COMPILE WHAT COUNTS: HOW QUESTION SUPERVISION SHAPES QUERYABLE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

A per-document key–value (KV) code is memory for a frozen reader: optimized once against the document, it is queried in place of the document thereafter. Optimized to store the document, however, such a code need not expose it through the reader's question-answering interface: storage and access are different objectives. What makes the memory queryable is the supervision that bridges them, question–answer pairs that current practice samples from a language model and that guarantee neither per-fact coverage nor question-distribution structure. Question-Grammar Compilation (QGC) instead compiles the supervision: a closed question grammar with three phrasing axes and 27 cells per relation is expanded at exact per-fact exposure, which turns the question distribution into something we can intervene on. A full factorial over axis widths and a path that moves only third-order joint mass at fixed low-order marginals show, on two synthetic platforms, that widening each axis causes gains of up to $+0.22$ accuracy with subadditive interactions, while third-order structure and cell-level exactness show no detected response. At matched budget, unconstrained self-study-style generation loses 0.24–0.40 against the compiled grammar, whereas LLM paraphrases compiled per fact match or exceed it. Axis directions replicate on ClinicalTrials.gov under a prospective gate, where compiled codes reach 0.910 accuracy against 0.869 for the zero-shot full document at $36\times$ fewer attention slots than the document. The dominant factor is systematic, fact-grounded supervision with adequate question support; who writes the questions matters only where the grammar's coverage of interrogative forms is thin.

1 INTRODUCTION

Replacing a long document with a small, per-document key–value (KV) code amortizes long-context reuse: optimize 64 KV slots against a frozen reader once, then answer any number of questions about its facts against the code instead of the document. The code is a memory for the reader, and the property that matters is whether it is queryable. The obstacle is not storage. A code can be optimized to hold a document's content and still fail to surface that content when the reader is asked about it, because storage objectives and interrogative access are different targets. What closes the gap in practice is question–answer supervision: the Cartridges line trains per-document codes on QA pairs that a language model generates about the document (Eyuboglu et al., 2026); this "self-study" recipe is what makes such codes answerable.

Sampling supervision from a generator, however, has two shortcomings. There is no coverage guarantee: a fact the generator never asks about is a fact the code never learns to answer, and the line's own follow-up analysis documents systematic coverage bias in self-generated curricula (Alimaskina et al., 2026). More fundamentally, the field does not know which structural properties of the question distribution the code optimization actually consumes. Without that knowledge, supervision design is guesswork: we cannot say what a generator must be steered toward, or what a cheaper process would have to preserve.

We answer the structural question, and the answer motivates a different way of producing supervision: treat the question distribution as a designed object. Question-Grammar Compilation (QGC) extracts a span-grounded fact graph from the document, closes the space of questions into a finite grammar with three axes, query-frame F, lexical L, and word-order O, and deterministically ex-

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.

pands its $3\times3\times3 = 27$ cells per relation into a supervision stream in which every admitted fact receives exactly 108 exposures under a designed joint distribution over cells. Because the compiler fixes the distribution, we can intervene on it directly. We run two preregistered interventions against a frozen 7B reader: a full 2^3 factorial that widens each axis from one to three variants, and a λ-path that moves only third-order joint mass while holding every single-axis and pairwise cell-exposure marginal exactly fixed, with per-fact token lengths matched throughout.

Widening any phrasing axis raises accuracy substantially, by up to 0.22, but the axes overlap: their pairwise interactions are uniformly negative, and neither the three-way interaction nor a reweighting that moves only third-order structure at fixed low-order marginals shows a detected response. Varying instead where the supervision comes from, at the same budget, sampling cells at random within the grammar changes nothing detectable, unconstrained self-study-style generation falls 0.24–0.40 behind the compiled grammar, and LLM paraphrases compiled per fact match or exceed it. The axis directions carry over to real documents from ClinicalTrials.gov, where compiled codes reach 0.910 accuracy against 0.869 for the reader given the full document, at a median $36\times$ fewer attention slots than the document itself.

Taken together, the results say what a supervision compiler has to guarantee and what it can leave alone. It has to reach every fact and give each one enough distinct ways of being asked; the exact allocation of exposure across cells can be left to chance without detected loss, and moving third-order structure at fixed low-order marginals changes nothing detectable either. The source of the phrasings matters only where the hand grammar's interrogative support falls short, and the per-relation breakdown shows exactly where that is. Our contributions:

- QGC, a compiler that expands a span-grounded fact graph over a closed question grammar at exact per-fact exposure. Because the distribution is constructed, it admits factorial and marginal-fixed interventions and accepts any phrasing source under the same coverage control.
- An interventional account of which properties of question supervision the code consumes: axis widening yields large, subadditive gains; third-order reweighting and cell-level exactness show no detected response; at matched budget, unconstrained self-study-style generation loses 0.24–0.40, while LLM paraphrases inside the compiler match or exceed the hand grammar.
- The matched-budget findings hold under an independently generated question distribution, and the axis directions hold under an exploratory cross-family reader and a gated ClinicalTrials.gov deployment, where all three replicate and compiled codes reach 0.910 against 0.869 for the zero-shot full document at a median $36\times$ fewer attention slots than the document, with residual errors traced to the code rather than extraction.

2 Related Work

Supervision distributions for per-document codes. The closest line trains per-document KV representations on QA pairs that a language model generates about the document: Cartridges introduced the self-study recipe (Eyuboglu et al., 2026), and a follow-up analysis of self-generated curricula documents systematic coverage bias in what such supervision asks about (Alimaskina et al., 2026). This line established that QA supervision is what makes codes answerable; it treats the question distribution as a sampler's output. We instead construct it: QGC replaces sampled supervision with supervision compiled from a closed grammar at fixed per-fact exposure, and then asks which structure of that distribution carries the value.

Learned and amortized context compression. Gist tokens (Mu et al., 2023), the in-context autoencoder (Ge et al., 2024), and AutoCompressors (Chevalier et al., 2023) train compressors that map contexts to soft tokens or summary vectors, amortized across inputs; Still compacts KV caches amortized in a single forward pass (O'Neill et al., 2026); and prompt-pruning methods compress the token sequence itself (Jiang et al., 2023). These systems optimize storage or generic reuse of context. Retrieval-augmented generation (Lewis et al., 2020) retrieves textual context per query; a per-document code fixes the attention state once, and our question is the supervision distribution for optimizing that code. Our object differs on both sides of the optimization: the code is optimized per

document rather than amortized, and the supervision distribution, not the compressor architecture, is the experimental variable.

Inference-time KV selection. H_2O and SnapKV reduce the KV cache by selecting or evicting entries at inference time (Zhang et al., 2023; Li et al., 2024). They control which state is retained from a full forward pass; they do not construct or supervise a code, so the question of supervision structure does not arise there.

Phrasing diversity and knowledge storage. For knowledge stored in model weights, Allen-Zhu & Li (2024) showed that paraphrase-diverse pretraining determines whether facts become extractable, and later work injects paraphrase variation at finetuning time (Kang et al., 2024). This line established that phrasing diversity matters; it operates at the weight level and does not decompose axis interactions under fixed exposure. We work at the frozen-reader, per-instance code layer, freeze the exposure budget, and estimate the full interaction structure of a factored phrasing space: main effects, pairwise, and third-order.

Synthetic question generation and supervision design. A separate line generates QA supervision automatically: roundtrip-consistent synthetic QA corpora (Alberti et al., 2019) and, at scale, model-generated instruction data filtered post hoc for invalid or near-duplicate generations (Wang et al., 2023). That literature controls supervision by filtering samples after generation; QGC instead constructs the distribution so the controlled properties hold exactly, which is what makes the factorial and marginal-fixed interventions of §4 possible at all.

Instance-specific soft state. Prefix-tuning optimizes per-layer key–value prefixes and prompt tuning soft input embeddings, in both cases for a task rather than a document (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022). QGC optimizes per-layer K/V state per document, initialized from the document's own KV, with the training distribution as the object of study rather than a fixed dataset.

3 QGC: Compiling Documents into Queryable KV Codes

Setting. A document d is to be replaced by a code z_d: per-layer key–value tensors occupying 64 slots, injected into a frozen reader θ (Qwen2.5-7B-Instruct; Yang et al., 2024) as its key–value cache at inference. Nothing else is trained: no encoder, no adapter, no reader update. The question is what supervision to optimize z_d against so that the reader can ask d through z_d. Figure 1 summarizes the pipeline and the two interventions built on it.

From document to admitted fact graph. A frozen extractor produces subject–relation–object facts, each carrying a source span; a deterministic verifier rejects any fact whose quoted span does not occur verbatim in the document or whose value fails the field's normalization and type check (Appendix G). The surviving facts form the admitted fact graph $G(d)$. Admission is thus an explicit stage: the guarantees below are stated over $G(d)$, not over every fact latent in the raw document. The relation schema is an input to the compiler: on the synthetic platforms it is the generator's schema, and on ClinicalTrials.gov it is the registry's structured field paths (Appendix G).

A closed question grammar. For each relation we close the space of question surface forms into a grammar with three axes: query-frame F, which fixes the interrogative frame; lexical L, which paraphrases the content words; and word-order O, which permutes the constituents. Each axis has three variants, giving $3{\times}3{\times}3 = 27$ cells per relation. Closing the space is what makes the supervision distribution controllable: any target distribution on the compiler's integer-count lattice (108 exposures per fact) is realized exactly, not approximated by sampling. Concretely, for the sponsor relation the base cell asks "What is the sponsor of trial c?"; changing only F gives "Name the sponsor of trial c.", only L "What is the sponsoring organization of trial c?", and only O "For trial c, what is the sponsor?".

Compilation with exact exposure. Given a target joint distribution, QGC expands $G(d)$ into a supervision stream in which every admitted fact receives exactly 108 question–answer exposures, allocated across cells by integer counts. Three invariants are enforced at compile time (Appendix B):

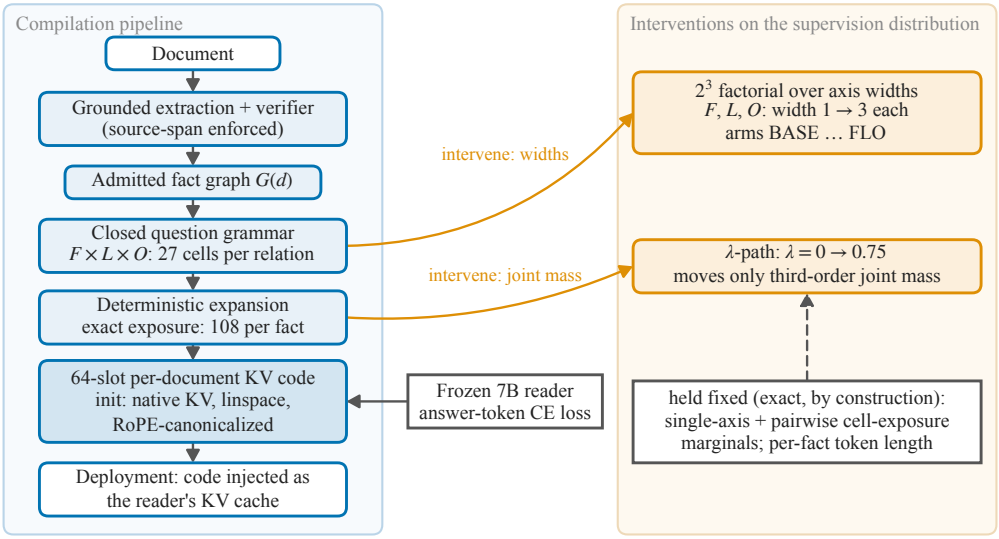


Figure 1: **Question-Grammar Compilation.** Left: a document is compiled into a per-document KV code: span-grounded extraction builds an admitted fact graph; a closed question grammar over query-frame (F), lexical (L), and word-order (O) axes (27 cells per relation) is expanded deterministically at exact integer exposure (108 per fact); the 64-slot code is initialized from the document's native KV and optimized with answer-token cross-entropy against the frozen reader, then injected as the reader's key–value cache. Right: the two interventions, a full 2^3 factorial over axis widths and a λ-path that moves only third-order joint mass while holding all single-axis and pairwise cell-exposure marginals exactly fixed and matching per-fact token length.

schedules are arm-invariant; question token lengths are matched per fact across arms; and cell exposures are exactly uniform over each arm's active support, while the λ family additionally holds every single-axis and pairwise cell-exposure marginal exactly fixed along the path. These invariants are what turn training-data variation into identified intervention: arms differ only in the distributional property under study.

Code optimization. z_d is initialized from the document's own KV state, 64 evenly spaced entries of the native document KV with keys re-rotated to the code's RoPE positions (Su et al., 2024), and optimized in a single pass over the exact exposure multiset with cross-entropy on the answer tokens; question tokens are masked from the loss. Answering through the code uses a median $34\times$ fewer document-side attention slots than the full document, from 23 to $40\times$ across the synthetic corpus and a median $36\times$ on the deployment corpus. The saving is in attention state: the code itself, 3.5 MiB per document in fp16, is not a compressed form of the text.

What compilation guarantees. The compiler fixes the low-order structure of the supervision: per-fact exposure counts and single-axis and pairwise marginals are set as integers, and every fact admitted to $G(d)$ is covered. Whether interventions on these statistics move QA accuracy, and whether the structure the compiler leaves unconstrained has any detectable effect, is the question the next two sections answer.

4 Experimental Design for Mechanism Identification

Why controlled synthetic corpora. Causal identification here requires exact balancing of integer exposure per fact, of cell marginals, and of per-fact token length across arms, which is only possible when the complete fact inventory is known and question surface forms can be instantiated freely. All entities are fictional: no-context accuracy is 0.119 and 0.117 on the two platforms, and in the pooled evaluations it is zero on every compiled relation except phase (0.012), the remainder coming from the yes/no feature items (Table 7). On these platforms the fact graph is emitted by the corpus

Table 1: Effect-coded linear coefficients (95% CIs) for the full 2^3 width factorial over query-frame F, lexical L, and word-order O (widths 1 vs. 3) on both synthetic platforms (supporting: 40 documents; primary: 20). Main effects are positive, all pairwise interactions negative, and the three-way term is not detected on either platform. Coefficient definitions, arm means, logit-scale effects, and permutation p-values: Appendix C.

	Platform 1 (supporting)		Platform 2 (primary)	
Effect	coef.	95% CI	coef.	95% CI
F	$+0.061$	$[+0.047, +0.076]$	$+0.104$	$[+0.088, +0.119]$
L	$+0.106$	$[+0.092, +0.120]$	$+0.168$	$[+0.148, +0.189]$
O	$+0.217$	$[+0.206, +0.228]$	$+0.203$	$[+0.183, +0.223]$
$F\times L$	-0.028	$[-0.040, -0.016]$	-0.022	$[-0.037, -0.007]$
$F\times O$	-0.029	$[-0.040, -0.018]$	-0.031	$[-0.047, -0.014]$
$L\times O$	-0.042	$[-0.053, -0.031]$	-0.038	$[-0.063, -0.012]$
$F\times L\times O$	-0.001	$[-0.012, +0.010]$	-0.006	$[-0.022, +0.010]$

generator itself; extraction (§3) is bypassed, so no uncontrolled extraction stage enters the mechanism experiments. Ground truth is constructive and scoring is programmatic, so arm differences of one to three points are not swamped by annotation noise. External validity is assessed separately on natural documents (§6), where axis directions replicate under a prospective gate. Held-out evaluation phrasings are checked at compile time against the training grammar: none has half or more of its content 4-grams covered by the pool of training questions.

Platforms. Platform 1 contains 40 fictional-biotech documents evaluated over 12 three-seed code arms plus single-seed reference arms; because its first 25 documents were shared with pilot development, it serves as supporting evidence. Platform 2 is an independently generated domain (fictional deep-space operations bulletins; 20 documents), sealed before any evaluation and completed under a protocol frozen before the confirmatory run; it is the primary platform for factorial inference (Appendix A).

Interventions. Two preregistered interventions probe the supervision distribution. (i) A full 2^3 factorial widens each axis from one to three variants, producing arms BASE through FLO; in width-1 arms the active singleton variant is Latin-balanced across relations and seeds, so width is not confounded with template identity. (ii) A λ-path over full-support joint distributions moves only third-order co-occurrence mass (a mod-3 construction; $\lambda \in \{0, 0.5, 0.75\}$, with λ=1 a support-collapsed descriptive endpoint), holding every single-axis and pairwise cell-exposure marginal exactly fixed and matching per-fact token length. Both directions of the λ conclusion were preregistered; the data decide between them.

Estimands and tests. The estimand is document-level mean held-out QA accuracy, averaged first over seeds within a document and then over documents. Factorial effects are effect-coded and tested with 10^5 permutations; $p = 10^{-5}$ is the resolution floor of that count. We report 95% CIs throughout, analyze the linear probability scale as primary, and check signs on the logit scale (Appendix C). Controls include a same-document, same-relation label derangement, a no-context floor, and the zero-shot full document (Appendix B).

5 What the Supervision Distribution Does to the Code

On a common effect-coded accuracy scale, both synthetic corpora show a low-order-dominated response surface: mean main effects of $+0.128$ and $+0.158$, mean pairwise terms of -0.033 and -0.030, and three-way terms of -0.001 and -0.006 that are not detected (Figure 2a). We call this pair of findings low-order concentration: main effects dominate with subadditive pairwise terms, and the one intervention that moves only third-order structure at fixed low-order marginals shows no detected response.

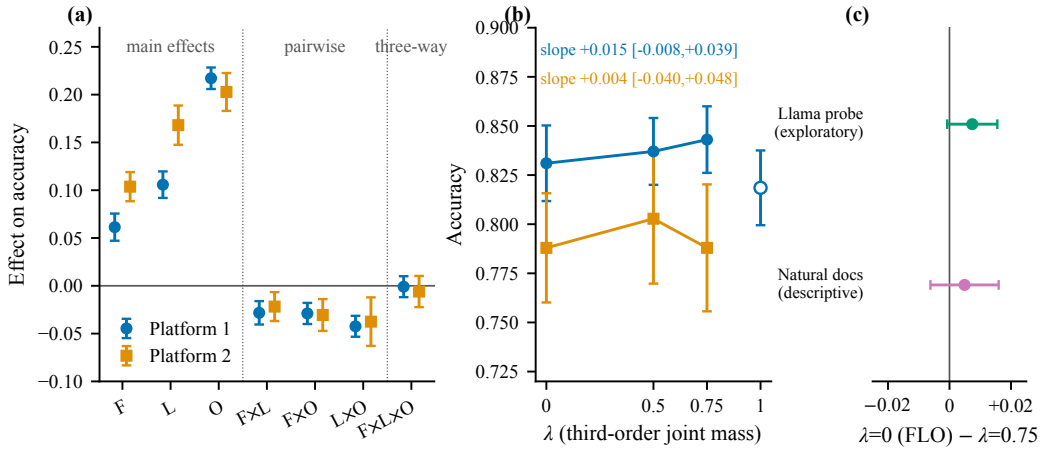


Figure 2: **Low-order concentration.** (a) Effect-coded linear coefficients with 95% CIs for the full 2^3 factorial over query-frame (F), lexical (L), and word-order (O) widths on both synthetic platforms (platform 1, supporting, 40 documents; platform 2, primary, 20 documents): positive main effects, uniformly negative pairwise interactions, three-way term not detected. (b) Accuracy along the λ-path, which moves only third-order joint mass with single-axis and pairwise cell-exposure marginals exactly fixed: no trend is detected on either platform (slopes with 95% CIs inset); the λ=1 endpoint (platform 1, hollow) is support-collapsed and descriptive. (c) Pre-collapse endpoint contrasts, λ=0 (FLO) minus λ=0.75, not detected for the exploratory Llama probe or for natural documents; whiskers are document-level 95% CIs.

5.1 Main effects are large and causal

Widening query-frame F, lexical L, or word-order O from one to three variants raised held-out accuracy on both platforms (Table 1): $+0.061$, $+0.106$, and $+0.217$ for F, L, and O on platform 1 and $+0.104$, $+0.168$, and $+0.203$ on platform 2, each at the permutation floor $p = 10^{-5}$. The point estimates order $O{>}L{>}F$ on both platforms, word order being the axis that moves the relation and entity tokens to different positions within the question (§7).

5.2 Pairwise interactions are uniformly subadditive

On the linear-probability scale, all three effect-coded pairwise interactions were negative on both platforms: $F{\times}L$, $F{\times}O$, and $L{\times}O$ are -0.028, -0.029, and -0.042 on platform 1 and -0.022, -0.031, and -0.038 on platform 2, with $p \leq 2{\times}10^{-5}$ and $p \leq 0.0046$ respectively, indicating less-than-additive average gains from jointly widening axes. On the logit scale all six estimates share the negative sign, and platform-2 $L{\times}O$ is not detected there, at -0.114 $[-0.231, +0.003]$, $p = 0.055$. This pattern is consistent with the axes supplying partially redundant phrasing-invariance signal.

5.3 Third-order structure shows no detected response

We probed third-order structure in two complementary ways, preregistering the estimand, test, and decision threshold for the confirmatory one. Across both corpora, neither the effect-coded $F{\times}L{\times}O$ width interaction, at -0.001 $[-0.012, +0.010]$ and -0.006 $[-0.022, +0.010]$, nor a trend along the full-support mod-3 λ-path, with slopes $+0.015$ $[-0.008, +0.039]$ and $+0.004$ $[-0.040, +0.048]$, was detected (Figure 2b). The λ-path is the sharpest control: with one- and two-axis marginals and per-fact token length fixed, it moves only the structure a low-order compiler leaves unconstrained. The λ=1 endpoint, which collapses support to a Latin square of 9 of 27 cells and is therefore reported descriptively, differs from λ=0 by -0.013 $[-0.029, +0.004]$. The λ-path moves the mod-3 family, one direction in the eight-dimensional space of third-order structure a $3{\times}3{\times}3$ table admits (Agresti, 2002); the remaining directions are untested.

Table 2: Document-level accuracy by supervision source at matched budget (fact graph, reader, code, objective, schedule, and total row budget shared by every code arm; §5.4), on the held-out template evaluation and on the independently generated question distribution of Appendix F. Parentheses: paired difference to the compiled grammar (40 and 20 documents; code arms average three seeds per document, references use one run). Paired 95% CIs and p-values: Appendix F.

	Platform 1		Platform 2	
Supervision source	template	independent	template	independent
Compiled grammar (QGC)	0.831	0.819	0.789	0.775
Grammar, cells sampled	0.838 (+0.007)	0.819 (+0.000)	0.781 (−0.007)	0.778 (+0.003)
LLM paraphrases, per fact	0.867 (+0.036)	0.857 (+0.038)	0.857 (+0.069)	0.849 (+0.074)
Unconstrained self-study	0.588 (−0.243)	0.569 (−0.250)	0.390 (−0.399)	0.375 (−0.400)
Zero-shot full document	0.778 (−0.053)	0.775 (−0.044)	0.728 (−0.061)	0.728 (−0.047)
No context	0.119 (−0.711)	0.124 (−0.695)	0.117 (−0.672)	0.122 (−0.653)

Codes bind facts, not just question formats. In both synthetic platforms, same-document, same-relation label derangement reduced accuracy from $0.831/0.788$ to $0.131/0.138$, within the analysis's 0.05 point-estimate band around the no-context references ($0.119/0.117$). The gains therefore require correct fact-specific bindings; the question formats alone, which the derangement leaves intact, buy nothing.

5.4 Compiled versus generated supervision at matched budget

The interventions above vary structure within the compiler. To test what the compiler contributes relative to generation, we held the fact graph, reader, code size, initialization, objective, optimization schedule, and total row budget ($108\times$facts) fixed and varied only the source of the supervision, whose phrasings, answers, and coverage vary with it (Table 2; Appendix F). Three alternatives bracket the design space: the same grammar with cells sampled uniformly at random rather than balanced exactly; LLM paraphrases written per fact by the reader itself and compiled at the same exposure with fact-graph answers; and unconstrained self-study-style generation, in which the reader reads the document and writes question–answer pairs freely, with no coverage matching and its own answers. Each arm is evaluated on the held-out template phrasings used throughout and on an independently generated question distribution, written by Llama-3.1-8B and screened by a gemma-2-9b-it judge reading the full document; the findings below persist on the items below the 4-gram overlap threshold of §4 (Appendix F). Both evaluations include yes/no feature items outside the compiled relation families (22% of items); every compiled arm sits near the no-context reference on them and only self-study lifts them (Appendix F).

The comparison separates three things. Exact cell balance carries no detected value of its own: sampled cells differ from the compiled grammar by $+0.007$ $[-0.014, +0.029]$ on platform 1 and -0.007 $[-0.035, +0.020]$ on platform 2 under the template evaluation, with paired document-level 95% CIs here and below, so exact balance is what makes the factorial and marginal-fixed interventions possible, not an accuracy lever in itself. Unconstrained generation at the same budget falls short: it loses 0.243 $[0.203, 0.282]$ and 0.399 $[0.355, 0.443]$; the generated pairs touch 91% and 83% of the facts, but exposure across facts is uneven, with Gini coefficients of 0.44 and 0.49, and only 34% and 21% of the pairs have an answer that matches a fact of the entity the question names. Generation placed inside the compiler, with per-fact coverage and fact-graph answers, works: LLM paraphrases exceed the hand grammar by $+0.036$ $[+0.014, +0.058]$ and $+0.069$ $[+0.043, +0.094]$, and the gain is localized. Eight of ten platform-1 and five of ten platform-2 relations differ by less than 0.04; the rest are the unit-bearing numeric relations (duration in weeks and reduction in percent on platform 1, orbit days and downlink in terabytes on platform 2), the two date relations, and the operational-segment relation on platform 2, whose grammar cells carry neither a unit cue nor a when/how-many interrogative form. The residual dependence on phrasing source is thus a gap in the grammar's interrogative support, and the per-relation breakdown of a compiled distribution shows exactly where. Per-fact paraphrase generation costs about as much as code optimization itself; unconstrained generation costs four to five times as much (Table 8).

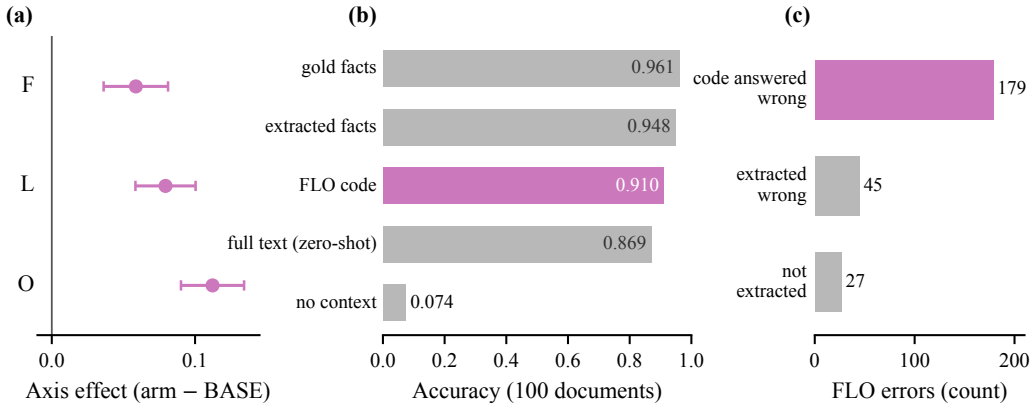


Figure 3: **Compiled codes on natural documents** (ClinicalTrials.gov, protocol frozen in advance). (a) Direction gate: all three axis effects positive (one-sided permutation $p = 10^{-5}$ each, conjunction passed; 60-document gate population); whiskers are document-level 95% CIs. (b) Descriptive accuracy ladder on the 100-document population. (c) Error accounting for the FLO code arm on the same 100 documents: most residual errors are code-side (179 vs. 72 extraction-side, pooled over three seeds).

A second reader family. In an exploratory Llama-3.1-8B (Llama Team, 2024) probe on the platform-2 corpus, all three single-axis contrasts against BASE were positive, at $+0.060$, $+0.099$, and $+0.111$ for F, L, and O, with the same order $O{>}L{>}F$, the direct O versus L contrast not detected ($+0.012$ $[-0.013, +0.037]$), and the FLO minus $\lambda{=}0.75$ contrast not detected ($+0.007$ $[-0.001, +0.016]$): the axis directions hold under a second reader family (Appendix E).

6 Compiled Codes on Natural Documents

The mechanism results motivate one external-validity question: do the axis directions hold on real documents, where extraction is imperfect and phrasing is not synthetic? We answer it on 100 ClinicalTrials.gov registry records (U.S. National Library of Medicine, 2024),[1] with the evaluation population, models, templates, and analysis all frozen before the confirmatory run (Appendix G).

6.1 Prospective Direction Gate

The single gated claim is an intersection–union conjunction: every axis must show a positive direction effect. It passed 3 of 3 on the 60-document gate population: F $+0.059$ $[+0.036, +0.081]$, L $+0.079$ $[+0.058, +0.100]$, O $+0.112$ $[+0.090, +0.134]$, each one-sided $p = 10^{-5}$ (Figure 3a). The point estimates again order $O{>}L{>}F$, for the fourth time across the tested settings. The λ endpoint contrast is again not detected, at $+0.005$ $[-0.006, +0.016]$.

6.2 Deployment Characterization

On the full 100-document population accuracy rises from 0.074 with no context to 0.869 for the zero-shot full document and 0.910 for the compiled FLO code, and to 0.948 and 0.961 when the reader is given the extracted or the gold facts as text (Figure 3b). Evaluation phrasings come from a fixed template set below the 4-gram overlap threshold of §4 against the training grammar, verbalized by a model from a different family that never sees gold values.

Extraction losses are small: mean extraction F1 is 0.974 (per-document accuracy by F1 bin in Appendix G). Accuracy is averaged over the three seeds within a document and then over documents; error counts pool the 914 evaluation items over the three seeds (2,742 item–seed evaluations). Of the 251 FLO errors, 179 are code-side, where the fact was extracted but the code answered wrongly, against 72 extraction-side, where the fact was extracted wrongly (45) or not at all (27; Figure 3c).

[1] Study-level public registry metadata; ten structured fields per record.

Most of the remaining error is therefore in the code itself rather than in the extraction pipeline. The two fact-text reference arms provide the extracted or the gold facts as text at every query, a median of 116 and 112 tokens against 2,320 for the full document. The 64-slot code scores 0.910 against their 0.948 and 0.961, with roughly half their attention state and $36\times$ less than the document.

7 Discussion

Why word order? A positional-geometry hypothesis. Per-fact token-length matching fixes where the answer starts, so the O effect cannot be about answer position. What word-order variation does change is which query tokens occupy which RoPE phases relative to the code slots. A frozen reader's attention is phase-sensitive, so training over multiple orders would teach the code to be readable from multiple phase alignments, consistent with O carrying the largest point estimate in all four settings. Querying existing codes under position perturbation would test this.

What phrasing diversity does to the code. Gold-label derangement collapses accuracy to the no-context band on both platforms even though the question formats are unchanged, so phrasing-diverse supervision is not reducible to a reusable answering skill; what it does is make individually bound facts askable. Together with subadditivity, this points to codes acquiring partially shared invariance to phrasing variation, built on fact-specific bindings.

Design implications. For compiler design the results suggest where to spend effort: guarantee per-fact coverage and low-order support (exactness beyond that showed no detected accuracy benefit, §5.4), spend width budget where marginal returns are highest (pairwise subadditivity), and audit the interrogative support relation by relation, since the hand grammar trailed LLM paraphrases mainly on relations whose cells lacked unit cues and interrogative forms. The support-collapsed Latin square (Hedayat et al., 1999), 9 of 27 cells with all low-order marginals preserved, differed by -0.013 $[-0.029, +0.004]$; the contrast is descriptive, and the design rule it suggests is untested.

Scope. The evaluation is single-hop factual retrieval; the relation schema is an input to the compiler. The mechanism evidence covers the tested axes and width range (1 vs. 3) and the mod-3 third-order family, on two synthetic generators, one confirmatory reader family, and one natural domain. The matched-budget comparison covers the synthetic platforms and one generator family; natural documents and uncovered relation families are its direct next tests.

8 Conclusion

Compiling the supervision of a per-document KV code turns its question distribution into something that can be intervened on, and the interventions give a consistent answer. Every detected effect came from systematic, fact-grounded supervision and question support: axis widening and its pairwise combinations moved accuracy, third-order structure and cell-level exactness showed no detected response, unconstrained generation at the same budget lost 0.24–0.40, and LLM paraphrases inside the compiler matched or exceeded the hand grammar. That is the case for compiling supervision: not that a compiler writes better questions, but that it enforces the properties a queryable memory depends on and exposes, relation by relation, where question support falls short. Compiling what counts, every fact at exact exposure with enough ways of being asked, is what made the memory queryable.

AI use statement

In this work, we used generative AI tools to implement methods, to assist with translation, and to generate synthetic data sets (the LLM-generated supervision pools and evaluation questions studied in the paper; the synthetic corpora themselves are generated programmatically). We did not use generative AI tools to formulate mathematical claims, provide critical ingredients for proofs, or assist with writing proofs; the remaining tasks requiring disclosure were not applicable to this work. Additionally, we used generative AI tools to polish the paper's language and check its grammar. We reviewed all AI-assisted work: AI-assisted code was tested by the authors, reported numbers were checked against the frozen result artifacts, and AI-translated or AI-polished text was revised by the

authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Alan Agresti. *Categorical Data Analysis*. Wiley, 2nd edition, 2002.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6168–6173, 2019.

Ekaterina Alimaskina, Denis Shveykin, Gleb Molodtsov, Igor Shalygin, Alexey Kadeishvili, and Aleksandr Beznosikov. Self-study reconsidered: The hidden fragility of learning from self-generated QA. *arXiv preprint arXiv:2606.32002*, 2026.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 1067–1077, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, 2023.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Ré. Cartridges: Lightweight and general-purpose long context representations via self-study. In *International Conference on Learning Representations*, 2026.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*, 2024.

A. S. Hedayat, N. J. A. Sloane, and John Stufken. *Orthogonal Arrays: Theory and Applications*. Springer, 1999.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023.

Minki Kang, Sung Ju Hwang, Gibbeum Lee, and Jaewoong Cho. Latent paraphrasing: Perturbation on layers improves knowledge injection in language models. In *Advances in Neural Information Processing Systems 37*, 2024.

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*, pp. 3045–3059, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33*, 2020.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems 37*, 2024.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jesse Mu, Xiang Li, and Noah D. Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems 36*, 2023.

Charles O'Neill, Alex Sandomirsky, Harry Partridge, Mudith Jayasekara, and Max Kirkby. Still: Amortized KV cache compaction in a single forward pass. *arXiv preprint arXiv:2606.07878*, 2026.

Belinda Phipson and Gordon K. Smyth. Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

U.S. National Library of Medicine. ClinicalTrials.gov data API, version 2.0, 2024. Study-level registry metadata; accessed August 2026.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems 36*, 2023.

A Preregistration and Confirmatory Claims

This appendix records what was frozen before which run, and which claims are confirmatory.

The λ hypothesis was preregistered with two admissible conclusions. The platform-1 λ-path experiment registered, before execution, both a detection outcome (third-order joint mass carries value) and a nondetection outcome (no detected effect with reported intervals), with a prespecified negative-trend test at $\alpha = 0.05$. The observed slope was $+0.015$ $[-0.008, +0.039]$ ($p_{\text{neg}} = 0.933$; positive direction also not significant, $p_{\text{pos}} = 0.067$), selecting the nondetection outcome. The factorial three-way term, estimated later on both platforms, agrees with the λ-path conclusion; the two estimators are complementary (one moves joint mass under exact marginal fixing, one contrasts width configurations).

Platform 1 is supporting evidence. Its first 25 documents were shared with pilot development, so all platform-1 results are labeled supporting. Nothing was changed after platform-1 outcomes were seen except through the preregistered extension described next.

Platform 2 was completed in two stages. The sealed second-domain corpus was first evaluated on six focal arms under a preregistered replication check, which passed. The remaining factorial arms were added under a preregistered extension frozen before the confirmatory run. Because the initial arms had been inspected beforehand, we treat the confirmatory main-effect gate (passed at $p = 10^{-5}$ per axis) as primary and report pairwise interactions as preregistered estimation targets with post-hoc Bonferroni.

Deployment was frozen in two steps. The corpus and evaluation population were fixed before extraction; models, templates, and analysis were fixed before the confirmatory run. The only gated deployment claim is the three-axis direction conjunction of §6.1; every other deployment number is descriptive.

Label-derangement overlap. Label derangement shares gold values for 1.4% of question pairs; observed floors ($0.131/0.138$ vs. no-context $0.119/0.117$) sit inside the preregistered band, so no conclusion changes.

B Synthetic Platform Construction

Corpora. Platform 1: 40 fictional-biotech documents (median 2,160 reader tokens; 40–70 facts per document, mean 57). Platform 2: 20 fictional deep-space operations bulletins in an independently generated domain with its own relation schema, entity names, style, and date formats, fixed before any evaluation.

Grammar and exposure. Each relation carries a $3\times3\times3$ grammar over query-frame, lexical, and word-order variants. A factorial arm activates widths $(w_F, w_L, w_O) \in \{1, 3\}^3$; exposure is exactly uniform over the active support, totaling 108 exposures per fact in every arm. Per-fact question token lengths are equal across arms line-by-line: a shorter question is padded to the per-fact maximum by appending one or two neutral instructions from a fixed bank ("Reply.", "Answer briefly.", "Give only the short answer."), chosen by measured token count under the reader tokenizer inside the chat template. Padding follows the question, so the answer begins at the same position in every arm; optimization schedules are arm-invariant.

Template inventory. The query-frame axis instantiates three interrogative frames ("What is … ?", "Name …", "State …"), and the word-order axis three constituent orders (standard: "What is the {L} of trial c?"; fronted: "For trial c, what is the {L}?"; topicalized: "Trial c: what is the {L}?"), crossed identically over all relations. The lexical axis supplies three synonymous relation phrases per relation (e.g., for the sponsor relation: "sponsor", "sponsoring organization", "funder"; for the investigator relation: "lead investigator", "principal investigator", "lead researcher"). The full lexical bank and template matrix are released with the compiler; evaluation uses held-out phrasings below the 4-gram overlap threshold of §4 against this training grammar.

Objective. For document d with exposure multiset $E(d) = \{(q_i, a_i)\}$ compiled as above, the code solves

$$z_d \;=\; \arg\min_{z} \sum_{(q,a)\in E(d)} \mathrm{CE}_\theta\big(a \,\big|\, z, q\big),$$

where CE_θ is the frozen reader's cross-entropy on answer tokens (plus end-of-sequence) only, with z injected as the reader's key–value cache; question tokens are loss-masked and θ receives no gradient.

Optimization and evaluation configuration.

Code	64 slots per layer, K and V, per document
Initialization	64 evenly spaced entries of the native document KV, keys re-rotated to code positions
Objective	CE on answer tokens + EOS; question tokens masked
Optimizer	Adam, lr 0.01, micro-batch 16, single pass over the exact multiset
Schedule	deterministic; final micro-batch neither dropped nor wrapped; no early stopping
Exposure	108 rows per fact per arm (exact integer allocation)
Seeds	3 per code arm (incl. label derangement); text references single-seed on platform 1
Evaluation	18 held-out questions per document (synthetic); one per eligible field (deployment)
Scoring	programmatic string match against constructed gold answers

Initialization. The code is seeded from the reader's own KV cache for the document: keys and values are taken at 64 positions spaced evenly over the document, and each key is re-rotated from its original RoPE position to the code's position, one of 64 consecutive slots immediately after the chat prefix, by applying the rotation for the position difference; values carry no positional phase and are copied unchanged.

Compile-time invariants. The compiler verifies marginal uniformity (single-axis and pairwise), per-fact cross-arm token-length equality, arm-invariant schedules, validity of the label derangement, and 4-gram isolation of all held-out evaluation phrasings from the training grammar. Gold answers are constructed by the generator and scored programmatically.

Controls. A same-document, same-relation label derangement, a no-context reference, and the zero-shot full document. Codes are stored in fp16 (3.5 MiB per document) with no loss of per-item accuracy relative to fp32.

Table 3: Full arm means with document-level 95% CIs (mean over seeds per document, then over documents). Arm labels: F/L/O mark the widened axes; λ arms move third-order joint mass only.

	Platform 1 (40 docs)		Platform 2 (20 docs)	
Arm	mean	95% CI	mean	95% CI
BASE	0.448	[0.420, 0.475]	0.319	[0.283, 0.356]
F	0.565	[0.540, 0.591]	0.469	[0.425, 0.514]
L	0.623	[0.596, 0.650]	0.541	[0.499, 0.583]
O	0.735	[0.709, 0.761]	0.584	[0.544, 0.624]
FL	0.686	[0.662, 0.710]	0.659	[0.626, 0.693]
FO	0.797	[0.779, 0.814]	0.685	[0.652, 0.718]
LO	0.828	[0.807, 0.849]	0.743	[0.709, 0.776]
FLO	0.831	[0.812, 0.850]	0.788	[0.760, 0.816]
λ=0.5	0.837	[0.820, 0.854]	0.803	[0.770, 0.836]
λ=0.75	0.843	[0.826, 0.860]	0.788	[0.756, 0.820]
λ=1	0.819	[0.799, 0.838]	—	—
Label derangement	0.131	[0.115, 0.147]	0.138	[0.115, 0.161]
Full document	0.778	[0.744, 0.811]	0.731	[0.668, 0.793]
No context	0.119	[0.100, 0.139]	0.117	[0.099, 0.134]

C Statistical Methodology and Full Estimates

Effects are effect-coded over the 2^3 factorial with $s_a \in \{-1, +1\}$ for width 1 versus 3 on axis a: for a term $T \subseteq \{F, L, O\}$, $\beta_T = \frac{1}{4} \sum_{\text{arms}} \left(\prod_{a \in T} s_a \right) \bar{y}_{\text{arm}}$, the mean of the four arms whose sign product is $+1$ minus the mean of the four whose product is -1. For main effects this is the marginal width-3 minus width-1 contrast; for a pairwise term it is half the change in one axis's simple effect between the two levels of the other axis, averaged over the remaining axis; for the three-way term it is half the change in a pairwise interaction between the two levels of the third axis. Tests are permutation-based with 10^5 resamples (document-level sign flips for deployment; $p = 10^{-5}$ is the resolution floor of that resample count (Phipson & Smyth, 2010)), with 95% CIs from the same permutation/bootstrap procedure. The linear probability scale is primary; logit-scale estimates check that subadditivity is not a ceiling artifact. Table 4 reports both scales with p-values; Table 3 reports all arm means with document-level CIs.

Table 4: Effect-coded coefficients on linear and logit scales with two-sided permutation p-values (100,000 permutations; 10^{-5} is the resolution floor). p-values are raw; platform-2 pairwise terms are judged against the Bonferroni threshold $0.05/3$, which all three pass on the linear scale. Platform-2 $L{\times}O$ on the logit scale is not detected.

	Linear			Logit		
Effect	coef.	95% CI	p	coef.	95% CI	p
Platform 1						
F	$+0.061$	$[0.047, 0.076]$	10^{-5}	$+0.280$	$[0.208, 0.352]$	10^{-5}
L	$+0.106$	$[0.092, 0.120]$	10^{-5}	$+0.526$	$[0.450, 0.602]$	10^{-5}
O	$+0.217$	$[0.206, 0.228]$	10^{-5}	$+1.098$	$[1.037, 1.158]$	10^{-5}
$F{\times}L$	-0.028	$[-0.040, -0.016]$	$2{\times}10^{-5}$	-0.131	$[-0.195, -0.066]$	0.0002
$F{\times}O$	-0.029	$[-0.040, -0.018]$	$2{\times}10^{-5}$	-0.108	$[-0.167, -0.049]$	0.0007
$L{\times}O$	-0.042	$[-0.053, -0.031]$	10^{-5}	-0.112	$[-0.174, -0.051]$	0.0006
$F{\times}L{\times}O$	-0.001	$[-0.012, 0.010]$	0.8683	-0.029	$[-0.091, 0.032]$	0.3383
Platform 2						
F	$+0.104$	$[0.088, 0.119]$	10^{-5}	$+0.466$	$[0.396, 0.537]$	10^{-5}
L	$+0.168$	$[0.148, 0.189]$	10^{-5}	$+0.767$	$[0.663, 0.870]$	10^{-5}
O	$+0.203$	$[0.183, 0.223]$	10^{-5}	$+0.917$	$[0.829, 1.006]$	10^{-5}
$F{\times}L$	-0.022	$[-0.037, -0.007]$	0.0045	-0.085	$[-0.154, -0.017]$	0.0142
$F{\times}O$	-0.031	$[-0.047, -0.014]$	0.0015	-0.115	$[-0.189, -0.041]$	0.0050
$L{\times}O$	-0.038	$[-0.063, -0.012]$	0.0046	-0.114	$[-0.231, 0.003]$	0.0551
$F{\times}L{\times}O$	-0.006	$[-0.022, 0.010]$	0.4559	-0.013	$[-0.086, 0.059]$	0.7080

D THE λ-PATH CONSTRUCTION

The λ-path interpolates, at fixed per-fact exposure 108, between the uniform full-support joint ($\lambda = 0$, the FLO arm: 4 exposures per cell) and a mod-3 concentrated family: the 9 cells of the class $\Pi_c = \{(f, l, o) : f{+}l{+}o \equiv c \pmod 3\}$ receive per-cell exposure w_{in} and the remaining 18 cells w_{out}, with $(w_{\text{in}}, w_{\text{out}}) = (4, 4)/(8, 2)/(10, 1)/(12, 0)$ at $\lambda = 0/0.5/0.75/1$; the class index c rotates across relations and seeds (counterbalancing cell identity), and support collapses to a Latin square of 9 cells exactly at $\lambda = 1$. By construction every distribution on the path has identical single-axis and pairwise cell-exposure marginals (integer-exact), and per-fact token lengths are matched; only the third-order co-occurrence structure moves. For $\lambda \in \{0, 0.5, 0.75\}$ the support is full, so the slope estimand is a pure third-order intervention; $\lambda = 1$ additionally collapses support to an orthogonal-array (Latin-square) design (Hedayat et al., 1999), which is why its contrast is reported descriptively. The mod-3 family is one direction in the eight-dimensional third-order structure of a $3{\times}3{\times}3$ table (Agresti, 2002); probing the remaining directions under the same marginal-fixing is future work.

E CROSS-FAMILY READER PROBE

The probe re-optimizes codes for the platform-2 corpus against Llama-3.1-8B-Instruct (with that reader's own RoPE frequencies and chat template), preregistered as exploratory: one reader on one corpus supports directional replication, not effect-size comparison. Table 5 reports all contrasts.

Table 5: Exploratory cross-family probe (Llama-3.1-8B reader, platform-2 corpus, 20 documents). Single-axis contrasts are positive with the same descriptive point-estimate order; O and L are not distinguishable in a direct paired contrast; the λ endpoint contrast is not detected.

Quantity	estimate	95% CI
$F - \text{BASE}$	$+0.060$	$[0.026, 0.094]$
$L - \text{BASE}$	$+0.099$	$[0.064, 0.134]$
$O - \text{BASE}$	$+0.111$	$[0.085, 0.137]$
$O - L$ (paired)	$+0.012$	$[-0.013, +0.037]$
FLO $-$ λ=0.75	$+0.007$	$[-0.001, +0.016]$
full text	0.831	$[0.780, 0.881]$
no context	0.108	$[0.083, 0.134]$
BASE arm	0.710	$[0.678, 0.743]$

F Matched-Budget Supervision Comparison

Arms. Four compiled arms share the fact graph, frozen reader, 64-slot code, native-KV initialization, answer-token cross-entropy, arm-invariant schedule, and total row budget of $108\times$facts (27 question slots per fact each used four times in the grammar and per-fact paraphrase arms; 108 i.i.d. cell draws per fact in the sampled-cell arm; $27\times$facts document-level pairs each repeated four times in the self-study-style arm). Compiled grammar is the FLO arm of the main experiments, re-run alongside the other arms. Grammar, cells sampled draws each fact's 108 cells i.i.d. uniformly from the 27, so per-fact exposure is exact but cell counts are multinomial. LLM paraphrases, compiled per fact prompts the reader (Qwen2.5-7B-Instruct, temperature 1.0, top-p 0.95) for 40 questions per fact given the relation description, entity identifier, and value (one retry on shortfall), keeps the first 27 after structural filtering (contains the entity identifier, does not contain the answer, 4–40 words, deduplicated), fills a fact still short after the retry by cycling its accepted questions (44 of 29,430 platform-2 slots), and compiles them with the fact-graph answer under the same schedule and cell permutation as the grammar. Self-study-style generation, unconstrained gives the reader the full document and one of eight rotating instructions (specific facts, numbers, dates, people and organizations, places, easily overlooked details, yes/no statements, a random passage), requests 25 short-answer pairs per call (temperature 1.0, top-p 0.95), deduplicates by question, and truncates the pool to exactly $27\times$facts document-level pairs, each repeated four times; the generated answers are used as they are. Pools are generated once per document and shared across the three seeds, which then vary only the training order, as in the grammar arm; the sampled-cell arm redraws its cells per seed. It follows the self-study recipe under our shared training pipeline rather than reimplementing a published system. Per-fact token lengths are not matched for the LLM arms (the length-matching invariant of Appendix B is waived there); training tokens are reported in Table 8.

Independent evaluation distribution. For each of the 18 evaluation slots per document (same entity, relation or feature, and scoring rule as the template item), Llama-3.1-8B-Instruct was prompted for six candidate questions (temperature 0.9, top-p 0.95) under a prompt fixing the target relation and requiring the entity identifier, no answer leakage, and a single-valued answer; structural filters (identifier present, no answer, 5–40 words, no verbatim grammar question); gemma-2-9b-it then confirmed, without the document, that each candidate asks for the intended relation, and, with the document, answered it; the first candidate whose answer was scored correct was kept, falling back to the first intent-checked candidate where none did. Counting an item as overlapping when half or more of its content 4-grams (stopwords removed) occur in the grammar's questions, 8 of 720 and 41 of 360 items overlap the grammar's questions; the subset with no such overlap against any arm's training pool (673 of 720 and 270 of 360 items) gives platform 1 $0.816/0.814/0.850/0.572$ and platform 2 $0.743/0.739/0.801/0.383$ for grammar, sampled cells, paraphrases, and self-study, so the findings persist there, with no detected difference between the grammar and sampled-cell arms. Yes/no feature items lie outside the compiled relation families; every compiled arm sits near the no-context reference on them (Table 7), and the self-study arm, which asks about features freely, is the only supervision that lifts them.

G Deployment Protocol Detail

Population and freeze. 100 ClinicalTrials.gov records; ten structured field paths per record mapped to relation families; the evaluation population was enumerated and fixed before extraction, and the extractor (Qwen3.5-9B; Qwen Team, 2026), verbalizer (Llama-3.1-8B, cross-family, never shown gold values), evaluation templates, and analysis were fixed before the confirmatory run. The direction gate uses the 60-document gate population with BASE/F/L/O arms; ladder arms use all 100 documents.

Evaluation templates. The template set was written at development time, semantically validated, instantiated deterministically at evaluation, and below the 4-gram overlap threshold of §4 against the training grammar. Comparisons therefore concern held-out phrasings within covered relation families.

Transduction. The verifier requires the extractor's quoted span to occur verbatim in the record text after Unicode NFC normalization and the value to pass the field's normalization and type check; a

Table 6: Paired document-level contrasts against the compiled grammar (difference, 95% t-CI over documents, two-sided permutation p with 10^5 resamples; 10^{-5} is the resolution floor), for both evaluations on both platforms. Each document contributes the mean of three seeds for code arms and one run for the references.

Platform, evaluation	Supervision source	diff.	95% CI	p
1, template	Grammar, cells sampled	$+0.007$	$[-0.014, +0.029]$	0.494
	LLM paraphrases, per fact	$+0.036$	$[+0.014, +0.058]$	0.002
	Unconstrained self-study	-0.243	$[-0.282, -0.203]$	10^{-5}
	Zero-shot full document	-0.053	$[-0.092, -0.014]$	0.010
	No context	-0.711	$[-0.734, -0.689]$	10^{-5}
1, independent	Grammar, cells sampled	$+0.000$	$[-0.023, +0.024]$	0.971
	LLM paraphrases, per fact	$+0.038$	$[+0.012, +0.064]$	0.005
	Unconstrained self-study	-0.250	$[-0.290, -0.210]$	10^{-5}
	Zero-shot full document	-0.044	$[-0.089, +0.001]$	0.055
	No context	-0.695	$[-0.724, -0.666]$	10^{-5}
2, template	Grammar, cells sampled	-0.007	$[-0.035, +0.020]$	0.611
	LLM paraphrases, per fact	$+0.069$	$[+0.043, +0.094]$	10^{-5}
	Unconstrained self-study	-0.399	$[-0.443, -0.355]$	10^{-5}
	Zero-shot full document	-0.061	$[-0.129, +0.007]$	0.080
	No context	-0.672	$[-0.706, -0.639]$	10^{-5}
2, independent	Grammar, cells sampled	$+0.003$	$[-0.014, +0.020]$	0.831
	LLM paraphrases, per fact	$+0.074$	$[+0.051, +0.097]$	2×10^{-5}
	Unconstrained self-study	-0.400	$[-0.446, -0.354]$	10^{-5}
	Zero-shot full document	-0.047	$[-0.118, +0.023]$	0.171
	No context	-0.653	$[-0.693, -0.613]$	10^{-5}

Table 7: Per-relation accuracy by supervision source on the synthetic platforms (template and independent evaluations pooled; n = evaluation items per relation; code arms average three seeds per item, references one run). Yes/no feature items lie outside the compiled relation families and are listed for completeness. Zero-shot full document and no context are references.

Relation	n	Grammar	Sampled	Paraphrases	Self-study	Full doc.	No ctx.
Platform 1							
compound	74	0.955	0.955	0.968	0.586	0.973	0.000
phase	82	0.972	0.992	0.959	0.780	0.951	0.012
sponsor	240	0.950	0.954	0.950	0.536	0.729	0.000
city	86	0.950	0.969	0.953	0.496	0.733	0.000
investigator	78	0.966	0.966	0.962	0.538	0.846	0.000
participants	102	0.941	0.948	0.967	0.516	0.725	0.000
reduction (%)	124	0.901	0.879	0.957	0.530	0.790	0.000
duration (weeks)	94	0.507	0.539	0.936	0.443	0.755	0.000
start date	120	0.936	0.942	0.972	0.539	0.708	0.000
completion date	120	0.919	0.903	0.950	0.508	0.708	0.000
yes/no feature (outside grammar)	320	0.545	0.550	0.530	0.708	0.784	0.544
Platform 2							
vehicle	40	0.967	0.967	1.000	0.158	1.000	0.000
operator	120	0.933	0.950	0.942	0.175	0.642	0.000
site	34	0.971	0.961	0.951	0.186	0.500	0.000
director	38	0.974	0.930	0.965	0.289	0.711	0.000
segment	48	0.736	0.764	0.799	0.389	0.771	0.000
instruments	58	0.977	0.977	0.977	0.460	0.603	0.000
downlink (TB)	46	0.703	0.681	0.971	0.471	0.696	0.000
orbit days	56	0.464	0.506	0.982	0.351	0.714	0.000
launch date	60	0.867	0.856	0.956	0.228	0.750	0.000
rendezvous date	60	0.850	0.844	0.944	0.306	0.783	0.000
yes/no feature (outside grammar)	160	0.556	0.535	0.523	0.700	0.794	0.537

field failing either is rejected and scored as not extracted. Mean extraction F1 is 0.974; Table 9 reports FLO accuracy by extraction-F1 bin, arm populations, and the FLO error accounting (179 code-side vs. 45 extracted-wrong + 27 not-extracted). One of 100 documents yielded no extracted facts and is scored with its initialization-only code.

H Compute and Release

Experiments ran on two RTX 5090 GPUs and total roughly 60 GPU-hours, of which about 15 are the matched-budget comparison of §5.4. Optimizing one code takes a median of about 50 s

Table 8: Per-document cost by supervision source (generation and token columns are means over documents and seeds, optimization time the median; generation on one RTX 5090 with the frozen reader as generator; code optimization identical in every arm). Training tokens count question and answer tokens of the $108\times$facts-row stream.

Supervision source	gen. prompt tok.	gen. output tok.	gen. s	train tok.	optim. s (median)
Platform 1					
Compiled grammar (QGC)	0	0	0	186,521	51.5
Grammar, cells sampled	0	0	0	186,521	52.4
LLM paraphrases, per fact	7,167	33,722	45	187,909	52.3
Unconstrained self-study	216,961	57,379	251	167,402	51.6
Platform 2					
Compiled grammar (QGC)	0	0	0	180,414	51.2
Grammar, cells sampled	0	0	0	180,414	51.2
LLM paraphrases, per fact	6,702	31,259	46	179,423	51.7
Unconstrained self-study	182,336	48,346	191	157,554	46.0

Table 9: Deployment detail (ClinicalTrials.gov). Top: descriptive arm means with their populations (gate arms use the 60-document gate population; ladder arms use all 100 documents). Middle: FLO accuracy by extraction-F1 bin. Bottom: error accounting for the FLO code arm, pooled over three seeds.

Arm	accuracy	population
BASE	0.708	60 docs
F	0.767	60 docs
L	0.788	60 docs
O	0.820	60 docs
FLO code	0.910	100 docs
λ=0.75 code	0.905	100 docs
full text (zero-shot)	0.869	100 docs
no context	0.074	100 docs
extracted facts	0.948	100 docs
gold facts	0.961	100 docs
FLO accuracy by extraction F1		
F1 < 0.5	0.100	1 doc
F1 $\in [0.5, 0.8)$	0.630	1 doc
F1 $\in [0.8, 1)$	0.838	12 docs
F1 $= 1$	0.933	86 docs
FLO error accounting (count)		
code answered wrong	179	
extracted wrong	45	
not extracted	27	

for a synthetic document (40–70 admitted facts) and 10 s for a deployment document. We release the compiler, the analysis, per-document results for every arm and seed, and the figure and table generators.