PROXIMITY, PRECEDENCE AND POSITION: A MINIMUM LINEAR ARRANGEMENT ACCOUNT OF ORDERING EFFECTS IN LLM PROMPTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context learning is highly sensitive to the ordering of demonstrations, but the root factors of context ordering remain undiscovered. Existing work treats a demonstration set as an unordered bag and searches over permutations based on a certain metric, without modeling the underlying relational structure. We give a structural account of order sensitivity built on three key properties of an arrangement: *proximity* (how far apart dependent facts sit), *precedence* (whether each fact appears after the facts it depends on), and *position* (where a fact sits in the context). We model proximity and precedence using minimum linear arrangement (MinLA), with precedence as a hard constraint and isolate position through controlled experiments. We treat the MinLA-constrained order as an oracle and measure how much of the total order effect it recovers relative to the worst achievable order. This provides a direct measure of how well our account explains observed behavior. On small-scale instances with known ground-truth ordering accuracy, it recovers 81% of the order effect in the referential domain and 63% in the associative domain. Across two domains with different dependency semantics – referential (GSM8K arithmetic, where a step reuses an unresolved prior value) and associative (synthetic narratives, where a sentence names a concept introduced earlier) – orders that are both MinLA-optimal and precedence-legal outperform the natural order by +34.5% accuracy in arithmetic and +16.3% in narrative on Llama-3.2-3B-Instruct, while MinLA optimization without the legality constraint does not reliably help. Decomposing the order effect into clustering (proximity) and direction (precedence), direction accounts for 61% of it in arithmetic and clustering for 56% in narrative, consistent with a referential-versus-associative account of how each domain encodes dependency. A third, independent, U-shaped position effect, isolated under experimental control, connects our results to lost-in-the-middle phenomena reported in previous work only observationally.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities across question answering (Min et al., 2024), dialogue (Feng et al., 2023), reasoning, and code generation (He et al., 2024). In-context learning (ICL) (Brown et al., 2020) lets models adapt to new tasks – Formal Language Learning (Ghosh et al., 2026), Entity Recognition (Chen et al., 2023), Knowledge Editing (Zheng et al., 2023) – by conditioning on demonstrations in the prompt rather than updating parameters. Despite its success, ICL is highly sensitive to the order demonstrations are presented in (Li et al., 2025b; Zhang et al., 2025; Nori et al., 2023; Lu et al., 2022), but *why* remains unsettled: is it proximity (how far apart related facts sit), precedence/direction (whether a reference points at something already shown), or position (where a fact sits relative to the context boundary)? Prior work treats a demonstration set as flat and unordered, searching over permutations for one satisfying a label-fairness or output-likelihood criterion (Guo et al., 2024; Xu et al., 2024; Bhope et al., 2025), without asking whether the demonstrations carry relational structure the ordering problem should respect. Separately, work using graphs to represent reasoning (He et al., 2026) builds a dependency structure inside a single example for retrieval, but never addresses how multiple dependency-linked facts should be laid out in a prompt.

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.

What is missing is a structural account of order sensitivity: one that predicts when rearranging related facts helps or hurts, identifies which properties of an arrangement drive the effect, and explains why those properties matter differently across domains. We examine three key properties: *proximity*, *precedence*, and *position*. We model proximity and precedence using minimum linear arrangement (MinLA) with precedence as a hard constraint. MinLA provides a natural model of proximity because it orders nodes to minimize the total distance between connected pairs. We isolate position separately through controlled experiments. We hypothesize that proximity and precedence trade off differently depending on whether a dependency is referential (a step needs the value of an earlier step establishes, as in arithmetic) or associative (a sentence names a concept introduced earlier, as in narrative).

This motivates three questions. **1:** Do proximity and precedence explain **reordering effects** as more distracting facts are added? **2:** How much of the order effect comes from **clustering vs. precedence**, and does this differ between referential and associative dependencies? **3:** Beyond clustering and precedence, does a fact's **position in the prompt** affect how reliably the model uses it?

We construct dependency-annotated datasets in two domains: arithmetic word problems (GSM8K), where a step's correct interpretation requires substituting a specific prior value with no fallback, and synthetic narratives, where a sentence names a concept introduced earlier. We scale each to up to 45 interleaved facts, using an approximate MinLA solver to instantiate orders (Section 4.4, evaluate accuracy under natural, MinLA-optimized, and precedence-violating orderings, decompose the effect into clustering and direction, and run controlled experiments isolating positional distance and absolute position.

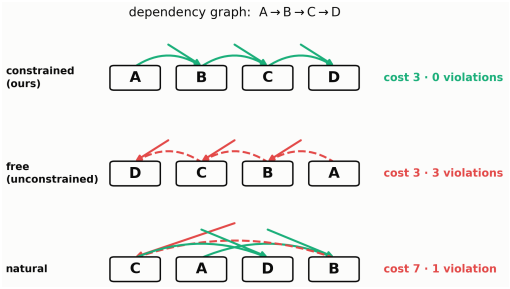


Figure 1: The same four-fact chain under three orderings. *Constrained* (top, cost 3, 0 violations): MinLA-optimal subject to precedence legality. *Free* (middle, cost 3, 3 violations): globally MinLA-optimal, indifferent to direction. *Natural* (bottom, cost 7, 1 violation): an arbitrary source order. Solid green arcs mark legal edges; dashed red arcs mark violations.

Figure 1 illustrates the central argument with a toy four-fact chain A→B→C→D: MinLA assigns each ordering a cost (the total distance between dependency-linked facts, summed over all such pairs) and the globally MinLA-optimal order (*Free*) clusters the chain just as tightly as the precedence-legal order (*Constrained*), both reaching the same minimal cost of 3, yet *Free* lays the chain out exactly backwards while *Constrained* reads forward; cost alone therefore cannot be the whole story.

Our main contribution is a structural account of order sensitivity among dependency-linked facts in a prompt. Specifically:

1. treated as an oracle order, it explains reordering effects reliably (+34.5% accuracy in arithmetic, +16.3% in narrative over natural ordering) and grows as prompt length scales from 10 to 45 facts;
2. we decompose the order effect into clustering and direction, and show their shares differ by domain (direction ≈61% in arithmetic, clustering ≈56% in narrative);
3. on small-scale instances scored exactly, this oracle recovers 81% of the achievable order effect in the referential domain and 63% in the associative domain;
4. we propose and test a referential-versus-associative account of this asymmetry – arithmetic dependencies require resolving an otherwise-unresolvable symbolic reference, while narrative dependencies remain recoverable from surface semantics wherever they sit;
5. we identify a U-shaped absolute-position effect, independent of clustering and direction;
6. the account holds across a four-model size ladder (Qwen3 4B–32B) and non-linear dependency structure, with scale eroding the benefit of clustering but not of precedence.

2 Related Works

ICL Retrieval, Ordering, and Position Bias. ICL's sensitivity to how a prompt is built is well established (Liu et al., 2022; Lu et al., 2022; Zhao et al., 2021), driving work on retrieval strategies – lexical (Robertson & Zaragoza, 2009), dense-embedding (Reimers & Gurevych, 2019; Rubin et al., 2022; Li & Qiu, 2023; Wang et al., 2024), diversity- and active-learning-aware (Si et al., 2023; Zhang et al., 2022b; Li et al., 2023; Zhang et al., 2022a; Margatina et al., 2023), and retrieval-augmented (Guu et al., 2020; Lewis et al., 2020; Shi et al., 2022; Yasunaga et al., 2022), with recent adaptive variants deciding dynamically what to include (Wu et al., 2023; Moskvoretskii et al., 2025; Nafee et al., 2025). A parallel theoretical line explains *how* demonstrations exert influence, as an implicit forward-pass optimization (Akyürek et al., 2022; Xie et al., 2021; Dai et al., 2023; Von Oswald et al., 2023), but treats a demonstration's contribution as emergent rather than as a design variable prompt construction should account for.

A separate line treats the demonstration set as fixed and flat, searching its $k!$ orderings for one satisfying a scoring criterion: content-free-input calibration (Guo et al., 2024) or closeness to a label prior (Xu et al., 2024) (both requiring a fixed label set, with reported degradation on non-classification tasks), or the log-probability the model assigns its own stripped-prompt output (Bhope et al., 2025). Li et al. (2025a) shows ordering alone contributes as much variance as demonstration selection, and neither transfers across datasets. Others treat ordering sensitivity as a property of the model to measure: Liu et al. (2024) find a U-shaped middle-of-input accuracy drop that recurs, as we show in Section 5.4, at far shorter prompt lengths than their long-context regime; Xiang et al. (2024) trace one source to the causal attention mask; Zhang et al. (2024) aggregate one-shot passes' meta-gradients to make predictions largely order-insensitive. None examines relational structure among demonstrations beyond sequence position. A separate line uses graphs in ICL for retrieval rather than arrangement – over mined predicate co-occurrence (He et al., 2026), text-attributed graphs (Sun et al., 2025), reasoning-graph kernels (Lin et al., 2025), or knowledge-graph paths (Tan et al., 2025) – governing what to retrieve, not how a chosen set of k demonstrations should be arranged.

Linear Arrangement and Precedence-Constrained Sequencing. MinLA, placing the nodes of a graph on a line to minimize total edge length, is a classical NP-hard layout problem (Garey et al., 1976; Diaz et al., 2000), and sequencing jobs under precedence constraints to minimize a weighted cost is a classical scheduling problem (Lawler, 1978). We borrow these formulations as a model of how dependency structure interacts with arrangement in a prompt; prior work on prompt ordering has not modeled the arrangement of dependency-linked facts in either form.

3 Problem Formulation

3.1 In-Context Prompts as Ordered Fact Sequences

A task instance consists of a query q and a set of n facts $F = \{f_1, ..., f_n\}$, indexed by the order they appear in originally, that jointly support answering q. Unlike the classical few-shot setting, where each demonstration is independent and self-contained, the facts in F are the constituent steps of a single reasoning chain (arithmetic domain) or statements of a single narrative (NL domain): the model must synthesize across all n of them, not select among them. A prompt is constructed by choosing a permutation $\pi \in \Pi(F)$, a bijection $\pi : \{1, ..., n\} \to \{1, ..., n\}$ assigning each fact's original index i a display position $\pi(i)$. We write

$$\mathrm{pos}_\pi(f_i) := \pi(i) \in \{1, ..., n\} \tag{1}$$

for the position fact f_i occupies under π. Rendering F in the order induced by π and appending q yields the prompt

$$P_\pi(F, q) = \left[f_{\pi^{-1}(1)},\ f_{\pi^{-1}(2)},\ \ldots,\ f_{\pi^{-1}(n)} \right] \oplus q, \tag{2}$$

where $\pi^{-1}(k) \in \{1, ..., n\}$ is the original index of whichever fact is assigned display position k, so $f_{\pi^{-1}(k)}$ is simply the fact occupying position k: reading k from 1 to n lists the facts in their new, rendered order. This is passed to a language model M, which produces an answer $\hat{y} = M(P_\pi(F, q))$. Throughout, we assume M, F and q are held fixed for a given instance and the manipulated variable is π. Our object of study is how the choice of π affects the correctness of $\hat{y}$.

3.2 Dependency Graphs over Facts

We represent the relational structure among facts as a directed acyclic graph $G = (F, E, w)$, where an edge $(f_j, f_i) \in E$ indicates that f_i cannot be correctly interpreted without the value or concept f_j establishes, and w assigns a nonnegative weight used by the cost function of Section 3.3. G is acyclic by construction: a fact can only depend on one that precedes it in the underlying solution or narrative. Precedence legality (Section 3.3) is defined on E identically in both domains; w, described concretely below, is instantiated differently by domain. We instantiate G in two domains that realize the dependency relation through different linguistic forms.

Referential dependency (arithmetic domain, §3.2). We follow GSM8K's calculator-annotated solution format (Cobbe et al., 2021) – one fact per step, each step's own computed value masked out – but generate problems synthetically rather than drawing on GSM8K's own problem set, since real GSM8K solutions rarely exceed 8 annotated steps and cannot reach the $n \in \{10, ..., 45\}$-fact range this paper's main results require (Section 5.1 describes the generator). If a later step f_i reuses an earlier step f_j's value verbatim, f_i's text contains an unresolved placeholder $\{\texttt{REF}:j\}$, resolved to `[Fact pos`$_\pi(f_j)$`]` only at prompt-construction time for the specific π being rendered, so G (built once from the original solution) is invariant to π: $E = \{(f_j, f_i) : f_i$'s text contains $\{\texttt{REF}:j\}\}$.

Associative dependency (narrative domain, §3.2). We construct synthetic multi-fact word problems with the same edge semantics, realized through recoverable surface form rather than an unresolved symbol: a child fact names its parent's concept in prose (e.g., "the bakery's running total") rather than leaving a placeholder, and E is built directly from the narrative's ground-truth dependency structure at generation time. Unlike in the arithmetic domain, a fact's referent stays nameable regardless of where f_j falls relative to f_i.

3.3 The Minimum Linear Arrangement Objective

Given G, the cost of a permutation π is the sum of weighted linear distances between dependency-linked facts:

$$\mathrm{cost}(\pi) \;=\; \sum\nolimits_{(f_j,\, f_i) \in E} w_{ji} \,\big|\, \mathrm{pos}_\pi(f_i) - \mathrm{pos}_\pi(f_j) \,\big|. \tag{3}$$

w is fixed per domain before any ordering is chosen. In the narrative domain, $w_{ji} = 1$ for every $(f_j, f_i) \in E$. In the arithmetic domain, we instead score every pair of facts with a gold-free heuristic, $w_{ji} = \max\big(1.0 \cdot \mathrm{numref}(f_j, f_i),\ 0.3 \cdot \mathrm{ngram}(f_j, f_i)\big)$ for all $j \neq i$, where $\mathrm{numref}(f_j, f_i) \in \{0, 1\}$ indicates that one fact's computed numeric result reappears as a literal number token in the other fact's text, and $\mathrm{ngram}(f_j, f_i) \in [0, 1]$ is the character-4-gram Jaccard similarity between their raw text; $\mathrm{cost}(\pi)$ for this domain is then the corresponding sum over all pairs, not only those in E. This asymmetry reflects how each domain's dependencies are recovered in practice: the narrative domain's edges come from the generator's ground-truth structure, so unit weight on E alone is exact, while the arithmetic domain's heuristic approximates E without access to it, at the cost of also placing small nonzero weight on pairs that are not truly dependent. Equation 3 states the general form, of which the narrative domain's unit-weight-on-E instantiation is a special case.

Definition 1. We write $\Pi_{\text{legal}}(G) \subseteq \Pi(F)$ for the set of precedence-legal permutations, and $\mathrm{legal}(\pi) = \big|\{(f_j, f_i) \in E : \mathrm{pos}_\pi(f_j) < \mathrm{pos}_\pi(f_i)\}\big|$ for the number of edges any π satisfies.

Together, $\mathrm{cost}(\pi)$ and precedence legality separate two properties of an order that are logically independent of each other: cost captures *clustering* (how close dependency-linked facts sit to one another), while legality captures *direction* (whether a dependency is displayed in a readable order at all). Section 4 builds directly on this separation. Figure 2 grounds proximity, precedence, and position in one real instance from our dataset before we turn to aggregate results.

4 A Structural Model of Order Effects

4.1 Order Families

Crossing cost-optimality with precedence legality (Definition 1) gives six order families: π_{natural}, the original order; $\pi_{\mathrm{free}} = \mathrm{argmin}_{\pi \in \Pi(F)} \mathrm{cost}(\pi)$, globally cost-optimal and unconstrained

Arithmetic domain, π_{natural}, 5 of 18 facts shown. Positions 1, 3, 5 are a bakery's critical chain; positions 2, 4 are distractor storylines interleaved between them, exactly as rendered in our prompts.

1. At the bakery, it starts with 17 loaves of bread and receives 80 more, so it has $17 + 80 =$___ loaves of bread.
2. At the warehouse, it starts with 72 boxes and adds 23 more, so it has $72 + 23 =$___ boxes.
3. At the bakery, it then ships out 21 more loaves of bread, using its own running total: [Fact 1]$-21 =$___ loaves of bread.
4. At the stable, it starts with 33 bales of hay and gathers 14 more, so it has $33 + 14 =$___ bales of hay.
5. At the bakery, it then adds 88 more loaves of bread, using its own running total: [Fact 3]$+88 =$___ loaves of bread.

Figure 2: A real 5-fact excerpt illustrating all three properties. **Proximity**: the chain's own facts sit two positions apart rather than adjacent, giving $\text{cost} = |3 - 1| + |5 - 3| = 4$ instead of the minimum of 2. **Precedence**: Fact 3 names "[Fact 1]" directly and cannot be resolved before it – showing 3 before 1 breaks the chain. **Position**: holding this same chain fixed and contiguous but sliding it deeper into a much longer context degrades accuracy on its own (Section 5.4), independent of proximity or precedence.

(`minla_recovered`); $\pi_{\text{constrained}} = \text{argmin}_{\pi \in \Pi_{\text{legal}}(G)} \text{cost}(\pi)$, cost-optimal subject to precedence legality as a *hard* constraint (`minla_constrained`), the objective we use to model proximity and precedence together; $\pi_{\text{tiebreak}} = \text{argmin}_{\pi \in \Pi(F)} \big(\text{cost}(\pi), -\text{legal}(\pi)\big)$, legality only breaking cost ties (`minla_tiebreak`); and $\pi_{\text{random}}, \pi_{\text{reversed}}$ as sanity floors. We also report *legal random*, a uniformly random precedence-legal order, which isolates the effect of cost optimization from that of legality alone (Section 4.4 discusses solver quality and why legal random is the cleaner comparison to $\pi_{\text{constrained}}$ at scale).

4.2 Decomposing an Ordering Effect into Clustering and Direction

Section 3.3 defines $\text{cost}(\pi)$ and precedence legality as two formally separate properties of an order (cost is agnostic to direction, legality is agnostic to proximity), giving a direct way to test whether they act independently on accuracy rather than as one conflated "ordering quality" score. For a dependency edge (f_j, f_i), we define its *span* under π as

$$\text{span}_\pi(f_j, f_i) = \big|\, \text{pos}_\pi(f_i) - \text{pos}_\pi(f_j) \,\big|, \tag{4}$$

the same per-edge quantity $\text{cost}(\pi)$ sums over, restricted here to the single critical dependency chain under study. We call an order **clustered** if the maximum span across that chain falls at or below the median span observed across the orders being compared, and **dispersed** otherwise; independently, we call an order **legitimate** if $\pi \in \Pi_{\text{legal}}(G)$, and **illegitimate** otherwise. Crossing these two binary properties isolates a **direction-only effect** (accuracy difference between legitimate and illegitimate orders, clustering held fixed) and a **clustering-only effect** (accuracy difference between clustered and dispersed orders, legitimacy held fixed). If cost and legality act on accuracy independently, as their formal separation suggests, these two effects should be additive and domain-general in sign, even if their relative magnitude differs by domain.

To express each component as a share of the total, we define the *order effect* as the accuracy difference between the best and worst cells of the 2×2 design (clustered and legitimate versus dispersed and illegitimate) and attribute it along the path dispersed-illegitimate $\rightarrow$ clustered-illegitimate $\rightarrow$ clustered-legitimate.

4.3 An Referential-versus-Associative Account of Domain Differences

Why should direction carry more of the order effect in one domain and clustering in another? We hypothesize a difference in how each domain realizes a dependency edge. Arithmetic dependencies are **referential**: a fact's correct interpretation requires literally substituting an antecedent's value – the $\{\texttt{REF}:j\}$ placeholder of Section 3.2 is analogous to a variable-use-before-definition problem, with no fallback if the antecedent has not yet been shown. Narrative dependencies (Section 3.2) are **associative**: a fact names its parent concept directly in prose, so the relationship stays recoverable from surface semantics wherever the text sits, with nothing left as an unresolved symbol. This

Table 1: Accuracy by order family and domain, at $n \in \{10, ..., 45\}$ facts. Values are mean $\pm$ sample standard deviation across two independent seeds for both domains.

Domain	π_{natural}	π_{free}	$\pi_{\text{constrained}}$	π_{tiebreak}
GSM8K	0.510 ± 0.028	0.488 ± 0.011	$\mathbf{0.855 \pm 0.014}$	0.508 ± 0.004
NL	0.705 ± 0.021	0.600 ± 0.035	$\mathbf{0.868 \pm 0.039}$	0.543 ± 0.067

predicts that the direction share of the order effect should be substantially larger in the referential (arithmetic) domain than in the associative (narrative) domain. This is tested directly in Section 5.3.

4.4 Computing the Orders

Exact search over $\Pi_{\text{legal}}(G)$ or $\Pi(F)$ is $O(n!)$, and MinLA is NP-hard in general (Garey et al., 1976), closely related to precedence-constrained sequencing in scheduling (Lawler, 1978). At the scales we study, we use an $O(n^2)$ approximate solver – randomized topological construction for a legal starting point, greedy legal-position insertion, then legal-only 2-opt/or-opt local search – with the legality check removed to compute π_{free}, or only breaking ties to compute π_{tiebreak}. Because π_{free} and $\pi_{\text{constrained}}$ are both approximate at these scales, and the constrained search starts from a much smaller, legal-only space, π_{free}'s cost can exceed $\pi_{\text{constrained}}$'s – so a π_{free}-vs-$\pi_{\text{constrained}}$ comparison mixes precedence with solver quality, while $\pi_{\text{constrained}}$-vs-legal-random (both legal, differing only in cost optimization) does not. The solver's role is to make the MinLA oracle computable at scale, not to serve as a deployable ordering procedure: it matches exact brute-force on 293/300 instances at $n \leq 8$, matches certified CP-SAT on 20/24 certifiable instances at $n \in [15, 20]$, and never returns an illegal order (Appendix A.9). At $n \leq 4$ we bypass it entirely and enumerate all permutations, an exact oracle (Section 5.3).

5 Experiments

5.1 Experimental Setup

A branched-distractor synthetic generator, built separately for the arithmetic and narrative domains, decouples problem size from problem depth: a short critical dependency chain (4-7 facts) determines the gold answer, padded with independent distractor chains interleaved round-robin to reach $n \in \{10, ..., 45\}$ facts total (the dataset behind Sections 5.2-5.3). A second, purpose-built distance/position generator (Section 5.4) embeds a single referenced pair in k total facts, holding either the *gap* between them or their *absolute position* fixed while the other varies. Unless noted, experiments use Llama-3.2-3B-Instruct with greedy decoding (Section 5.5 repeats the main comparison on Qwen3 4B–32B and Llama-3.1-8B-Instruct), facts rendered per their domain's representation (Section 3.2), and accuracy scored against the gold numeric answer, gated on low and comparable null-prediction rates across conditions (generation length scales with n to avoid exhausting the budget). Each reported condition covers 200 problems at $n \in \{10, ..., 45\}$ facts across all six order families; GSM8K and narrative are each run under two independent seeds. Two small-scale ($n = 4$) datasets with every order enumerated, Python function-call chains (a referential code domain) and short natural-language narratives (associative), are used in Sections 5.3 and 5.5.

5.2 Accuracy by Order Family

Table 1 reports accuracy for all six order families in both domains at $n \in \{10, ..., 45\}$ facts (mean $\pm$ sample standard deviation, two seeds). $\pi_{\text{constrained}}$ outperforms π_{natural} by a wide margin in both domains ($+34.5$ points GSM8K; $+16.3$ narrative), an order of magnitude larger than the seed-to-seed standard deviation in either domain.

Figure 3 breaks this down by context size: π_{natural} degrades steadily as distractor facts are added ($0.854 \rightarrow 0.179$ in GSM8K) while $\pi_{\text{constrained}}$ stays flat, and the gap widens (full per-bucket numbers, both seeds, in Appendix Tables 3-10).

π_{free} does not outperform π_{natural} in either domain (0.488 vs. 0.510 GSM8K; 0.600 vs. 0.705 narrative), so clustering alone does not help without a precedence guarantee. π_{tiebreak} tracks π_{free}

rather than $\pi_{\text{constrained}}$, since it is never fully precedence-legal at these scales (0/200 instances; Appendix A.6). $\pi_{\text{constrained}}$ over legal random shows cost optimization adds benefit beyond legality alone; π_{random} and π_{reversed} confirm accuracy is far from order-invariant. The $\pi_{\text{constrained}}$-over-π_{natural} gap itself differs by domain (34.5 vs. 16.3 points) even though narrative's natural-order baseline starts higher (0.705 vs. 0.510) – foreshadowing the referential-versus-associative account, tested directly in Section 5.3.

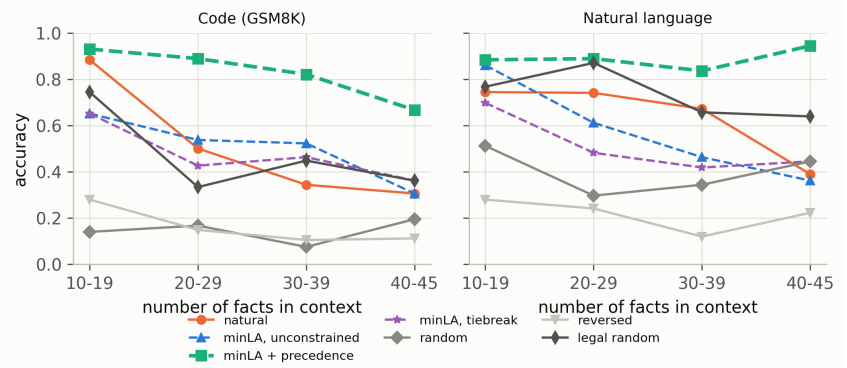


Figure 3: Accuracy vs. number of facts in context in all six order families plus legal random, arithmetic (left) and narrative (right). $\pi_{\text{constrained}}$ (green, square) stays flat in both domains; π_{natural} (orange, circle) declines steadily. π_{free} (blue, dashed triangle) and π_{tiebreak} (purple, dashed star) fall between the two; legal random (dark gray, thin diamond) shows a legal but random topological order; π_{random} (gray, diamond) and π_{reversed} (gray, downright triangle) mark the floor.

Figure 4 shows one real 11-fact instance, same 6-fact critical chain, scattered (left) vs. contiguous (right). Only the last link's position differs, and that alone separates answering 143 from 189.

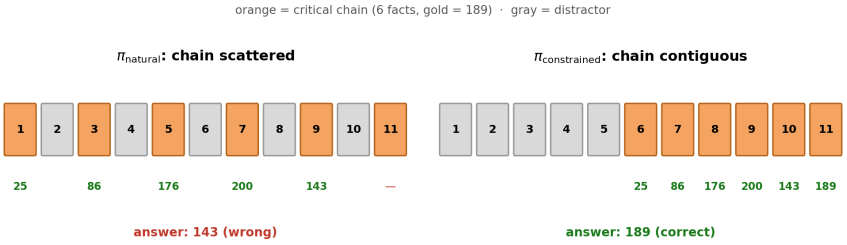


Figure 4: One real instance's critical chain (orange) among distractors (gray). Left (π_{natural}): scattered; the model works through five links but never reaches the sixth, answering 143. Right ($\pi_{\text{constrained}}$): contiguous; it reads through to 189.

5.3 Decomposing the Order Effect

Section 5.2 shows that constraining order to be precedence-legal produces a large accuracy gain. What is it about a legal order that helps? Section 4.2 separates this into two logically independent properties: whether the critical dependency chain sits *contiguously* in context (clustering), and whether it is read in a *forward*, precedence-respecting direction (direction).

Each problem's critical dependency chain is classified as contiguous or scattered under π_{free}, using the generator's own branch labels. Comparing contiguous against scattered accuracy within π_{free} isolates the clustering-only effect; comparing contiguous π_{free} against $\pi_{\text{constrained}}$ isolates the direction-only effect (raw cells in Table 2). Both effects are positive in both domains, but their shares of the order effect differ as the referential-versus-associative account (Section 4.3) predicts. In arithmetic, direction accounts for $+0.356 \pm 0.003$ and clustering for $+0.230$, i.e. about 61% and 39% of the order effect; in narrative, clustering accounts for $+0.260$ and direction for $+0.201$, about 56% and 44%. The asymmetry is therefore one of degree: clustering carries somewhat more of the effect in the narrative domain, whose contiguous-but-undirected accuracy already sits well above arithmetic's in absolute terms, while direction still contributes substantially. The two components are approximately additive where both cells are populated (in arithmetic, the direction effect is +0.359

among contiguous order and +0.274 among scattered ones); the narrative domain's scattered-and-legal cell is empty, so additivity cannot be tested there. Legality alone still helps even without full clustering, as seen by GSM8K's scattered-but-legal condition outperforming π_{natural}.

Table 2: Accuracy by clustering (scattered/contiguous) and legality (π_{free}/$\pi_{\text{constrained}}$), both domains. Sample size n per cell in parentheses.

	π_{free} (illegitimate)	$\pi_{\text{constrained}}$ (legitimate)
GSM8K, scattered	0.319 (47)	0.593 (27)
GSM8K, contiguous	0.549 (153)	0.908 (173)
Narrative, scattered	0.434 (53)	n/a (0)
Narrative, contiguous	0.694 (147)	0.895 (200)

Small-scale check: how much does the oracle explain? We repeat the decomposition on the two small-scale ($n = 4$) datasets, with all 24 orders of each of 100 problems scored, and ask how much of the achievable order effect the MinLA oracle recovers, treating each problem's best- and worst-scoring permutation as its ceiling and floor. In the code dataset the direction-only effect is $+0.151$ and the clustering-only effect is $+0.102$; in narrative neither factor is individually detectable (-0.027, -0.022). Yet $\pi_{\text{constrained}}$ – our oracle order – still recovers most of each domain's oracle-to-floor range: 81% in the referential (code) domain (oracle 1.000, floor 0.000, $\pi_{\text{constrained}}$ 0.806) and 63% in the associative (narrative) domain (oracle 0.950, floor 0.100, $\pi_{\text{constrained}}$ 0.639), where 63/100 problems per domain have several tied minimum-cost legal permutations and we report the mean accuracy across ties. Proximity and precedence jointly explain most, but not all, of the order effect, more completely so where dependencies are referential, consistent with Section 4.3.

5.4 Position and Distance Effects

A natural alternative explanation for Sections 5.2- 5.3 is that raw positional distance between dependent facts, independent of clustering or direction, is what drives accuracy. A controlled experiment ($k = 5$ facts, gap 1-4 between the referenced pair, legality never violated) finds no such effect: accuracy is essentially flat across gap 1 through 4 (Figure 5, left). Holding the gap fixed at 1 but moving where the pair sits within a much larger context ($k = 45$) isolates absolute position instead, and reveals a distinct effect (Figure 5, right): a clear U-shape, with accuracy far higher at the start and end of the context (0.947, 0.933) than in the middle (0.747) – the familiar "lost in the middle" phenomenon (Liu et al., 2024), demonstrated here under experimental control.

A second $k = 45$ experiment (Appendix Table 11) varies gap instead: it pins the final fact at the last slot and sweeps the root fact's slot from 0 to 43. Both experiments, re-expressed on the root fact's shared slot axis, dip toward the middle and recover toward both edges, converging almost exactly at their shared midpoint (slot 22: 0.747 in both; Figure 5). Because the two designs trace one curve on a shared axis, and the $k = 5$ full shows no distance effect, we treat position as a single effect that is separate from proximity, rather than as a further consequence of the MinLA objective.

5.5 Scope and Robustness Checks

Sections 5.2-5.4 establish the effect on one model (Llama-3.2-3B-Instruct) using the true dependency graph G and comparing against internal order families. We report three further checks that delimit the scope of the account, addressing, respectively, model generality, whether G must be given rather than recovered, and how a realistic ordering-selection method compares.

Cross-model generalization. In Figure 6, we repeat on the Qwen3 size ladder (4B–32B) and Llama-3.1-8B-Instruct for GSM8K; full protocol, per-model tables, and caveats (thinking-mode truncation, Qwen3-8B looping under shuffled orders) are in Appendix A.2. $\pi_{\text{constrained}}$ has the highest mean accuracy in seven of eight Qwen3 model-domain cells. Scale substitutes for proximity but not precedence: the $\pi_{\text{constrained}}$-over-legal-random gap shrinks from $+5.5$ points at 4B to $+0.7$ at 32B, while the gap to π_{reversed} stays wide ($+51.2$ to $+41.9$) – a 4B model given $\pi_{\text{constrained}}$ still beats a 32B model given π_{reversed} by 40.7 points. Llama-3.1-8B shows the same pattern with a smaller gap (Appendix Table 6).

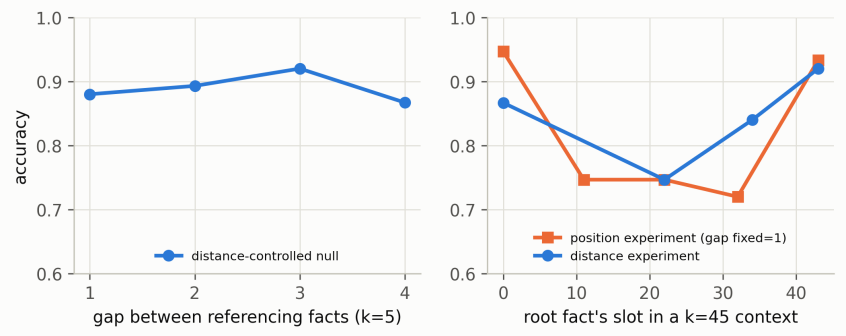


Figure 5: Left: accuracy vs. gap between the referenced pair at $k = 5$ facts (distance-controlled null) – flat within noise. Right: accuracy vs. the root fact's slot in a $k = 45$-fact context, from two experiments on a shared x-axis: position (orange squares, gap fixed at 1) and distance (blue circles, final fact pinned at the last slot). Both dip toward the middle and recover toward both edges, converging near the shared midpoint (slot 22).

Non-linear structure and automatic graph extraction. Two non-linear DAG topologies (chain, V-shape) replicate the central pattern (Appendix Table 7): $\pi_{\text{constrained}}$ exceeds π_{natural} by 26.5 and 10.0 points. Recovering G automatically (5 sampled model responses, parsed for revealed dependencies, at small scale) reaches 0.940 mean accuracy on code – nearly free, since resolving a reference is already required to execute the prompt – but only 0.580 on narrative (best-order ceiling 0.950), failing entirely in 21% of cases: the referential-versus-associative gap again.

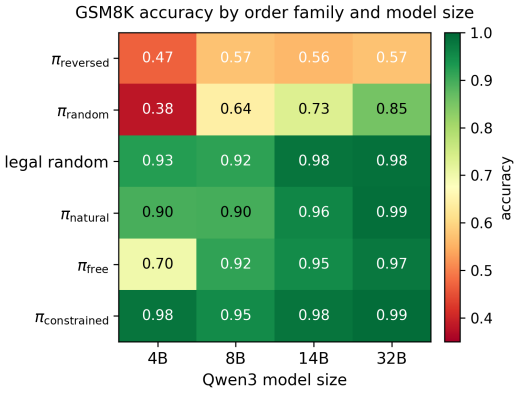


Figure 6: Accuracy of six order families across the Qwen3 size ladder (4B–32B). $\pi_{\text{constrained}}$ has the highest mean accuracy in seven of eight Qwen3 model-domain cells.

Comparison to a realistic ordering-selection baseline. OptiSeq (Bhope et al., 2025) ($K = 16$ candidates, argmax by stripped-prompt log-probability – DEmO (Guo et al., 2024) and PDO (Xu et al., 2024) require a fixed label set and degrade on non-classification tasks) reaches 0.615 in GSM8K and 0.655 in narrative, well below $\pi_{\text{constrained}}$ (0.855, 0.868) and, in narrative, below $\pi_{\text{natural}} = 0.705$; its picks are not concentrated on $\pi_{\text{constrained}}$ either (18.5%, 10.0% of the time; Appendix Table 8). A log-probability criterion, unlike the MinLA oracle, has no notion of dependency structure to condition its search on.

6 Conclusion

We gave a structural account of ordering effects in LLM prompts, built on three properties of an arrangement: proximity, precedence and position. Proximity and precedence, modeled by minimum linear arrangement, recover a large accuracy gap over natural order that grows as distractor facts are added, and on small-scale instances explain 81% (referential) and 63% (associative) of the achievable order effect. The order effect decomposes into clustering and direction, whose shares differ by domain as a referential-versus-associative account predicts: direction dominates where reading requires resolving an unresolved reference (arithmetic), clustering where meaning survives relocation (narrative). A third, independent, U-shaped position effect connects to lost-in-the-middle phenomena reported elsewhere only observationally, and the account generalizes across model scale, non-linear structure, and a realistic ordering-selection baseline.

AI USE STATEMENT

In this work, we used generative AI tools to assist with editing and polishing the paper, primarily using as a tool to improve the wording, grammar, flow, and organization of the paper. We also used generative AI for developing figures (specifically Figures 1 and 4). We have not used generative AI tools for developing code or ideas presented in this work. We have reviewed all AI-assisted work and reviewed and revised all suggested writing improvements to ensure it accurately reflects the ideas, results and conclusions we work to present. All AI-assisted text was reviewed and revised by the authors to ensure that it accurately reflects the methods, results, and conclusions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Rahul Atul Bhope, Praveen Venkateswaran, KR Jayaram, Vatche Isahagian, Vinod Muthusamy, and Nalini Venkatasubramanian. Optiseq: Ordering examples on-the-fly for in-context learning. In *EMNLP (Findings)*, pp. 24864–24887, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13661–13675, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.

J. Diaz, J. Petit, and M. Serna. *A Survey on graph layout problems*. Number LSI–00–61–R. October 2000. URL https://hdl.handle.net/2117/96468.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. Towards llm-driven dialogue state tracking. 2023. URL https://arxiv.org/abs/2310.14970.

M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified np-complete graph problems. *Theor. Comput. Sci.*, 1:237–267, 1976. URL https://api.semanticscholar.org/CorpusID:205090369.

Bishwamittra Ghosh, Soumi Das, Till Speicher, Qinyuan Wu, Mohammad Aflah Khan, Deepak Garg, Krishna P Gummadi, and Evimaria Terzi. Fine-tuning vs. in-context learning in large language models: A formal language learning perspective. *arXiv preprint arXiv:2604.23267*, 2026.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. What makes a good order of examples in in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14892–14904, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.884. URL https://aclanthology.org/2024.findings-acl.884/.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Runyu He, Anyu Ying, and Xiaoyu Hu. Improving opendevin: Boosting code generation llm through advanced memory management. *Applied and Computational Engineering*, 68:313–320, 07 2024. doi: 10.54254/2755-2721/68/20241506.

Yao He, Jiajun Liu, Wantao Wen, Yikai Guo, Hanchao Chen, Yue Ma, Wenjun Ke, and Peng Wang. Logicicl: Towards instructive logical-aware demonstrations for stimulating llms mathematical reasoning ability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 10:1669–1683, 2026. URL https://api.semanticscholar.org/CorpusID:285442107.

Eugene L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of discrete mathematics*, 2:75–90, 1978. URL https://api.semanticscholar.org/CorpusID:118733237.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Warren Li, Yi-Qing Wang, Zihan Wang, and Jingbo Shang. Order matters: Rethinking prompt construction in in-context learning. *ArXiv*, abs/2511.09700, 2025a. URL https://api.semanticscholar.org/CorpusID:282992398.

Warren Li, Yiqian Wang, Zihan Wang, and Jingbo Shang. Order matters: Rethinking prompt construction in in-context learning. *arXiv preprint arXiv:2511.09700*, 2025b.

Xiaonan Li and Xipeng Qiu. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6354–6374, 2023.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 5315–5333, 2023.

Yukang Lin, Bingchen Zhong, Shuoran Jiang, Joanna Siebert, and Qingcai Chen. Reasoning graph enhanced exemplars retrieval for in-context learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9737–9759, 2025.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić (eds.), *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5011–5034, 2023.

Dehai Min, Nan Hu, Rihui Jin, Nuo Lin, Jiaoyan Chen, Yongrui Chen, Yu Li, Guilin Qi, Yun Li, Nijun Li, and Qianren Wang. Exploring the impact of table-to-text methods on augmenting llm-based question answering with domain hybrid data. 2024. URL https://arxiv.org/abs/2402.12869.

Viktor Moskvoretskii, Maria Marina, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. Adaptive retrieval without self-knowledge? bringing uncertainty back home. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6355–6384, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.319. URL https://aclanthology.org/2025.acl-long.319/.

Mahmud Wasif Nafee, Maiqi Jiang, Haipeng Chen, and Yanfu Zhang. Dynamic retriever for in-context knowledge editing via policy optimization, 2025. URL https://arxiv.org/abs/2510.21059.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. volume 3, pp. 333–389, 01 2009. doi: 10.1561/1500000019.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2655–2671, 2022.

Peng Shi, Rui Zhang, Richard He Bai, and Jimmy Lin. Xricl: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5248–5259, 2022.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*, 2023.

Yuanfu Sun, Zhengnan Ma, Yi Fang, Jing Ma, and Qiaoyu Tan. Graphicl: Unlocking graph learning potential in llms through structured prompt design. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2440–2459, 2025.

Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. Paths-over-graph: Knowledge graph empowered large language model reasoning. In *Proceedings of the ACM on Web Conference 2025*, pp. 3505–3522, 2025.

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*, pp. 35151–35174. PMLR, 2023.

Liang Wang, Nan Yang, and Furu Wei. Learning to retrieve in-context examples for large language models. 2024. URL https://arxiv.org/abs/2307.07164.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1423–1436, 2023.

Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. Addressing order sensitivity of in-context demonstration examples in causal language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6467–6481, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Zhichao Xu, Daniel Cohen, Bei Wang, and Vivek Srikumar. In-context example ordering guided by label distributions. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2623–2640, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.167. URL https://aclanthology.org/2024.findings-naacl.167/.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Retrieval-augmented multimodal language modeling. *arXiv preprint arXiv:2211.12561*, 2022.

Kaiyi Zhang, Ang Lv, Yuhan Chen, Hansen Ha, Tao Xu, and Rui Yan. Batch-ICL: Effective, efficient, and order-agnostic in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10728–10739, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.638. URL https://aclanthology.org/2024.findings-acl.638/.

Xiang Zhang, Juntai Cao, Chenyu You, and Dujian Ding. Why prompt design matters and works: A complexity analysis of prompt search space in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32525–32555, 2025.

Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9134–9148, Abu Dhabi, United Arab Emirates, December 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.622. URL https://aclanthology.org/2022.emnlp-main.622/.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022b.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models, 2021. URL https://arxiv.org/abs/2102.09690.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning?, 2023. URL https://arxiv.org/abs/2305.12740.

A Appendix

A.1 Narrative Second Seed: Accuracy by Context-Length Bucket

Table 3 gives the by-bucket breakdown for the narrative domain's second seed (seed 1), underlying Table 1's two-seed mean $\pm$ std for narrative and the summary in Section 5.2. Table 10's narrative row reports seed 0.

Table 3: Narrative accuracy by number of facts in the context window (n), all six order families, seed 1.

Domain	n (count)	π_{natural}	π_{free}	$\pi_{\text{constrained}}$	π_{tiebreak}	π_{random}	π_{reversed}
Narrative	10–19 (41)	0.854	0.561	0.902	0.634	0.390	0.244
Narrative	20–29 (69)	0.783	0.565	0.841	0.478	0.261	0.217
Narrative	30–39 (62)	0.597	0.565	0.790	0.452	0.419	0.097
Narrative	40–45 (28)	0.643	0.643	0.857	0.429	0.321	0.071

A.2 Cross-Model Accuracy: Qwen3 Size Ladder and Llama-3.1-8B-Instruct

Tables 4 and 5 give accuracy for Qwen3-{4B, 8B, 14B, 32B} (all dense; `Qwen3ForCausalLM`) under five order families plus legal random, two seeds, 200 problems per run, $n \in \{10, \ldots, 45\}$ facts. Generation is greedy with thinking mode disabled (`enable_thinking=False`), an answer-token budget of $\max(2048,\ 100n + 512)$ sized from measured response lengths, and a maximum prompt length of 8192 tokens (a run aborts rather than truncating a prompt).

Thinking mode. Qwen3 opens a `<think>` block by default. With our original budget ($50n + 200$ tokens, i.e. 1000 for a 16-fact problem), generation stopped mid-arithmetic before an answer was emitted, and the item was scored wrong, at a rate that tracked order difficulty (5% for $\pi_{\text{constrained}}$ vs. 69% for π_{reversed} at GSM8K-4B). Restricted to generations that finished, NL-8B scored 1.000 under six of seven orders and 0.995 under the seventh: essentially the whole apparent effect in that cell was truncation. The $\pi_{\text{constrained}}$-minus-π_{random} gap on NL-4B was $+0.410$ with thinking on and truncation uncontrolled and is $+0.083$ here. We therefore report the thinking-off runs.

Qwen3-8B. This model has the highest unparsed rate (15.5% for GSM8K π_{reversed}, 7.0% for NL π_{reversed}, versus at most 2% elsewhere). The affected responses are not budget-starved: the model loops ("Step 41: Back to Fact 14", a running total drifting far from the true value) until generation stops. This is a reasoning failure caused by the shuffled order and is counted as incorrect; it is also why $\pi_{\text{constrained}}$ at 8B (0.950) sits below 4B (0.980), so the size trend is not monotone in every row.

Table 4: GSM8K accuracy, Qwen3 size ladder. Mean ± sample standard deviation across two seeds, 200 problems per run.

Order	Qwen3-4B	Qwen3-8B	Qwen3-14B	Qwen3-32B
π_{natural}	0.895 ± 0.014	0.897 ± 0.039	0.958 ± 0.004	0.988 ± 0.004
π_{free}	0.698 ± 0.025	0.915 ± 0.000	0.950 ± 0.014	0.975 ± 0.007
$\pi_{\text{constrained}}$	$\mathbf{0.980} \pm 0.007$	$\mathbf{0.950} \pm 0.000$	$\mathbf{0.982} \pm 0.004$	$\mathbf{0.992} \pm 0.004$
legal random	0.925 ± 0.021	0.917 ± 0.032	0.980 ± 0.007	0.985 ± 0.007
π_{random}	0.383 ± 0.025	0.643 ± 0.060	0.732 ± 0.011	0.855 ± 0.035
π_{reversed}	0.468 ± 0.053	0.573 ± 0.060	0.560 ± 0.028	0.573 ± 0.032

Table 5: Narrative accuracy, Qwen3 size ladder. Same protocol as Table 4.

Order	Qwen3-4B	Qwen3-8B	Qwen3-14B	Qwen3-32B
π_{natural}	0.962 ± 0.004	0.978 ± 0.011	0.998 ± 0.004	$\mathbf{0.998} \pm 0.004$
π_{free}	0.958 ± 0.011	0.920 ± 0.021	0.980 ± 0.007	0.982 ± 0.011
$\pi_{\text{constrained}}$	$\mathbf{0.998} \pm 0.004$	$\mathbf{1.000} \pm 0.000$	$\mathbf{1.000} \pm 0.000$	0.992 ± 0.004
legal random	0.972 ± 0.039	0.985 ± 0.007	0.998 ± 0.004	0.985 ± 0.014
π_{random}	0.915 ± 0.007	0.738 ± 0.004	0.923 ± 0.018	0.867 ± 0.018
π_{reversed}	0.913 ± 0.011	0.728 ± 0.039	0.907 ± 0.018	0.887 ± 0.039

A.3 Cross-Model (Llama-3.1-8B-Instruct) Accuracy by Context-Length Bucket

Table 6 gives the by-bucket breakdown for the GSM8K cross-model check of Section 5.5 (single run, 200 problems, seed 0 dataset).

Table 6: GSM8K accuracy by number of facts in context, Llama-3.1-8B-Instruct, all six order families.

Domain	n (count)	π_{natural}	π_{free}	$\pi_{\text{constrained}}$	π_{tiebreak}	π_{random}	π_{reversed}
GSM8K (8B)	10–19 (43)	0.977	0.907	0.977	0.860	0.488	0.465
GSM8K (8B)	20–29 (54)	0.981	0.870	0.963	0.870	0.463	0.426
GSM8K (8B)	30–39 (67)	0.910	0.836	0.940	0.791	0.433	0.358
GSM8K (8B)	40–45 (36)	0.889	0.722	0.972	0.778	0.361	0.361

A.4 Non-Linear Topologies: Full Order-Family Breakdown

Table 7 gives full six-family accuracy for the two topologies (chain, v_shape) with a reliable scoring criterion (Section 5.5).

Table 7: Full six-family accuracy for the two reliably-scored non-linear topologies, $n \in \{10, ..., 45\}$ facts, GSM8K, single seed, 200 problems per topology.

Topology	π_{natural}	π_{free}	$\pi_{\text{constrained}}$	π_{tiebreak}	π_{random}	π_{reversed}
Chain	0.575	0.460	**0.840**	0.490	0.260	0.115
V-shape	0.750	0.465	**0.850**	0.495	0.310	0.045

A.5 OptiSeq-Style Baseline: Selection Distribution

Table 8 gives the full distribution of which candidate order the OptiSeq-style selection rule of Section 5.5 actually picked, across the 200-problem run per domain (16 candidates per problem: the six order families plus 10 additional random precedence-legal orders, collapsed here into "other random-legal").

Table 8: OptiSeq-style selection: which candidate order was picked, out of 200 problems per domain.

Chosen order	GSM8K (n=200)	Narrative (n=200)
π_{natural}	45	16
$\pi_{\text{constrained}}$	37	20
$\pi_{\text{recovered}}$ (free)	35	11
π_{tiebreak}	12	29
π_{random}	3	16
π_{reversed}	3	9
other random-legal (10 candidates, pooled)	65	99

A.6 Tiebreak Ablation Detail

Table 9 reports, for each domain, the fraction of instances at which π_{tiebreak} (Section 4.1) was fully precedence-legal, alongside the accuracy of all four order families discussed in Section 5.2 on the same single-seed run. In neither domain was π_{tiebreak} ever fully legal, which is why its accuracy tracks π_{free} rather than $\pi_{\text{constrained}}$.

A.7 Accuracy by Context-Length Bucket

Table 10 gives the full per-bucket breakdown underlying Figure 3 and the summary numbers in Section 5.2, for all six order families in both domains.

Table 9: Fraction of instances at which π_{tiebreak} was fully precedence-legal, and accuracy of all four order families on the same 200-instance run per domain.

Domain	Fully legal	π_{natural}	π_{free}	π_{tiebreak}	$\pi_{\text{constrained}}$
GSM8K	0 / 200	0.530	0.495	0.505	0.865
Narrative	0 / 200	0.690	0.625	0.590	0.895

Table 10: Accuracy by number of facts in the context window (n), all six order families, both domains (single seed per domain; GSM8K seed matches Table 1's primary seed).

Domain	n (count)	π_{natural}	π_{free}	$\pi_{\text{constrained}}$	π_{tiebreak}	π_{random}	π_{reversed}
GSM8K	10–19 (41)	0.854	0.512	0.927	0.659	0.268	0.122
GSM8K	20–29 (69)	0.609	0.594	0.884	0.536	0.116	0.116
GSM8K	30–39 (62)	0.387	0.403	0.823	0.435	0.065	0.081
GSM8K	40–45 (28)	0.179	0.429	0.821	0.357	0.000	0.107
Narrative	10–19 (43)	0.814	0.605	0.953	0.674	0.442	0.302
Narrative	20–29 (54)	0.833	0.759	0.852	0.574	0.519	0.204
Narrative	30–39 (67)	0.642	0.597	0.896	0.522	0.373	0.149
Narrative	40–45 (36)	0.417	0.500	0.889	0.639	0.306	0.194

A.8 Position and Distance: Raw Condition Accuracies

Table 11 gives the raw per-condition accuracies underlying Figure 5 and Section 5.4: the $k = 5$ distance-controlled null (75 instances per gap value), and the two $k = 45$ experiments (75 instances per condition each) re-expressed in Section 5.4 on their shared root-fact-slot axis.

Table 11: Left: $k = 5$ distance-controlled null, accuracy by gap between the referenced pair. Right: the two $k = 45$ experiments, accuracy by condition as originally parameterized (position bucket with gap fixed at 1; gap with the final fact pinned at the last slot).

$k = 5$ distance null (gap, accuracy)	
Gap 1	0.880
Gap 2	0.893
Gap 3	0.920
Gap 4	0.867

$k = 45$ position exp. (bucket, accuracy)	
Start	0.947
Quarter	0.747
Mid	0.747
Three-quarter	0.720
End	0.933

$k = 45$ distance exp. (gap, accuracy)	
Gap 1	0.920
Gap 10	0.840
Gap 22	0.747
Gap 44	0.867

A.9 Solver Validation

To confirm the approximate solver of Section 4.4 is trustworthy before relying on it for the experiments, we validated it against exact brute-force search on 300 randomly generated small-n instances – small enough ($n \leq 8$) for full enumeration to serve as ground truth. The solver's returned order matched the brute-force-optimal cost exactly on 293 of 300 instances (97.7%); on the remaining 7, the returned cost exceeded the brute-force optimum by a mean of 16.2% (max 51.1%) rather than failing outright. Across all 300 instances, 0 returned orders violated precedence legality.

Because brute force is intractable past $n \approx 9$, we additionally checked the solver at $n \in [15, 20]$ (close to the scales used in Sections 5.2–5.4) against an exact constraint-programming (CP-SAT) formulation, given a 30-second-per-instance budget to reach a certified global optimum. Across 60 random instances, CP-SAT certified a global optimum on only 24 (a dense, NP-hard problem at this scale is not always solvable to certification in 30s); among those, the approximate solver matched

the certified optimum exactly on 20 (83.3%), with the remaining 4 showing a median relative cost gap of 15.1% (one outlier with a near-zero certified-optimal cost shows a 154.7% relative gap that is an absolute difference of only 2.0). We exclude the 36 uncertified instances from this comparison, since an uncertified CP-SAT solution is only a best-effort upper bound, not a proven optimum. This confirms the solver's output is a faithful stand-in for $\pi_{\text{constrained}}$ at the scales used in Sections 5.2–5.4, where exact search is no longer tractable. We evaluated it only on these instances and make no claim that it is a general-purpose ordering method.