,s_2-\delta}^0}. \tag{26}$$

Proof. The assumptions imply equality of the corresponding logits. Softmax preserves pairwise probability ratios because

$$\frac{a_i}{a_j} = \exp(\ell_i - \ell_j).$$

The support assumption ensures that the compared entries exist in both executions. □

When $P = \varnothing$, all retained keys are jointly shifted query tokens, and therefore

$$\widetilde{B} = B^0. \tag{27}$$

For standard fixed-frequency RoPE,

$$q_p = \hat{q}R_p, \qquad k_s = \hat{k}R_s,$$

and Eq. equation 25 follows from

$$R_{p+\delta}R_{s+\delta}^\top = R_p R_s^\top.$$

A causal sliding-window support

$$s \le t, \qquad t - s < w$$

is also invariant under a common shift.

E Shared Control Rank

E.1 Activation Virtualization

Fix a layer ℓ and a query q. Let

$$Z_S(q) = N\big(X^c_{\ell,S}(q)\big)$$

denote the normalized retained-token state of the Skill-conditioned trajectory. In the layerwise construction of Appendix I, this is also the state presented to layer ℓ by the compiled model once all preceding retained states have been matched.

Define

$$B^0(q) \in \mathbb{R}^{|S|\times|S|}$$

as the causal attention matrix obtained from this same retained state after removing the Skill tokens C from the attention support and applying the deployment positional convention. Thus the superscript 0 denotes context removal at the current attention site, rather than a free-running context-free trajectory. Define the corresponding baseline retained values by

$$V^0_S(q) := Z_S(q)W_V. \tag{28}$$

Causality and self-attention imply that $B^0(q)$ is lower triangular with

$$B^0_{tt}(q) > 0.$$

Hence

$$\det B^0(q) = \prod_t B^0_{tt}(q) > 0, \qquad B^0(q) \text{ is invertible.} \tag{29}$$

Given the retained-token target $H^c_S(q)$ from Eq. equation 24, define

$$V^\star(q) = (B^0(q))^{-1}H^c_S(q). \tag{30}$$

Then

$$B^0(q)V^\star(q) = H^c_S(q),$$

and the unique virtual-value correction at this attention site is

$$D_V(q) = V^\star(q) - V^0_S(q). \tag{31}$$

A fixed V update realizes this control on query q exactly when

$$Z_S(q)\Delta W_V = D_V(q). \tag{32}$$

Stacking the matched retained states and control targets over $q \in \mathcal{Q}$ gives

$$\mathbf{Z}\Delta W_V = \mathbf{D}_V. \tag{33}$$

E.2 FIXED-MAP LEMMA

Lemma. For arbitrary matrices X, W, D, the equation

$$XW = D$$

has a solution if and only if

$$XX^\dagger D = D. \tag{34}$$

When a solution exists,

$$\boxed{\min_{XW=D} \operatorname{rank} W = \operatorname{rank} D.} \tag{35}$$

Proof. Since $XX^\dagger$ is the orthogonal projector onto $\mathrm{col}(X)$, Eq. equation 34 is equivalent to $\mathrm{col}(D) \subseteq \mathrm{col}(X)$, which is the solvability condition for $XW = D$. Every solution satisfies

$$\operatorname{rank} D = \operatorname{rank}(XW) \le \operatorname{rank} W.$$

Conversely,

$$W^\star = X^\dagger D$$

is feasible under Eq. equation 34, and

$$\operatorname{rank} W^\star \le \operatorname{rank} D.$$

The two bounds coincide. □

E.3 EXACT V CONTROL

Define the exact V-control rank on a query family $\mathcal{Q}$ by

$$r_V^{\mathrm{ctrl}}(\mathcal{Q}) := \inf \left\{ \operatorname{rank}(\Delta W_V) : \mathbf{Z}\Delta W_V = \mathbf{D}_V \right\}, \tag{36}$$

with the convention $\inf \varnothing = +\infty$. This quantity is the minimum rank required to realize the architecture-derived value-control target exactly.

Let

$$P_{\mathbf{Z}} = \mathbf{Z}\mathbf{Z}^\dagger. \tag{37}$$

Theorem. An exact shared V update exists if and only if

$$\boxed{P_{\mathbf{Z}}\mathbf{D}_V = \mathbf{D}_V.} \tag{38}$$

Moreover,

$$\boxed{r_V^{\mathrm{ctrl}}(\mathcal{Q}) = \begin{cases} \operatorname{rank}(\mathbf{D}_V), & P_{\mathbf{Z}}\mathbf{D}_V = \mathbf{D}_V, \\ +\infty, & P_{\mathbf{Z}}\mathbf{D}_V \neq \mathbf{D}_V. \end{cases}} \tag{39}$$

In the finite case, a minimum-rank witness is

$$\Delta W_V^\star = \mathbf{Z}^\dagger \mathbf{D}_V. \tag{40}$$

Proof. Apply the fixed-map lemma to Eq. equation 33. □

E.4 GENERAL RANK-CONSTRAINED SHARED-MAP APPROXIMATION

For a matrix D, define

$$e_r(D)^2 = \sum_{j>r} \sigma_j^2(D),$$

and let

$$P_X = XX^\dagger$$

be the orthogonal projector onto $\mathrm{col}(X)$.

Proposition. For arbitrary matrices X and D and any $r \geq 0$,

$$\boxed{\min_{\mathrm{rank}(W)\leq r} \|XW - D\|_F^2 = \|(I - P_X)D\|_F^2 + e_r(P_X D)^2.} \tag{41}$$

Proof. For every W,

$$XW \in \mathrm{col}(X).$$

Since

$$D = P_X D + (I - P_X)D$$

is an orthogonal decomposition with respect to the row coordinates,

$$\|XW - D\|_F^2 = \|XW - P_X D\|_F^2 + \|(I - P_X)D\|_F^2. \tag{42}$$

The second term is independent of W.

Let $(P_X D)_r$ be a best rank-r approximation of $P_X D$. Since

$$\mathrm{col}((P_X D)_r) \subseteq \mathrm{col}(P_X D) \subseteq \mathrm{col}(X),$$

the matrix

$$W_r^\star = X^\dagger (P_X D)_r$$

satisfies

$$XW_r^\star = (P_X D)_r, \qquad \mathrm{rank}(W_r^\star) \leq r.$$

Conversely, $\mathrm{rank}(XW) \leq r$ whenever $\mathrm{rank}(W) \leq r$, so the Eckart–Young–Mirsky theorem gives

$$\min_{\mathrm{rank}(W)\leq r} \|XW - P_X D\|_F = e_r(P_X D).$$

Substitution proves Eq. equation 41. □

Corollary. The optimal unrestricted shared-map residual is

$$\boxed{\min_W \|XW - D\|_F = \|(I - P_X)D\|_F.} \tag{43}$$

The minimum rank required to attain this residual is

$$\boxed{\min\{\mathrm{rank}(W) : \|XW - D\|_F = \|(I - P_X)D\|_F\} = \mathrm{rank}(P_X D).} \tag{44}$$

When $P_X D = D$, Eq. equation 41 reduces to

$$\min_{\mathrm{rank}(W)\leq r} \|XW - D\|_F = e_r(D), \tag{45}$$

recovering the exact-fixedness result.

E.5 GQA and Combined Projections

Let $\mathcal{H}_g$ be the query heads sharing KV group g. For head h, let B_h^0 be the compiled retained-token routing and H_h^c the target head output before W_O. A V-only realization requires a single shared value matrix $V_g^\star$ satisfying

$$B_h^0 V_g^\star = H_h^c \qquad h \in \mathcal{H}_g.$$

Proposition. Exact V-only matching within group g requires and, at the activation level, is implied by

$$\boxed{(B_h^0)^{-1} H_h^c = (B_{h'}^0)^{-1} H_{h'}^c \qquad \forall h, h' \in \mathcal{H}_g.} \tag{46}$$

Proof. Each equation uniquely determines

$$V_g^\star = (B_h^0)^{-1} H_h^c.$$

A common $V_g^\star$ exists exactly when these values agree across the group. □

Let D_g denote the resulting group-wise value control. Parameter realizability further requires

$$P_{\mathbf{Z}} D_g = D_g.$$

For combined projections,

$$W_Q \in \mathbb{R}^{d \times H_q d_k}, \qquad W_K \in \mathbb{R}^{d \times H_{kv} d_k}, \qquad W_V \in \mathbb{R}^{d \times H_{kv} d_v}, \qquad W_O \in \mathbb{R}^{H_q d_v \times d}.$$

For combined W_V, define

$$D_{\text{all}} = [D_1, \ldots, D_{H_{kv}}].$$

Then

$$\mathbf{Z}\Delta W_V = D_{\text{all}},$$

so the fixed-map lemma gives

$$P_{\mathbf{Z}} D_{\text{all}} = D_{\text{all}} \tag{47}$$

as the exact solvability condition, and

$$\boxed{r^{\text{ctrl}}_{V,\text{combined}} = \operatorname{rank} D_{\text{all}}.} \tag{48}$$

Consequently,

$$\max_g \operatorname{rank} D_g \le r^{\text{ctrl}}_{V,\text{combined}} \le \sum_g \operatorname{rank} D_g.$$

For combined W_Q, if $\widetilde{U}_h$ is any feasible de-positioned head-local Q control field,

$$\mathbf{Z}\Delta W_Q = [\widetilde{U}_1, \ldots, \widetilde{U}_{H_q}],$$

and therefore

$$\boxed{r^{\text{ctrl}}_{Q,\text{combined}} = \min \operatorname{rank}[\widetilde{U}_1, \ldots, \widetilde{U}_{H_q}],} \tag{49}$$

where the minimum is over jointly feasible fields satisfying the shared fixed-map condition.

For combined W_O, let

$$H^0_{\text{cat}} = [H^0_1, \ldots, H^0_{H_q}], \qquad H^0_{\text{cat}} \Delta W_O = D_O.$$

The fixed-map lemma gives

$$P_{\mathbf{H}^0_{\text{cat}}} D_O = D_O$$

and, whenever this condition holds,

$$\boxed{r^{\text{ctrl}}_{O,\text{combined}} = \operatorname{rank} D_O.} \tag{50}$$

For a combined V projection, the post-W_O perturbation is generally

$$\Delta O = \sum_{h=1}^{H_q} A_h Z \Delta W_V^{g(h)} W_O^{(h)},$$

so the multi-head V route does not in general reduce to a single head-local factorization $L\Delta R$.

F NORMALIZATION DEBT

Assume $P = \varnothing$ and the same-state routing identity $\widetilde{B} = B$ from Eq. equation 27. Then

$$A^c_{QQ} = (I - M)B.$$

Define

$$H^0 = BV_Q.$$

The exact context decomposition becomes

$$H^c = (I - M)H^0 + M\bar{C},$$

hence

$$H^c - H^0 = M(\bar{C} - H^0). \tag{51}$$

Keeping B fixed, the virtual-value correction is

$$\boxed{D_V^{\text{virt}} = B^{-1}M(\bar{C} - H^0) = B^{-1}M\bar{C} - B^{-1}MH^0.} \tag{52}$$

Across the finite query family, let

$$\mathbf{D}_V = \mathcal{B}^{-1}\mathcal{M}(\bar{\mathbf{C}} - \mathbf{H}^0),$$

where $\mathcal{B}$ and $\mathcal{M}$ are the corresponding block-diagonal matrices. Define

$$\mathbf{C}_{\text{mem}} = \mathcal{M}\bar{\mathbf{C}}, \qquad \mathbf{D}_{\text{debt}} = \mathcal{M}\mathbf{H}^0,$$

and

$$r_{\text{mem}} = \operatorname{rank}\mathbf{C}_{\text{mem}}, \qquad r_{\text{debt}} = \operatorname{rank}\mathbf{D}_{\text{debt}}.$$

Since $\mathcal{B}^{-1}$ is invertible,

$$\boxed{\operatorname{rank}\mathbf{D}_V = \operatorname{rank}(\mathbf{C}_{\text{mem}} - \mathbf{D}_{\text{debt}}).} \tag{53}$$

F.1 EXACT RANK BOUNDS

Proposition.

$$\boxed{|r_{\text{debt}} - r_{\text{mem}}| \le \operatorname{rank}\mathbf{D}_V \le \min\{d_v, r_{\text{debt}} + r_{\text{mem}}\}.} \tag{54}$$

If $P_{\mathbf{Z}}\mathbf{D}_V = \mathbf{D}_V$, then

$$\boxed{|r_{\text{debt}} - r_{\text{mem}}| \le r_V^{\text{ctrl}} \le \min\{d_v, r_{\text{debt}} + r_{\text{mem}}\}.} \tag{55}$$

Proof. For arbitrary A, B,

$$\operatorname{rank}(A + B) \le \operatorname{rank}A + \operatorname{rank}B.$$

Applying this inequality to $A = (A + B) - B$ and to $B = (A + B) - A$ also gives

$$\operatorname{rank}(A + B) \ge |\operatorname{rank}A - \operatorname{rank}B|.$$

Use $A = \mathbf{C}_{\text{mem}}$ and $B = -\mathbf{D}_{\text{debt}}$ in Eq. equation 53. The second statement follows from Eq. equation 39. □

F.2 PERSISTENT-MASS LOWER BOUND

Assume

$$m_t > 0$$

for every relevant row. Then $\mathcal{M}$ is invertible. Since

$$\mathbf{H}^0 = \mathcal{B}\mathbf{V}_Q,$$

both $\mathcal{M}$ and $\mathcal{B}$ are invertible, and

$$r_{\text{debt}} = \operatorname{rank}\mathbf{V}_Q.$$

If

$$\operatorname{rank} \mathbf{V}_Q = d_v,$$

then

$$\boxed{\operatorname{rank} \mathbf{D}_V \geq d_v - r_{\text{mem}}.} \tag{56}$$

By causal-prefix independence, the same V_C is shared across queries, and

$$\mathbf{C}_{\text{mem}} = \mathcal{M}\mathbf{R}V_C$$

for the stacked retrieval matrix $\mathbf{R}$. Hence

$$r_{\text{mem}} \leq \operatorname{rank} V_C,$$

which yields

$$\boxed{\operatorname{rank} \mathbf{D}_V \geq d_v - \operatorname{rank} V_C.} \tag{57}$$

Under $P_{\mathbf{Z}}\mathbf{D}_V = \mathbf{D}_V$, the same bounds hold for r_V^{ctrl}.

F.3 Approximate spectral lower bound

Let

$$e_r(X)^2 = \sum_{j>r} \sigma_j^2(X).$$

Since $\operatorname{rank} \mathbf{C}_{\text{mem}} = r_{\text{mem}}$, the singular-value inequality

$$\sigma_{i+j-1}(A+B) \leq \sigma_i(A) + \sigma_j(B)$$

applied to

$$\mathbf{D}_{\text{debt}} = (\mathbf{D}_{\text{debt}} - \mathbf{C}_{\text{mem}}) + \mathbf{C}_{\text{mem}}$$

gives

$$\sigma_j(\mathbf{D}_{\text{debt}} - \mathbf{C}_{\text{mem}}) \geq \sigma_{j+r_{\text{mem}}}(\mathbf{D}_{\text{debt}}). \tag{58}$$

For invertible B,

$$\sigma_j(B^{-1}X) \geq \frac{\sigma_j(X)}{\|B\|_2}.$$

Therefore

$$\boxed{e_r(\mathbf{D}_V)^2 \geq \frac{1}{\|\mathcal{B}\|_2^2} \sum_{j>r+r_{\text{mem}}} \sigma_j^2(\mathbf{D}_{\text{debt}}).} \tag{59}$$

If, moreover,

$$m_t \geq m_{\min} > 0$$

for all relevant rows, then

$$\sigma_{\min}(\mathcal{M}) \geq m_{\min}$$

and

$$\mathbf{D}_{\text{debt}} = \mathcal{M}\mathcal{B}\mathbf{V}_Q.$$

Thus

$$\sigma_j(\mathbf{D}_{\text{debt}}) \geq m_{\min}\sigma_{\min}(\mathcal{B})\sigma_j(\mathbf{V}_Q).$$

Using

$$\sigma_{\min}(\mathcal{B}) = \|\mathcal{B}^{-1}\|_2^{-1},$$

we obtain

$$\boxed{e_r(\mathbf{D}_V)^2 \geq \frac{m_{\min}^2}{\kappa_2(\mathcal{B})^2} \sum_{j>r+r_{\text{mem}}} \sigma_j^2(\mathbf{V}_Q).} \tag{60}$$

F.4 SPARSE-DEBT UPPER BOUND

Let $\mathcal{S}$ be a set of token rows, and let $\mathbf{D}_{\mathrm{debt},\mathcal{S}}$ retain the rows of $\mathbf{D}_{\mathrm{debt}}$ in $\mathcal{S}$ and zero all remaining rows. Then

$$\operatorname{rank} \mathbf{D}_{\mathrm{debt},\mathcal{S}} \le |\mathcal{S}|.$$

Define

$$\widehat{\mathbf{D}}_V = \mathcal{B}^{-1}\left(\mathbf{C}_{\mathrm{mem}} - \mathbf{D}_{\mathrm{debt},\mathcal{S}}\right).$$

Since left multiplication by $\mathcal{B}^{-1}$ preserves rank,

$$\operatorname{rank} \widehat{\mathbf{D}}_V \le r_{\mathrm{mem}} + |\mathcal{S}|.$$

Moreover,

$$\|\mathbf{D}_V - \widehat{\mathbf{D}}_V\|_F = \left\|\mathcal{B}^{-1}(\mathbf{D}_{\mathrm{debt}} - \mathbf{D}_{\mathrm{debt},\mathcal{S}})\right\|_F \tag{61}$$

$$\le \|\mathcal{B}^{-1}\|_2 \left(\sum_{i\notin\mathcal{S}} m_i^2 \|h_i^0\|_2^2\right)^{1/2}. \tag{62}$$

Consequently,

$$\boxed{e_{r_{\mathrm{mem}}+|\mathcal{S}|}(\mathbf{D}_V) \le \|\mathcal{B}^{-1}\|_2 \left(\sum_{i\notin\mathcal{S}} m_i^2 \|h_i^0\|_2^2\right)^{1/2}.} \tag{63}$$

When $P_{\mathbf{Z}}\mathbf{D}_V = \mathbf{D}_V$, Eq. equation 45 converts this activation-field bound into the same rank-constrained shared-map error bound.

F.5 SPECTRAL EFFECT OF THE CAUSAL INVERSE

For any invertible B,

$$\frac{\sigma_j(X)}{\|B\|_2} \le \sigma_j(B^{-1}X) \le \|B^{-1}\|_2 \sigma_j(X), \tag{64}$$

and therefore

$$\frac{1}{\|B\|_2} e_r(X) \le e_r(B^{-1}X) \le \|B^{-1}\|_2 e_r(X). \tag{65}$$

For a causal lower-triangular B, write

$$B = D + L = D(I + N), \qquad N = D^{-1}L,$$

with D diagonal and L strictly lower triangular. For sequence length T, $N^T = 0$, so

$$\boxed{B^{-1} = \sum_{k=0}^{T-1} (-N)^k D^{-1}.} \tag{66}$$

If additionally $\|N\|_2 < 1$,

$$\|B^{-1}\|_2 \le \frac{\|D^{-1}\|_2}{1 - \|N\|_2} = \frac{1/\min_t b_{tt}}{1 - \|D^{-1}L\|_2}.$$

Since the eigenvalues of the triangular matrix B^{-1} include $1/b_{tt}$,

$$\|B^{-1}\|_2 \ge \max_t \frac{1}{b_{tt}}.$$

G EXACT ROUTING CONTROL

This section considers raw affine-logit softmax attention with fixed hard support. Let a_t^0 and $a_t^\star$ be baseline and target routing distributions on the same support Ω_t.

G.1 QUERY CONTROL

Softmax satisfies

$$\mathrm{softmax}(s_t + \Delta s_t) = a_t^\star$$

if and only if

$$\Delta s_t = \log a_t^\star - \log a_t^0 + \gamma_t \mathbf{1}$$

for some row constant γ_t. Choose a reference key $s_0(t) \in \Omega_t$ and define the gauge-free log-odds target

$$\lambda_{t,s} = \sqrt{d_k} \left(\log \frac{a_{ts}^\star}{a_{t,s_0}^\star} - \log \frac{a_{ts}^0}{a_{t,s_0}^0} \right). \tag{67}$$

With K fixed, a post-positional query displacement u_t realizes the target routing if and only if

$$u_t (k_s - k_{s_0})^\top = \lambda_{t,s}, \qquad s \in \Omega_t \setminus \{s_0\}. \tag{68}$$

Let K_t^Δ contain the rows $k_s - k_{s_0}$. Then the feasible set is

$$\mathcal{U}_t = \left\{ u : K_t^\Delta u^\top = \lambda_t \right\}.$$

If the positional Q map is $R_{p_t}^Q$, define the de-positioned control

$$\tilde{u}_t = u_t (R_{p_t}^Q)^{-1}.$$

For orthogonal RoPE,

$$\tilde{u}_t = u_t (R_{p_t}^Q)^\top.$$

Stacking all de-positioned controls gives $\widetilde{U}$, and a fixed Q update must satisfy

$$\mathbf{Z} \Delta W_Q = \widetilde{U}.$$

Theorem. For a target routing field $A^\star$,

$$\boxed{r_Q^{\mathrm{ctrl}}(A^\star) = \min_{\widetilde{U}} \operatorname{rank} \widetilde{U},} \tag{69}$$

where the minimum is over fields satisfying

$$u_t \in \mathcal{U}_t, \qquad \tilde{u}_t = u_t (R_{p_t}^Q)^{-1}, \qquad P_{\mathbf{Z}} \widetilde{U} = \widetilde{U}.$$

Proof. For any feasible $\widetilde{U}$, the fixed-map lemma gives a shared Q update if and only if $P_{\mathbf{Z}} \widetilde{U} = \widetilde{U}$, and the minimum update rank equals $\operatorname{rank} \widetilde{U}$. Minimizing over all routing-equivalent feasible fields gives Eq. equation 69. □

G.2 KEY CONTROL

With Q fixed, the required key displacements satisfy

$$q_t (\delta k_s - \delta k_{s_0})^\top = \lambda_{t,s}. \tag{70}$$

Let $\mathcal{Y}_K(\Lambda)$ denote the jointly feasible de-positioned key control matrices satisfying all equations equation 70. A fixed K update additionally requires

$$P_{\mathbf{Z}} Y = Y.$$

Hence

$$\boxed{r_K^{\mathrm{ctrl}} = \min_{Y \in \mathcal{Y}_K(\Lambda),\, P_{\mathbf{Z}} Y = Y} \operatorname{rank} Y.} \tag{71}$$

Unlike Q control, the equations are coupled across query rows because one key participates in every future query row that can attend to it.

H PERSISTENT ANCHOR ABSORPTION

Return to the general decomposition

$$S = P \cup Q, \qquad H_Q^c = (I - M)\widetilde{B}V_S + M\bar{C}.$$

H.1 EXACT ABSORPTION

Assume that for every query row t,

$$\bar{c}_t \in \mathrm{conv}\{v_p : p \in P\}. \tag{72}$$

Then there is a probability row vector r_t such that

$$\bar{c}_t = r_t V_P.$$

Stack the rows into R_P. Let E_P be the coordinate-selection map satisfying

$$E_P V_S = V_P.$$

Define

$$\boxed{A^\star = (I - M)\widetilde{B} + MR_P E_P.} \tag{73}$$

Proposition. If the entries used by R_P are contained in the compiled attention support, then

$$\boxed{A^\star V_S = H_Q^c.} \tag{74}$$

Proof.

$$\begin{aligned} A^\star V_S &= (I - M)\widetilde{B}V_S + MR_P E_P V_S && (75)\\ &= (I - M)\widetilde{B}V_S + MR_P V_P && (76)\\ &= (I - M)\widetilde{B}V_S + M\bar{C} && (77)\\ &= H_Q^c. && (78) \end{aligned}$$

□

Thus no additional value correction is required at this activation site. A fixed parameter realization of $A^\star$ is characterized by Appendix G.

H.2 APPROXIMATE ABSORPTION

Let

$$\hat{c}_t = \arg \min_{v \in \mathrm{conv}(V_P)} \|\bar{c}_t - v\|_2.$$

Replacing $\bar{c}_t$ by $\hat{c}_t$ leaves row error

$$m_t(\bar{c}_t - \hat{c}_t).$$

Hence the exact squared Frobenius error is

$$\boxed{E_{\mathrm{anchor}}^2 = \sum_t m_t^2 \,\mathrm{dist}^2\left(\bar{c}_t, \mathrm{conv}(V_P)\right).} \tag{79}$$

H.3 RETRIEVAL COMPLEXITY

Low-dimensional pre-softmax scores do not imply low-dimensional post-softmax retrieval.

Proposition. There are rank-one score matrices whose row-softmax retrieval matrix has full rank.

Proof. Choose distinct positive x_i and define

$$q_i = \log x_i, \qquad k_j = j - 1.$$

Then

$$S_{ij} = (\log x_i)(j-1)$$

has rank one, while

$$\exp(S_{ij}) = x_i^{j-1}$$

is a Vandermonde matrix and is full rank. Row normalization multiplies each row by a nonzero scalar and therefore preserves rank. □

A direct retrieval complexity is therefore

$$r_{\text{retrieve}} = \operatorname{rank} \operatorname{stack}_q(R_q V_C).$$

If a row-stochastic retrieval matrix R has nonnegative rank k, write

$$R = UV, \qquad U, V \geq 0.$$

Normalize each nonzero row of V. If D contains its row sums, define

$$\widetilde{V} = D^{-1}V, \qquad \widetilde{U} = UD.$$

Then

$$R = \widetilde{U}\widetilde{V}, \qquad \widetilde{V}\mathbf{1} = \mathbf{1}.$$

Since $R\mathbf{1} = \mathbf{1}$,

$$\widetilde{U}\mathbf{1} = \mathbf{1},$$

and

$$RV_C = \widetilde{U}(\widetilde{V}V_C). \tag{80}$$

Thus nonnegative rank k yields a representation by k shared virtual values.

In particular, if

$$s_{tj} = \alpha_t + \beta_j,$$

then

$$R_{tj} = \frac{e^{\beta_j}}{\sum_\ell e^{\beta_\ell}},$$

which is independent of t, and therefore

$$\operatorname{rank} R = 1.$$

H.4 Free virtual anchors

Let

$$\mathcal{C} = \{\bar{c}_t\}$$

be a finite set and

$$d_{\text{aff}} = \dim \operatorname{aff}(\mathcal{C}).$$

Proposition. The minimum number of freely chosen anchors whose convex hull contains $\mathcal{C}$ is

$$\boxed{k_{\min}^{\text{free}} = d_{\text{aff}} + 1.} \tag{81}$$

Proof. The convex hull of k points has affine dimension at most $k-1$, giving $k \geq d_{\text{aff}} + 1$. Conversely, any finite subset of a d_{aff}-dimensional affine space is bounded and is contained in a sufficiently large d_{aff}-simplex, which has $d_{\text{aff}} + 1$ vertices. □

H.5 Selective BOS value writing

Let $B \in P$ be a persistent BOS token with normalized state z_B, and stack all other retained-token states over the query family into $\mathbf{Z}_{\neg B}$. We seek

$$z_B \Delta W_V = d^\top, \qquad \mathbf{Z}_{\neg B} \Delta W_V = 0.$$

Proposition. Such an update exists for arbitrary d if and only if

$$\boxed{z_B \notin \mathrm{row}(\mathbf{Z}_{\neg B}).} \tag{82}$$

When this condition holds, a rank-one solution exists.

Proof. The null space of $\mathbf{Z}_{\neg B}$ is the orthogonal complement of its row space. Hence there exists a with

$$\mathbf{Z}_{\neg B} a = 0, \qquad z_B a \neq 0$$

if and only if Eq. equation 82 holds. Rescale a so that $z_B a = 1$, and set

$$\Delta W_V = a d^\top.$$

Then

$$z_B \Delta W_V = d^\top, \qquad \mathbf{Z}_{\neg B} \Delta W_V = 0,$$

and $\mathrm{rank}\, \Delta W_V \leq 1$. □

H.6 Rank-one anchor construction

Suppose

$$\bar{c}_t = \bar{v}_C \qquad \forall t,$$

the BOS token satisfies Eq. equation 82, the BOS position is available to every relevant query row, and the target anchor routing has a feasible de-positioned Q control $\widetilde{U}_{\mathrm{BOS}}$ satisfying

$$P_{\mathbf{Z}} \widetilde{U}_{\mathrm{BOS}} = \widetilde{U}_{\mathrm{BOS}}, \qquad \mathrm{rank}\, \widetilde{U}_{\mathrm{BOS}} = 1.$$

Then a rank-one V update writes $\bar{v}_C$ into the BOS value and a rank-one Q update realizes the required routing. Together they reproduce the head-level context effect.

I Layerwise Compilation

Let $S = P \cup Q$ be the retained tokens. Assume a causal decoder with token-wise normalization and MLP sublayers, aligned initial retained states, and exact matching of the retained-token attention output at every layer.

Theorem. For every layer ℓ,

$$\boxed{X^{\mathrm{comp}}_{\ell,S} = X^{c}_{\ell,S}.} \tag{83}$$

Proof. At the input layer the retained states agree by assumption. Suppose they agree at the input of layer ℓ. Token-wise normalization therefore produces identical normalized states. By the attention-matching assumption, the attention outputs agree, so the post-attention residual states agree. The MLP receives identical token states and acts token-wise, hence its gate, up-projection, nonlinearity, down-projection, and output agree. The post-MLP residual states therefore agree. Induction proves Eq. equation 83. □

In this construction,

$$\Delta W_{\mathrm{gate}} = \Delta W_{\mathrm{up}} = \Delta W_{\mathrm{down}} = 0.$$

If layer ℓ is compiled by a V control field $\mathbf{D}_{V,\ell}$ with

$$P_{\mathbf{Z}_\ell} \mathbf{D}_{V,\ell} = \mathbf{D}_{V,\ell},$$

then

$$r^{\mathrm{ctrl}}_{\ell,V} = \mathrm{rank}\, \mathbf{D}_{V,\ell}.$$

I.1 RESIDUAL TRANSPORT

For

$$x^+ = x + f(N(x)),$$

the local Jacobian is

$$J = I + Df\, DN.$$

If

$$\|Df\, DN\|_{\mathrm{op}} < 1,$$

then J is invertible by the Neumann criterion. Hence for every activation subspace U,

$$\dim J(U) = \dim U.$$

For RMSNorm,

$$N(x) = D_\gamma \frac{x}{s}, \qquad s^2 = \frac{\|x\|_2^2}{d} + \epsilon,$$

and

$$DN_x = D_\gamma \left[\frac{I}{s} - \frac{xx^\top}{ds^3} \right]. \tag{84}$$

The bracketed matrix acts by $1/s$ on $x^\perp$ and by ϵ/s^3 on x. Therefore DN_x is invertible whenever

$$\epsilon > 0 \qquad \text{and} \qquad \gamma_i \neq 0 \quad \forall i.$$

J ARCHITECTURE CONDITIONS

The layerwise control construction uses the matched-state convention of Appendix E: $B_\ell^0(q)$ is the retained-only attention operator evaluated at the Skill-conditioned retained state $X_{\ell,S}^c(q)$, with C removed from the attention support and with deployment positions. It is therefore a counterfactual attention operator at a matched state, rather than the attention matrix obtained from an independent free-running query-only trajectory. The equality of matched retained states across successive compiled layers is established by Appendix I.

The exact fixed-map results in Appendix E, the rank identities in Appendix F, and the combined-projection results require only their stated linear equations.

The same-state routing result in Appendix D uses Eq. equation 25 and translation-equivariant hard support. Standard fixed-frequency RoPE satisfies the required score identity because

$$R_{p+\delta} R_{s+\delta}^\top = R_p R_s^\top.$$

Relative-position biases depending only on $p - s$ satisfy the same identity. A causal sliding-window support is translation equivariant.

The exact Q/K control equations in Appendix G use a fixed hard support, finite logits, an invertible positional map, and an affine pre-softmax score in the controlled Q or K variable. For an architecture with an additional invertible score transform ϕ, the corresponding routing equation is obtained by replacing the target raw-logit difference with the appropriate difference after applying ϕ^{-1}.

The persistent-mass lower bounds in Appendix F use

$$m_t > 0$$

for the exact rank statement, or

$$m_t \geq m_{\min} > 0$$

for the quantitative spectral bound.

The anchor construction in Appendix H uses a compiled attention support that contains every anchor assigned positive target mass. Exact realization by finite-logit softmax additionally uses a target routing with the same positive support as the corresponding unmasked compiled row.

K CAUSAL-PREFIX INVARIANCE OF STATES AND CONTROLS

Fix a layer ℓ, a head, the positional convention, and the attention support. Let p and p' be two calibration inputs built from the same persistent prefix P and the same Skill block C, and let t and t' be retained positions in p and p' respectively. Say that t and t' have the *same causal prefix* when the token sequences up to and including those positions coincide, the positional indices assigned to corresponding tokens coincide, and the attention supports of corresponding positions coincide.

Two remarks fix the scope of this section. First, at the entrance block the states are those of the deployed execution itself, so the two positions occupy their own positions in their own inputs and the last two clauses hold automatically; they are conditions only for the matched-state construction of Appendix E, which governs the blocks above the entrance. Second, every row stacked into one system is required to carry the same persistent prefix P and the same Skill block C. Under that requirement the teacher prefix $P \oplus C \oplus q_{\le t}$ and the deployed prefix $q_{\le t}$ induce the same partition of retained positions, so the equivalence classes below, and their number, do not depend on which of the two executions supplies the states. Rows drawn from different Skills violate this requirement: the state matrix is then constant on a class where the control is not, and the corollary of Section K fails.

K.1 STATES

Proposition. If t and t' have the same causal prefix, then at every layer ℓ

$$X_{\ell,t}(p) = X_{\ell,t'}(p'), \qquad \text{hence} \qquad Z_{\ell,t}(p) = Z_{\ell,t'}(p'). \tag{85}$$

Proof. Induct on ℓ. At $\ell = 0$ the state at a position is a function of that position's token id and positional index alone, which agree by hypothesis. Assume Eq. equation 85 holds at layer ℓ for every pair of positions with the same causal prefix. Each position in the attention support of t precedes t, so its causal prefix is an initial segment of t's; by hypothesis the supports of t and t' correspond, and corresponding support positions have the same causal prefix as each other. The inductive hypothesis therefore gives equality of their layer-ℓ states, hence of their queries, keys and values, hence of the scores on the support. Row-wise softmax over corresponding supports gives equal attention rows, so the attention outputs at t and t' agree. The residual update, the token-wise normalization and the token-wise MLP act on the position's own state alone and therefore preserve the equality, giving it at layer $\ell + 1$. □

This extends Eq. equation 22, which is the special case in which the two positions lie in $P \cup C$ and the two inputs share that whole block.

K.2 CONTROLS

At the entrance block the control is formed from the deployed execution's own quantities: A^0 is the causal attention of that execution, $Z = N(X^0)$ its normalized retained states, and the control is the difference between the retained-token attention outputs of the two executions. The matched-state convention of Appendix E applies to the blocks above the entrance, where the preceding retained states have already been matched.

Proposition. If t and t' have the same causal prefix, then the corresponding rows of the value control agree,

$$\big[D_V(p)\big]_t = \big[D_V(p')\big]_{t'}. \tag{86}$$

Proof. Write $S(t)$ for the retained positions up to and including t, ordered causally, and likewise $S(t')$. The hypothesis puts these two lists in correspondence, and each corresponding pair has the same causal prefix, so Eq. equation 85 gives equality of their states.

Three consequences follow. The retained-token targets agree, $[H^c_S(p)]_s = [H^c_S(p')]_{s'}$ for corresponding s, s', since each is the attention output at a position whose support lies in its own causal prefix. The baseline values agree, $[V_S(p)]_s = [V_S(p')]_{s'}$, since $V_S = Z_S W_V$ is token-wise. And the entries of B^0 on these positions agree, $[B^0(p)]_{su} = [B^0(p')]_{s'u'}$ for corresponding pairs, since B^0 is recomputed from the same states under corresponding supports and positions.

At the entrance block each control row is a forward function of the positions in $S(t)$: the attention output at t is $\sum_{u \le t} A^0_{tu} Z_u W_V$ in the deployed execution and the corresponding sum over its own support in the Skill-conditioned one, and both supports lie inside the respective causal prefixes. Every quantity appearing in either sum has just been shown to agree, so the two control rows agree, which is Eq. equation 86.

Above the entrance block the same conclusion holds for the matched-state control of Appendix E, where $V^\star$ is defined by an inverse rather than a sum. There B^0 is lower triangular with positive diagonal (Eq. equation 29), so forward substitution expresses $V^\star_s$ in terms of entries indexed by $S(s)$ alone, and induction along the causal order gives $V^\star_s = V^\star_{s'}$ at every corresponding pair. □

K.3 Consequences for the stacked system

Let $\mathcal{E}$ be the partition of the retained positions of the calibration set into causal-prefix classes, and $N_{\mathrm{pre}} = |\mathcal{E}|$. Let

$$\mathcal{U}_{\mathrm{pre}} = \{\, v \in \mathbb{R}^N : v \text{ is constant on each class of } \mathcal{E} \,\}, \qquad \dim \mathcal{U}_{\mathrm{pre}} = N_{\mathrm{pre}},$$

where N is the number of stacked rows.

Corollary. Every column of $\mathbf{Z}$ and every column of $\mathbf{D}_V$ lies in $\mathcal{U}_{\mathrm{pre}}$. Consequently

$$\mathrm{rank}(\mathbf{Z}) \le \min\big(N_{\mathrm{pre}},\, d\big), \tag{87}$$

and if $\mathrm{rank}(\mathbf{Z}) = N_{\mathrm{pre}}$ then $\mathrm{col}(\mathbf{Z}) = \mathcal{U}_{\mathrm{pre}}$, so $\mathrm{col}(\mathbf{D}_V) \subseteq \mathrm{col}(\mathbf{Z})$ and

$$P_{\mathbf{Z}} \mathbf{D}_V = \mathbf{D}_V, \tag{88}$$

that is, the calibration system is exactly solvable.

Proof. Membership in $\mathcal{U}_{\mathrm{pre}}$ is Eqs. equation 85 and equation 86 read column-wise. Eq. equation 87 follows since $\mathrm{col}(\mathbf{Z})$ is a subspace of $\mathcal{U}_{\mathrm{pre}}$ and $\mathbf{Z}$ has d columns. When $\mathrm{rank}(\mathbf{Z}) = N_{\mathrm{pre}}$ the nested subspaces $\mathrm{col}(\mathbf{Z}) \subseteq \mathcal{U}_{\mathrm{pre}}$ have equal dimension and therefore coincide, and $\mathrm{col}(\mathbf{D}_V) \subseteq \mathcal{U}_{\mathrm{pre}}$ gives Eq. equation 88 by the fixed-map lemma. □

The converse fails: when $\mathrm{rank}(\mathbf{Z}) < N_{\mathrm{pre}}$, as happens where the rank deficiency comes from a low-dimensional embedding span rather than from repeated prefixes, $\mathrm{col}(\mathbf{Z})$ is a proper subspace of $\mathcal{U}_{\mathrm{pre}}$ and Eq. equation 88 need not hold.

L The Matched-State Entrance, and Where It Is Exact

Section 2.2 defines B^0 by holding the retained states Z_S from the Skill-conditioned execution and recomputing only this head's attention routing over the deployment positions. This is a counterfactual, not the deployed forward pass, and the distinction is measurable.

At the entrance layer the two coincide by construction: the route input is the token embedding, and under the fixed deployment positional convention the route inputs of corresponding retained tokens are the same token embeddings, so the matched state equals the actual state. At deeper layers they do not, because the residual stream reaching a deployed layer has itself been produced without the Skill. Scoring one fixed update at the matched entrance rather than at the actual deployed entrance therefore under-reports the error it will incur:

layer	matched entrance	actual deployed entrance	under-report
0	0.286	0.286	$\mathbf{1.0}\times$
6	1.219	7.157	$5.9\times$
13	1.017	5.574	$5.5\times$
19	0.690	1.565	$2.3\times$
25	0.325	1.013	$3.1\times$

Held-out relative residual $\|\mathbf{D} - \mathbf{X}\Delta W\|_F / \|\mathbf{D}\|_F$ for the same update, judged at the two entrances. The layer-0 row is the internal control that makes the others readable: where the two entrances provably coincide, the two columns agree to three decimals.

The control object of Section 2.2 is exact at the entrance layer and approximate at deeper layers. Appendix I extends the construction to successive layers by matching each entrance after correcting preceding layers; the main experiments evaluate single-site updates.

M Experimental Protocol

Models. Primary model Gemma-2-2B. The structural diagnostics are reproduced on Gemma-2-2B, Gemma-3-12B, Gemma-3-27B, Qwen3-8B, Llama-3.1-8B and Llama-3.1-70B: three families, 2B to 70B, all dense, GQA group sizes $2/4/8$. Qwen3-30B-A3B is measured separately as a sparse control and is not part of the ladder. Within the ladder the three Gemma models hold group size fixed at 2 while d grows $2304 \to 5376$ and depth $26 \to 62$, giving a pure scale axis; Meta-Llama-3-8B and Llama-3.1-8B are geometrically identical and differ only in pretraining, giving a weight axis. Layers follow the frozen rule $\{0, L/4, L/2, 3L/4, L-1\}$; head-to-KV mapping is measured on every model rather than assumed, reconstructing the captured projection input to relative error $3.4\mathrm{e}{-7}$ (fp32) and 2.6–$4.3\mathrm{e}{-3}$ (bf16).

Menu and recipe. $n_{\mathrm{cal}} \in \{16, 32\}$, subset $\in \{\text{random}, \text{low-}C, \text{high-}C\}$, layer $\in \{L00, L06, L13, L19, L25\}$, rank $\in \{4, 16, 64\}$ at `v_proj`: 90 configurations. One recipe for every candidate: 120 steps, learning rate 10^{-4}, batch 4, zero initialization, $\mathrm{lora_alpha} = 2r$ so rank is not confounded with effective scale, and targets taken from the deterministic interpreter rather than from teacher completions. Two further configurations outside the menu (`all_linear` over 26 layers and seven projection kinds, ranks 16 and 64) are trained as a fixed external reference and enter no ordering.

Endpoint. Merged fp32 weights, Skill absent from the prompt, free-running greedy decoding, exact match on 128 sealed examples. All 92 configurations were trained on each of 18 Skill seeds with zero failures – 1,656 arms, 4.17 GPU-hours in total, of which the 90-configuration menus account for 3.97 – which is what makes the cost replay exact. Seeds 0–2 were sealed and preregistered before the grid existed (0.691 GPU-h); seeds 3–19 were added afterwards, each with its priority order frozen to disk before its first arm trained. Two seeds of that range (13, 14) were never launched and are absent throughout; no seed that trained is excluded.

N Structural Results in Detail

Identifiability. The rank identity is attained exactly on all 950 non-degenerate cells of the primary model. On the six-model ladder the bound holds on $60/60$ cells with equality on $48/60$; the twelve exceptions are all at layer 0, where the residual stream is close to raw token embeddings and the design matrix is genuinely degenerate. All 950 cells lie on the $N_{\mathrm{pre}} < d$ side; the opposite branch is reachable only on Gemma-2-2B ($N_{\mathrm{pre}} = 2880$ against $d = 2304$), where equality holds on $4/5$ layers. On discriminating cells the prefix law is correct on $8/12$ and the nominal row-count law on $0/12$, the latter falsified on all 24 cell-by-layer comparisons; the two laws' predicted transitions differ by 2, 3 or 6 ladder steps and never by one.

Grouped-query realizability. Median group-joint relative residuals: 0.167 (Gemma-2-2B), 0.205 (Gemma-3-12B), 0.225 (Gemma-3-27B), 0.486 (Qwen3-8B), 0.401 (Llama-3.1-8B), 0.410 (Llama-3.1-70B); largest single cell 0.896. On the primary model the twenty testable cells span 0.226–0.944.

The group-joint residual is the informative statistic on this axis. A single head supplies 44 equations in 64 unknowns, giving an underdetermined system whose measured residual is ≈ 0 regardless of shared realizability; its nonzero residual is sensitive to the pseudo-inverse cutoff. The group-joint system is overdetermined and cutoff-insensitive.

Matched-equation subgroup control, median of cell medians: Llama-3.1-70B $0.125/0.260/0.361$ for group size $2/4/8$; Qwen3-8B $0.241 \to 0.440$ and Llama-3.1-8B $0.188 \to 0.331$ from size 2 to 4. Cross-model the ordering inverts – at matched subgroup size 2 the large-group models (0.241, 0.188, 0.125) sit at or below the group-size-2 Gemma models (0.167, 0.205, 0.225) – so the group-size claim is within-model only. Sparsity grants no exemption: all 48 layers of Qwen3-30B-A3B are sparse but its attention carries no expert routing, and its median 0.499 with matched size-2 value 0.209 falls

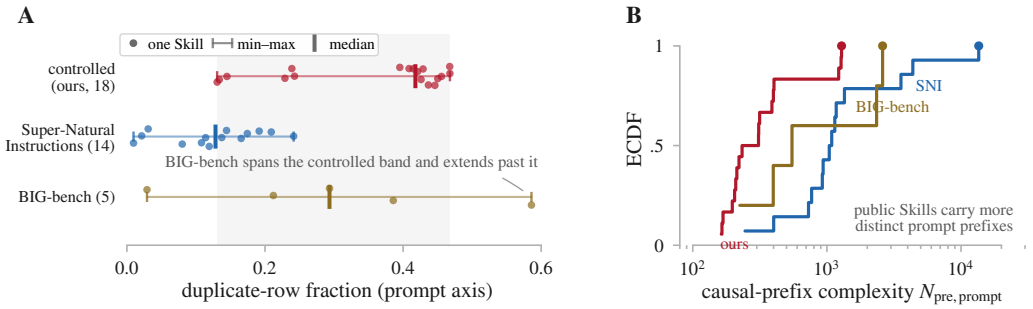


Figure 4: **The geometry Stage I indexes occurs in nineteen publicly authored Skills, and its corpus dependence is visible.** (A) Prompt-axis duplicate-row fraction per Skill. Super-NaturalInstructions sits below our controlled families; BIG-bench spans the controlled band (shaded) and extends past its top, so prompt-prefix duplication is a property of how a corpus was built, not of synthetic versus public. (B) Causal-prefix complexity, ECDF per source; $N_{\text{pre,prompt}}$ counts distinct causal prefixes on the prompt axis. These measurements are structural: no arm was trained on these Skills.

inside the dense band. Weights versus architecture: paired median -0.011 over 80 matched cells, Wilcoxon $p = 0.16$, which also sets the 0.01–0.02 noise floor.

Conditioning. The entrance matrix condition number has layerwise medians spanning roughly seventeen orders of magnitude, about 10^2 at $L25$ and 10^{18} at $L06$. This is why invertibility is stated structurally and every reported quantity is computed in residual form rather than by forming an inverse. Conditioning does not explain the realizability violation (Spearman $\rho = 0.316$, $p = 0.18$).

Capacity. Median observed/predicted ratio 1.0037 over 63 points. Separated: the 21 points at $L00$, independently diagnosed as not capacity-bound, have median 1.0008 and maximum 1.0035; the remaining 42 have median 1.0128 and maximum 1.6335. The pooled figure is pulled toward 1 by the former. The ratio grows with rank and separates at ranks 32 and 64.

The answer-axis zero-collision property holds on all nineteen Skills and all eighteen controlled cells. On the prompt axis Super-NaturalInstructions duplicates in 0.010–0.241 of rows against 0.131–0.468 for our generators, while BIG-bench spans 0.029–0.586, containing the controlled range and exceeding its top. Read from one corpus this looks like public Skills occupying a different structural region; read from both, prompt-prefix duplication tracks how a corpus was built rather than whether it is synthetic. No arm was trained on these Skills.

O Search-Cost Accounting

Table 3: Cost to reach a target fraction of the best configuration in the fully enumerated 90-configuration menu, over 18 Skill seeds. Ratios are paired within seed and aggregated as a geometric mean; "total" charges preflight's 0.0071 GPU-h in full against one seed. $k(\alpha)$ counts qualifying configurations.

Target α	Candidate-training	95% CI	Total cost	Seeds reversed	$k(\alpha)$ median
90%	3.87×	[2.90, 5.19]	1.13×	0/18	10
95%	**4.76×**	**[3.49, 6.46]**	**1.86×**	**0/18**	5.5
99%	6.44×	[4.30, 9.26]	3.23×	1/18	1

Ties are re-drawn within each replay sample, so a policy induces a distribution over orders rather than a single sequence. At the 95% target the p10–p90 cost spread is 1.7×, 1.3× and 4.0× across the three seeds, and the frozen order lies on the p90 side of its own distribution – that is, the reported saving is not a favourable draw.

Preflight overhead is one activation capture plus an SVD, measured at 0.0071 GPU-h per Skill seed. Against enumerating the menu (≈ 0.23 GPU-h) that is 3%; against the search cost it replaces at the

95% target it is 66%, 93% and 184% on the three seeds. Charged in full against a single seed the procedure saves 3.05× and 1.93× on seeds 1 and 2 and costs 0.93× on seed 0.

Out-of-menu reference points: `all_linear` rank 64 scores +0.031 and +0.133 above the best menu configuration on seeds 0 and 1 and −0.039 on seed 2, at $1/40$ the cost of enumerating the menu. Changing only its calibration subset moves it 0.805 (random) / 0.820 (high-C) / 0.445 (low-C); rank 4 on high-C scores 0.820, above rank 64 on random at 0.805 with 16× fewer trainable parameters.

P Which Skills Admit a Compilation-Cost Menu

Table 4: Candidate Skill families and their measured screening outcomes for the end-to-end cost study.

Candidate	Failure mode	Measured trigger
`policy_tool_v1`	—	**admitted**; menu spans 0.555–0.977
`kv_memory_v1`	information	at full coverage the calibration and test pools are the same 62 keys, so disjointness and non-degeneracy are mutually exclusive; at zero overlap every arm scores exactly 0.000 with train fit ≈ 1.0
`nonce_dsl_v1`, d1	information	the test primitive never appears in calibration: train 1.000, test 0.000
`nonce_dsl_v1`, d2	generalization	primitives are identifiable from calibration and capacity is ample, yet all nine probe arms score at the constant-predictor level across a 6000× parameter range, with train fit reaching 1.000
`secret_arithmetic_v1`	degenerate label	answers depend only on $x \bmod 10$; distinct answers over 62 test items are $5/5/2$ and a constant predictor scores 0.581
`nonce_transducer_v1`	absent capab.	teacher 0.005, base 0.000
`select_copy_v2`	ceiling	26 of 36 arms saturate
Eight public Skills	constant resp.	gain over always predicting the majority class is 0.000 / +0.023 / 0.000 / −0.172

The depth-2 row is the most informative, because it fails for a reason the theory does not cover. A constraint-propagation analysis of the calibration set identifies every primitive at the calibration sizes used, and the available capacity exceeds the compiled target by orders of magnitude, yet the compiled model memorizes the calibration chains and answers held-out compositions at chance with perfect training fit. Identifiability of the underlying rules is necessary but not sufficient for a trained adapter to recover compositional generalization: an information-theoretic argument over the calibration set does not predict what gradient descent extracts from it.

The last row generalizes into a reusable screening criterion. On a binary task, a qualifying teacher score together with a gain over the base model is not sufficient evidence that the task is usable, because a constant responder satisfies both while learning nothing. Balanced accuracy, the majority-class prior, and the gain over a constant baseline must be reported together.

Q Implementation Conventions

The predicted control error is a function of (seed, layer, rank) and invariant to the calibration axis, because it is computed on a fixed stratification of 64 modal-length examples rather than on any of the six calibration arms in the menu. Stage I therefore screens the calibration design while the capacity term scores a different one; measured over our menu each (seed, layer, rank) has exactly one value, giving 15 distinct values per seed.

The spectrum is computed per KV group and summed, whereas the rank constraint of Section 2.5 applies to the combined update, with $\max_g \operatorname{rank} D_g \le r_{\text{combined}} \le \sum_g \operatorname{rank} D_g$. Allocating rank r to each group separately permits a combined rank of up to $4r$, so the score is an optimistic lower bound on what a true rank-r adapter attains. The joint problem does not reduce to a single SVD, since each group carries its own design matrix.

R SKILL FAMILIES

Each controlled family is produced by a generator that emits the Skill text, the examples, and a deterministic interpreter. For every family we record three artifacts: the teacher prompt (Skill document, demonstrations, query), the gold answer from the interpreter, and the exact string the student receives at deployment. The third is the whole input at deployment – the Skill document is absent, and no demonstrations remain.

Families and roles. `policy_tool_v1` (procedural/conditional policy) is the admitted end-to-end cost family. `nonce_dsl_v1` (rule transformation) and `secret_arithmetic_v1` (procedural) test rule induction. `nonce_transducer_v1` (rule transformation) is an absent-capability control. `kv_memory_v1` (retrieval memory) has no generalizing rule by construction, and `select_copy_v2` (retrieval) is the saturating control. Table 4 gives the measured exclusion trigger for each.

Split discipline. Splits partition the canonical semantic unit, not the surface string. For `nonce_dsl_v1` the unit is the abstract syntax tree of the primitive chain, so every surface rendering of a chain lies on one side of the split and a student that memorized training strings transfers nothing. For `policy_tool_v1` the calibration and test pools draw disjoint amount ranges under one fixed rule.

Endpoint hygiene. The deployment endpoint is merged fp32 weights, Skill absent, free-running greedy decoding, exact match on sealed test items. We report the constant-predictor rate for every family used, since a task whose answers concentrate on a few values can be scored well by a constant response; families failing that check are excluded in Table 4 rather than reported.

S CONTROL GEOMETRY AND TASK OPTIMIZATION

The recipe used for every deployed configuration fits interpreter gold with a zero-initialized adapter; its loss contains no control term. Since Section 2 analyzes compilation through a virtual-value control, it is worth measuring directly what ordinary task optimization does to that control.

The control residual is 1.0 at zero update by definition. On two well-conditioned layers, a task-only recipe drives it monotonically *up*, to 3.56 and 5.78, and its minimum occurs at step 5 and step 8 respectively – that is, before the task is learned. Standard supervised fine-tuning is anti-aligned with the control object the structural analysis uses.

A composite objective does not damage the task on the better-conditioned of the two layers: the composite arm and the task-only arm both cross the 0.95 task threshold at step 59 and both reach IID exact match 1.0000, while the composite arm's control residual falls to 0.3956 – about four times the preregistered 0.10 threshold, still descending at roughly -0.003 per step at the end of the budget. On the other layer the task-only arm reaches only 0.6641, and under the preregistered rule that layer is not interpretable.

Better conditioning does not produce a lower residual: the image-space lower bound on that layer is $40\times$ smaller than in an earlier experiment while the composite arm's final value is higher, so the binding constraint is not the image-space floor.

These observations distinguish capability recovery from exact reproduction of the Skill-conditioned computation. In the configuration-search study, structural scores prioritize candidates that are trained exclusively on interpreter targets.

T DEPLOYMENT ENDPOINT DIAGNOSTICS

The compiled Skill-free model scores above the in-context teacher on the same sealed set ($0.789/0.844/0.773$ against $0.367/0.469/0.734$). Because this is an unusual direction, we record the mechanism.

Targets come from the deterministic interpreter, not from teacher completions. Teacher competence therefore bounds neither the training signal nor the compiled result; the teacher is a reference operating point for what the Skill achieves in context, not a ceiling.

The diagnostics rule out measurement artifacts as the source of the gap: format correctness is 1.0, argument-name correctness is 1.0, the stop string fires on $128/128$ rows, and four exact-match variants (exact, strict, first-line, and merged-IID) agree row for row. The single observed failure mode is selecting the wrong branch of the policy, which is a task error rather than a decoding or formatting error.

U Behavioral Breadth Across Skills and Models

Section 4.3 reports the ordering effect on a second Skill family and two further architectures. This appendix gives the per-seed figures, the component breakdown and the cost accounting behind those numbers.

A second prospective family. `select_copy_transform` was run under the protocol of Section 4.3: the priority order was hashed to disk (`602efb90c6582bde`) before the first adapter of any seed trained, and every shard verified that hash before its first arm. Ten prospective seeds, a 30-configuration menu (2 calibration sizes $\times$ 5 layers $\times$ 3 ranks).

Table 5: Ordering quality on the second family, 10 prospective seeds. Ratios are paired within seed and aggregated as a geometric mean, matching Table 3.

Ordering	Candidate-training	95% CI	Seeds reversed
Full preflight (Stage I + capacity)	$\mathbf{5.82}\times$	$[4.31,\ 7.97]$	$0/10$
Capacity term alone	$5.05\times$	$[3.59,\ 6.73]$	$0/10$
Stage I + rank heuristic	$2.52\times$	$[1.93,\ 3.39]$	$0/10$
Rank heuristic alone	$1.72\times$	$[1.43,\ 2.05]$	$0/10$
Stage I alone	$0.87\times$	$[0.68,\ 1.15]$	$8/10$

The full preflight beats the rank heuristic on 9 of 10 seeds (sign test $p = 0.0107$). Stage I alone does not order this family: its menu varies calibration size but not calibration construction, so the term that separates subsets on the primary menu has little to act on.

What the fixed cost does here. Preflight's own computation costs 0.0714 GPU-h per Skill seed on this family against 0.0071 on the primary one – the capture is the same operation, but the family's modal prompt is longer and its menu is a third the size. Charged against the blind search it replaces, in the convention of Section 4.3, the overhead-inclusive ratio is $0.259\times$: the ordering is strong and the fixed charge is not repaid at this menu size. A prospective nested-menu study on ten further seeds ($30 \to 60 \to 90$ configurations, targets held fixed) moves that ratio from $0.33\times$ to $0.40\times$ without reaching parity, with 2 of 10 seeds crossing individually at the largest menus. The two families differ in fixed cost by a factor of ten, which dominates the comparison; menu size alone does not account for it.

Two further architectures. The same family, menu and protocol on Qwen3-8B and Llama-3.1-8B, three prospective seeds each. Stage III for both models was captured before any 8B adapter trained.

Table 6: Ordering quality across models, paired geometric means over prospective seeds. Per-seed values are given because n is small.

Model	n	Full preflight	Capacity alone	Rank heuristic	Per-seed (full)
Gemma-2-2B	10	$\mathbf{5.82}\times$	$5.05\times$	$1.72\times$	—
Qwen3-8B	3	$\mathbf{4.87}\times$	$2.82\times$	$2.13\times$	3.57, 3.83, 8.42
Llama-3.1-8B	3	$1.74\times$	$\mathbf{2.88}\times$	$2.61\times$	2.71, 2.38, 0.82

The capacity term is the component that carries across models: it is above parity on all 16 prospective seeds measured, with a worst single seed of $1.47\times$. The composite ordering is ahead on Gemma-2-2B

and Qwen3-8B; on Llama-3.1-8B it aggregates to $1.74\times$, behind both baselines, and one seed reverses to $0.82\times$. Llama's menu is also the least discriminating of the three: 5–8 of its 30 configurations reach the target against 2 on Qwen3, so on two of its three seeds every ordering – including the baselines – finds a qualifying configuration first, and the aggregate turns on the remaining seed. At three seeds per model these are replications of direction, not significance tests: a unanimous $3/3$ cannot reach $p < 0.05$.

Preflight's cost is model-specific and was measured per model rather than carried over: 0.233 GPU-h per Skill on Llama-3.1-8B and 0.351 on Qwen3-8B against 0.0714 on Gemma-2-2B, driven by KV-group count, residual width and how verbosely each tokenizer encodes the Skill. Against menus that are themselves cheaper at 8B, the overhead-inclusive ratios are well below parity on both.

V Parameter-Route and Multilayer Scope

The menu of Section 3 fixes the parameter route to the value projection. This appendix measures what lies outside that choice, under the primary study's recipe with no route-specific tuning: `policy_tool_v1`, Gemma-2-2B, five Skill seeds (0–4), $n_{\text{cal}} = 32$, 120 steps, and the same sealed 128-example endpoint. Every route receives the same five-layer scan $\{0, 6, 13, 19, 25\}$, so best-over-layer is comparable across routes. None of these configurations is in the preflight menu and none is used to rank candidates.

Table 7: Isolated single-site routes at rank 16, five seeds. "Best" and "Median" are exact match over the frozen five-layer scan; Δ is the paired within-seed difference against `v_proj` and "Ahead" the seeds on which the route leads it; "Train loss" is the best final training loss reached.

Route	Best	Median	Δ vs v	Ahead	Train loss
`q_proj`	0.000	0.000	-0.705	$0/5$	2.238
`k_proj`	0.000	0.000	-0.705	$0/5$	2.375
`v_proj`	0.705	0.575	—	—	0.096
`o_proj`	0.722	0.667	$+0.017$	$3/5$	0.044
`gate_proj`	0.763	0.716	$+0.058$	$5/5$	0.017
`down_proj`	0.763	0.705	$+0.058$	$5/5$	0.026
`up_proj`	**0.791**	0.730	$+0.086$	$5/5$	0.018

Four of the six alternatives match or exceed the value route, three of them on every seed; `o_proj` is level with it (3 seeds ahead, 2 exact ties). Between-seed spread is large (sd ≈ 0.10), so the paired column is the informative one, and at five seeds a unanimous margin reaches only $p = 0.0625$: these are consistent directions, not significance tests. Scores are multiples of $1/128$.

The query and key routes behave differently from the rest. Both reach exactly 0.000 on all 50 arms, and the training curves say why: they allocate parameters and reduce the objective by 0.6–1.6 nats, but plateau at a final training loss of 2.00–2.93, whereas all 125 arms on the other five routes end between 0.0007 and 1.232. The two populations do not overlap. Under this single-site rank-16 recipe an isolated query or key update does not fit the calibration set at all – a statement about this recipe and this budget, not about the routes in general, for which Appendix G gives exact control results: the same projections contribute inside the multilayer attention adapter below.

Table 8: Multilayer adapters, same recipe and seeds. `all_linear` rows are the out-of-menu reference of Section 4.3, reused unchanged.

Adapter	Layers	Routes	Trainable params	Held-out exact
Best single-site `v_proj`	1	v	53K	0.705
All-layer `v_proj`, r=16	26	v	1.38M	0.742
All-layer `v_proj`, r=64	26	v	5.54M	0.730
All-layer attention, r=16	26	q, k, v, o	6.39M	0.813
All-layer MLP, r=16	26	gate, up, down	14.4M	0.852
`all_linear`, r=16	26	all seven	20.8M	0.816
`all_linear`, r=64	26	all seven	83.1M	**0.858**

Multilayer compilation is effective, and parameter count buys little at the top of the range: the strongest adapter uses $60\times$ the parameters of the all-layer value adapter for $+0.116$ exact match. Two differences that look like effects are not resolved by five seeds. All-layer `v_proj` is nominally ahead of the best single site ($+0.038$) but wins on only three seeds with one tie and one loss. Raising its rank from 16 to 64 leaves three of five seeds exactly unchanged, so the small negative mean reflects one seed rather than a rank effect.

A deterministic layer-count ladder on the same route and rank (layer sets fixed before training) gives 0.681, 0.713, 0.745, 0.780 and 0.742 for $k = 1, 2, 4, 8, 26$. Read against a between-seed sd of 0.09–0.11 and a largest adjacent gap of 0.038, this says compilation becomes competitive once a handful of layers carry the update; the five-seed design does not resolve an optimal count, and the k=8 entry is one seed above the rest. The k=26 rung is the same configuration as the all-layer rank-16 adapter above.

The preflight of Section 3 ranks candidates within one route at one site. These measurements place that choice: the controlled space is narrower than what compiles, and ranking jointly coupled multilayer configurations without training them would require accounting for how an update at one layer changes the states entering the next.

W SKILL INVENTORY

W.1 BEHAVIORAL SKILL INSTANCES

The main behavioral study uses a conditional tool-use Skill from `policy_tool_v1`. The prospective replication uses a copy-and-transform Skill from `select_copy_transform`. To make these experimental objects concrete, we reproduce below one instance of each, deterministically regenerated from the same frozen generator, configuration, and seed used in the reported experiments.

Primary behavioral Skill. Family: `policy_tool_v1`; seed: 0 (sealed, preregistered before the grid existed); configuration: `n_rules=3`, `n_exceptions=0`; generator: `experiment/E20_calibration_selection/pt_adapter.py` via `experiment/E15_task_families/_families/policy_tool.py`. The `skill_id` `policy_tool_s0` is confirmed against the `calibration_pair_report.json` used in experiment E28.

```
   Task:  read the request and output only the resulting
tool call, in the form name(tier="...", region="...") and
nothing else.
Manual:
Rules (checked in order; the FIRST one that applies wins):
1.  If region is "keld" and amount is at least 500, call
jimeg.
2.  If tier is "plus" and amount is at least 250, call
dubil.
3.  If tier is "elite" and region is "keld", call zodes.
Default:  if no rule applies, call juzen.

request:  tier=elite, region=keld, amount=958, flagged=no
call:  jimeg(tier="elite", region="keld")

request:  tier=plus, region=sorim, amount=711, flagged=no
call:  dubil(tier="plus", region="sorim")

request:  tier=basic, region=sorim, amount=196, flagged=no
call:  juzen(tier="basic", region="sorim")
```

The query format above ("`request: .../ call:`") is the student rendering; the Skill text precedes it in the model's prompt during both in-context evaluation and calibration training.

Prospective replication Skill. Family: `select_copy_transform`; seed: 3 (first prospective seed in E37); configuration: `n_fields=8`, `n_select=3`; generator:

`experiment/E37_family_campaign/select_copy_hard.py`. Deterministically regenerated from the frozen generator and seed; no cached text artifact exists for this family (the rendered prompt was never serialized to disk).

```
   Task: from each record, output a JSON object containing
only the fields denu, loge, daba, in that exact order, with
no spaces. Apply one transform per field: denu -> drops
the last digit; loge -> doubles the value; daba -> negates
the value.

record: {"denu":776,"demu":834,"loge":339,"fivu":973,
"bipu":259,"paso":756,"daba":209,"zuba":170}
output: {"denu":77,"loge":678,"daba":-209}

record: {"denu":580,"demu":830,"loge":556,"fivu":813,
"bipu":983,"paso":565,"daba":295,"zuba":808}
output: {"denu":58,"loge":1112,"daba":-295}

record: {"denu":347,"demu":615,"loge":736,"fivu":755,
"bipu":337,"paso":935,"daba":873,"zuba":749}
output: {"denu":34,"loge":1472,"daba":-873}
```

W.2 Controlled Skill families

The seven controlled Skill families used in the paper are listed in Table 9. Two are admitted to the behavioral results above; the remaining five are used to probe structural boundary regimes.

Table 9: Controlled Skill families. Admission criteria are given in Table 4 (Appendix P).

Skill family	Capability	Beh	Role in experiments
`policy_tool_v1`	Procedural policy	Y	Primary behavioral study: exhaustive menu enumeration, deployment endpoint, configuration search
`select_copy_transform`	Rule transform. + retrieval	Y	Independent prospective search replication
`nonce_dsl_v1` (d1/d2)	Rule transformation	N	Structural boundary probe
`secret_arithmetic_v1`	Procedural arithmetic	N	Structural boundary probe
`nonce_transducer_v1`	Rule transformation	N	Structural boundary probe
`kv_memory_v1`	Retrieval memory	N	Structural boundary probe
`select_copy_v2`	Retrieval	N	Structural boundary probe (saturation control)

W.3 Public-Skill census

The structural census additionally uses fourteen tasks from Super-NaturalInstructions and five from BIG-bench. Table 10 records their exact task identities.

SNI = Super-NaturalInstructions (allenai, Apache-2.0); BB = BIG-bench (google/BIG-bench, Apache-2.0). The public tasks are used for the structural census, while the behavioral compilation experiments use the controlled Skill instances documented above.

Table 10: Public Skills used for the structural census.

#	Source	Task / Skill ID
1	SNI	task001_quoref_question_generation
2	SNI	task020_mctaco_span_based_question
3	SNI	task034_winogrande_question_modification_object
4	SNI	task039_qasc_find_overlapping_words
5	SNI	task050_multirc_answerability
6	SNI	task102_commongen_sentence_generation
7	SNI	task121_zest_text_modification
8	SNI	task159_check_frequency_of_words_in_sentence_pair
9	SNI	task190_snli_classification
10	SNI	task249_enhanced_wsc_pronoun_disambiguation
11	SNI	task363_sst2_polarity_classification
12	SNI	task380_boolq_yes_no_question
13	SNI	task587_amazonfood_polarity_correction_classification
14	SNI	task670_ambigqa_question_generation
15	BB	crass_ai
16	BB	emoji_movie
17	BB	formal_fallacies_syllogisms_negation
18	BB	logical_args
19	BB	odd_one_out