Information Abundance Paradox: LONG-CONTEXT TRAINING UNDERMINES PARAMETRIC KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly trained and deployed with long contexts that span documents, code repositories, and interaction histories. This scaling reflects the implicit assumption that training on longer contexts will only help the model by exposing it to richer evidence. We challenge this view by studying how the context window shapes a model's *mode of learning*, shifting it between parametric *internalization* and *contextualization*. We propose the *Information Abundance Paradox*, which hypothesizes that abundant relevant information in the training context can reduce the incentive to encode that information *parametrically*, thereby increasing reliance on *context*. In pretraining with long documents, increasing the context window improves language modeling, natural language understanding, and closed-book MCQA only up to an intermediate optimum, after which performance consistently declines. In supervised fine-tuning, more task-relevant train-time context improves performance with supporting context, but reduces robustness when context is absent or misleading at test time. Our analysis suggests that this behavior arises when longer context provides a lower complexity solution. Mechanistically, training with informative context shifts gradient pressure from feed-forward networks, often linked to parametric knowledge, toward attention modules, and causal interventions show that this shift increases reliance on context during inference. Overall, these findings support the Information Abundance Paradox and suggest that scaling toward near-infinite context is not simply a matter of supplying more data, even when high-quality long-context data is abundant.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed in settings where useful evidence spans long documents (Bai et al., 2024), codebases (Liu et al., 2023c; Jimenez et al., 2024), and extended interaction histories (Zhou et al., 2024). In response, training practices have extended context windows to increasingly large scales (Peng et al., 2026; Chen et al., 2023), in some cases reaching millions of tokens (Team, 2024; Ding et al., 2024). Yet it remains unclear what, if anything, fundamentally limits this trajectory. This raises a basic question:

> *Are near-infinite context models simply a matter of more long-context data?*

The empirical trajectory of long-context scaling reflects an implicit assumption that sufficiently plentiful long-context data will provide the signal needed for continued progress. Therefore, when long-context training underperforms, prior work often attributes the bottleneck to the scarcity of naturally long, high-quality documents (Chen et al., 2025) and focuses on data-driven recipes as the path forward (Xiong et al., 2024; Fu et al., 2024; Gao et al., 2025b;a).

A motivating observation. This data-centric view overlooks the possibility that the context window is not a neutral conduit for data. Phi-3 (Abdin et al., 2024) and OLMo 3 (Olmo et al., 2026) models provide motivating examples, as their long-context (128K for Phi-3 and 65K for OLMo 3) variants consistently underperform their short-context (4K for Phi-3 and 8K for OLMo 3) counterparts in few-shot and zero-shot settings (Figure 1). Since the long-context variants are trained on *more* and *longer* data, this gap cannot be attributed to data quality alone (Abdin et al., 2024; Olmo et al., 2026). It instead raises a question: *how does long-context training shape the capabilities of the models?*

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.

Two modes of learning. Training a language model by minimizing next token cross-entropy can be viewed as a form of compression (Witten et al., 1987; Delétang et al., 2024), whereby reusable regularities in the training corpus are distilled into the model parameters (Grünwald & Roos, 2019).

Critically, such regularities enable two sources of predictive power: knowledge stored in the parameters (Petroni et al., 2019; Roberts et al., 2020), and information supplied in context (Lewis et al., 2021; Brown et al., 2020). This distinction suggests two *modes of learning* (Pan et al., 2023; Lin & Lee, 2024; Fang et al., 2025), in which a model either *internalizes* task-relevant information by encoding it in its parameters or *contextualizes* that information by learning to use evidence supplied in context. These two sources of predictive power offer alternative channels for prediction, with their relative use shaped by the training dynamics (Wang et al., 2023) and the informativeness of the context (Anand et al., 2025).

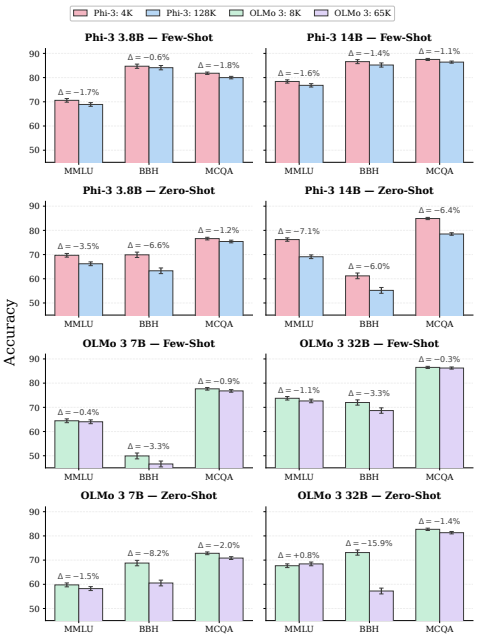


Figure 1: **Longer-context Phi-3 and OLMo 3 variants underperform, providing a motivating observation.** Across benchmarks, 128K and 65K variants lag behind 4K and 8K variants in few-shot and zero-shot evaluation settings (App. A). Error bars denote 95% CI.

Our hypothesis. Building on this view, we propose Information Abundance Paradox, which posits that information-rich long-context training can shift a model's learned strategy away from internalization and toward contextualization. As a result, when context is absent, missing, or insufficient at inference time, performance can degrade relative to models trained with less informative context, whether shorter or equally long but less relevant (§2).

Our evidence. We test Information Abundance Paradox in pretraining and supervised fine-tuning. In pretraining, longer context windows yield *inverted-U* performance, with language modeling, natural language understanding, and closed-book MCQA improving up to an intermediate optimum before performance degrades (§3.1). In supervised fine-tuning, task-relevant context improves accuracy under informative context, but reduces robustness when context is absent or misleading (§3.2). We then provide a theoretical account of these findings, showing that longer contexts can reduce the amount of task information that must be stored in the weights (§4). Lastly, our mechanistic analyses reveal that this phenomenon arises when longer context provides a lower complexity solution (§5.1) and shifts update pressure from feed-forward networks to attention modules (§5.2). Together, our findings show that continued scaling of the context window cannot be understood solely as a data problem: longer training contexts can fundamentally alter what models internalize and how strongly they depend on information supplied at inference time.

2 The Information Abundance Paradox

The train-time context window controls information available to a model during training. Increasing the context window is intended to expand the information a model can exploit, while ideally preserving short-context or context independent competence (Xiong et al., 2023; Ding et al., 2024).

We argue that this goal can face a competing effect: when task-relevant information is available in context during training, the model can reduce loss by using it directly rather than by encoding the same information in its weights. We call this hypothesis the *Information Abundance Paradox*:

> ***The Information Abundance Paradox.***
> When task-relevant information is made available through the training context, the model can reduce loss by using that information directly rather than by encoding it in its parameters. Consequently, this can shift the model's mode of learning away from parametric internalization and toward contextualization.

Here, *information abundance* refers to task-relevant information available within a context, and not the amount of training data or the entropy of the corpus. The central implication is that context length is not merely a data-delivery mechanism. It can also determine whether the model learns to rely on information stored parametrically or on information supplied through the context.

We refer to the behavioral manifestation of this shift in models as ***context addiction***, where a model trained with informative context performs *well when useful context is available* but deteriorates when that context is *absent* or *misleading*. Thus, context addiction provides an observable test of the *Information Abundance Paradox* through robustness under absent or misleading context.

3 Main Experiments: Testing the Information Abundance Paradox

Our hypothesis concerns *the amount of task-relevant information* available in the training context. We operationalize information abundance differently across the two training regimes. In pretraining, where relevance is difficult to control, we vary the length of coherent document spans as a proxy for information availability (§3.1). In supervised fine-tuning, where relevance can be controlled directly, we fix the context length and vary the amount of task-relevant information to isolate its role (§3.2).

3.1 Pretraining with Varying Context Length

We first test the Information Abundance Paradox in pretraining, where the context window controls how much within-document evidence is available during next token prediction. We *vary* the training context window while *fixing* the token budget, data, model configuration and optimization setup. We then evaluate how the training context window affects downstream task performance.

Model architecture. We pretrain language models at four scales, 20M, 55M, 259M, and 750M parameters. All models use the Llama-2 architecture and tokenizer (Touvron et al., 2023), RoPE (Su et al., 2023) for positional encoding, a standard causal attention mask, and the next token cross-entropy objective. We verify that our findings are robust to alternative positional encodings in App. B.

Data. We train on 10B tokens drawn from a subset of Project Gutenberg (Rae et al., 2019). We retain documents containing at least 65536 tokens, so that increasing the training context window exposes longer coherent within-document spans rather than merely increasing cross-document packing.

Training setup. For each model scale, we sweep the training context window over seven choices, $W \in \{512, 1024, \ldots, 32768\}$, in powers of two. For each W, the corpus is partitioned into non-overlapping sequences of exactly W tokens, with documents concatenated only as needed to fill complete sequences. Loss is computed over every token in the sequence. The global batch size is fixed at approximately 1.05M tokens, corresponding to 9537 optimization steps for every variant. Longer-context variants therefore use fewer sequences per batch. Within each model scale, all context window variants share the same model initialization, random seed, optimizer, and training hyperparameters. We train all models using the `nanotron` framework (Tazi et al., 2025), with full model configurations and hyperparameters provided in App. C.1. *Our comparison is therefore token-and-update-matched, isolating context window effects under a fixed token budget.*[1]

Evaluation. We evaluate the final checkpoint of each model in the zero-shot setting across three complementary testbeds: (i) a language modeling suite (e.g., LAMBADA (Paperno et al., 2016), WikiSPAN (Cheng et al., 2024), and Penn Treebank (Marcus et al., 1993)), measuring next token prediction over natural language text; (ii) SuperGLUE (Wang et al., 2020), measuring general language understanding; and (iii) a suite of closed-book MCQA benchmarks (e.g., ARC (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), PIQA (Bisk et al., 2019)), targeting parametric knowledge. Evaluation uses each benchmark as provided, with no context beyond the task input. We report cross-entropy loss on the language modeling suite and accuracy on SuperGLUE and MCQA tasks. For multiple-choice benchmarks, answers are selected by length-normalized log probability of the candidate answer text tokens, excluding the option label. We provide the complete benchmark list in App. D and motivate their inclusion in our evaluation.

[1]The comparison is not FLOP-matched, since longer windows incur higher attention cost as self-attention scales quadratically with sequence length (Vaswani et al., 2023).

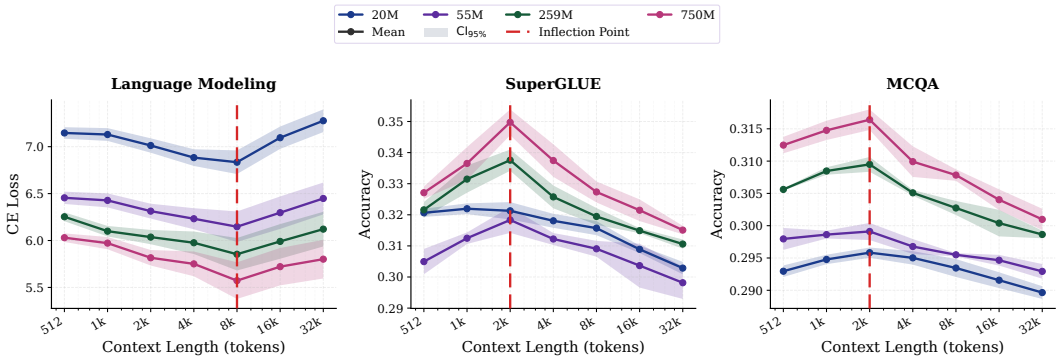


Figure 2: **Performance improves up to an intermediate optimum as pretraining context window grows.** Across model sizes, SuperGLUE and MCQA follow an inverted-U pattern, while language modeling shows a corresponding U-shaped loss curve. Dataset breakdowns are provided in App. E.

Results. As shown in Figure 2, performance on SuperGLUE and MCQA follows an *inverted-U* pattern as the pretraining context window grows, while language modeling loss follows a corresponding *U-shaped* pattern. Models initially benefit from longer contexts up to a testbed-dependent inflection point, as performance peaks around 2048 tokens for SuperGLUE and MCQA and around 8192 tokens for language modeling. Beyond these points, further increases in context length progressively erode the earlier gains. The presence of an inflection point across all three testbeds suggests that the effect reflects a systematic effect of pretraining context length, with statistically significant inflections confirmed across evaluation suites and model scales (App. F.2). Importantly, greater model capacity does not eliminate the long-context degradation, as the same qualitative pattern persists across all four model scales, from 20M to 750M parameters. One possible explanation for the earlier optimum on SuperGLUE and MCQA is that the optimal train-time context length depends partly on the length distribution of the evaluation instances, for which we provide supporting evidence in App. E. Taken together, these results show that, under matched token budgets, context length is not a free scaling axis, since longer training windows can degrade performance beyond an intermediate optimum.

3.2 Supervised Fine-Tuning with Varying Context Informativeness

We now test the Information Abundance Paradox in supervised fine-tuning for knowledge-rich tasks, where we *fix* the context budget and *vary* the task-relevant information in context. This setting isolates the role of informative train-time context while holding the task, model, and training process fixed.

Setup. We fine-tune `Qwen3 0.6B, 1.7B, 4B, 8B, 14B` (Yang et al., 2025) with LoRA (Hu et al., 2021) on four MMLU-Pro domains (Wang et al., 2024): Health, Economics, Law, and Psychology. Since MMLU-Pro provides only test splits, we partition the examples within each domain into 80% training and 20% evaluation sets. For each question, we construct a fixed context budget of $n = 8$ documents and vary the number of target-domain documents $k \in \{0, 4, 8\}$, where the remaining $8 - k$ documents are drawn from the other domains (App. G). Therefore, we only change the domain composition of the eight documents across k, controlling the fraction of *informative context* while holding the other aspects of fine-tuning fixed. We report document length statistics in App. G. We apply LoRA adapters to both attention modules and feed-forward networks, and compute the loss only on answer text tokens. Training details are provided in App. C.2.

Evaluation. We evaluate each model on held-out questions from the domain used for its fine-tuning, without in-context demonstrations. While training varies the number of target-domain documents, evaluation fixes documents to the target domain and varies whether they support the correct or an incorrect answer (see Table 14 in App. G). Accordingly, we conduct evaluation under three test-time context conditions: (i) *supporting context*, where the prepended documents support the correct answer; (ii) *conflicting context*, where the prepended documents support an incorrect answer; and (iii) *no context*, where the model receives only the question and answer choices. We use deterministic generation without sampling and report exact-match accuracy against the ground truth answer.

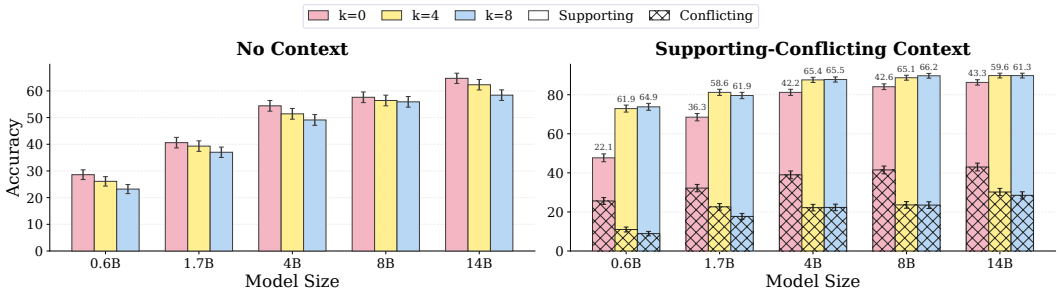


Figure 3: **Task-relevant train-time context induces context addiction.** Increasing target-domain documents from $k = 0$ to $k = 8$ improves Qwen3 models with supporting context, but reduces robustness without context (left) and with conflicting context (right). Error bars denote 95% CI.

Results. Figure 3 reports domain-averaged results, with domain-specific breakdowns deferred to App. H.1. The results reveal that train-time context affects model behavior primarily through its relevance to the task. Comparing fine-tuning runs with $k = 0, 4, 8$, we find that increasing the number of target-domain documents strengthens performance when supporting context is available at test time. However, this improvement comes with significantly reduced no-context accuracy and significantly greater vulnerability to conflicting context across model sizes and domains (App. F.3). This suggests that the observed shift is driven not by additional tokens alone, but by the task-relevant information they provide. Taken together, these results show that train-time context is not neutral background information, since its relevance can determine whether models internalize the task or contextualize it through the in-context evidence.

4 A Theoretical Account of Information Abundance Paradox

Our experiments (§3.2) show that increasing train-time context can improve performance when useful context is available at test time, while reducing robustness when that context is absent or misleading. We now formalize why this tradeoff is possible. The key idea is that context and weights can act as alternative carriers of task-relevant information. A longer context gives the predictor an additional channel through which task information can be accessed, and therefore can reduce the minimum amount of task information that must be stored in the weights.

Setup. Let $\tau \sim P_{\mathcal{T}}$ denote a discrete latent task variable indexing the data-generating distribution, treating the observed corpus as one realization from a broader population of all possible corpora.[2] For each context size k, let $(X^{(k)}, Y) \sim P_\tau^{(k)}$ denote an input-output pair with input $X^{(k)}$ of context size k. Let $W \sim \Pi(\cdot \mid \tau)$ denote the learned weights induced by training on data from task τ, with randomness due to data sampling and optimization. A language model with context size k defines a conditional predictor q_k, which induces predictions according to $\hat{Y} \sim q_k(\cdot \mid X^{(k)}, W)$. Let $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \to \mathbb{R}_+$ be a task loss (e.g., binary token error) between the target Y and prediction $\hat{Y}$. We define the risk $\mathcal{R}_k(\Pi, q_k) = \mathbb{E}[\ell(Y, \hat{Y})]$, where the expectation is over $(X^{(k)}, Y) \sim P_\tau^{(k)}$, $W \sim \Pi(\cdot \mid \tau)$, and $\hat{Y} \sim q_k(\cdot \mid X^{(k)}, W)$. Thus, $\mathcal{R}_k$ measures the expected task loss of the predictor given access to context of size k. We quantify task-specific information stored in the weights by $I(W; \tau)$ computed under the joint distribution induced by $P_{\mathcal{T}}$ and $\Pi(W \mid \tau)$. Equivalently, $I(W; \tau) = H(\tau) - H(\tau \mid W)$, so it measures the reduction in uncertainty about the task after observing the learned weights. While we vary the context size k, we hold fixed the latent task distribution, the underlying data-generating process, and the model architecture.

Definition 1 (Parametric information frontier)**.** *For risk threshold ρ, define*

$$\mathcal{I}_k(\rho) = \inf_{\Pi,\, q_k} I(W; \tau) \qquad \textit{such that} \qquad \mathcal{R}_k(\Pi, q_k) \le \rho.$$

Thus, $\mathcal{I}_k(\rho)$ is the minimum task information that must be stored in the weights to attain risk at most ρ when prediction has access to the input $X^{(k)}$ corresponding to context size k.

[2]For instance, our pretraining experiments use one filtered subset of Project Gutenberg, whereas many comparably sized subsets could have been sampled from the same underlying collection.

Without loss of generality, we assume the inputs are nested across context sizes through the packed token stream, where for each $k < m$, a shorter-context sequence is an aligned subwindow of a longer-context sequence. Accordingly, there exists a measurable projection map $T_{k,m}$ with $X^{(k)} = T_{k,m}(X^{(m)})$ almost surely. Here, $T_{k,m}$ selects the corresponding k-token block within the m-token window, matching standard pretraining pipelines that partition the same token stream into fixed-length windows at different context sizes (Tazi et al., 2025).

Proposition 4.1 (Monotonicity of the parametric information frontier)**.** *If $X^{(k)} = T_{k,m}(X^{(m)})$ almost surely for $k < m$, then*

$$\mathcal{I}_m(\rho) \leq \mathcal{I}_k(\rho) \qquad \textit{for all } \rho.$$

Proof sketch. A predictor with access to $X^{(m)}$ can simulate any k-context predictor q_k by first applying the projection map $T_{k,m}$ to recover $X^{(k)}$, and then using the resulting input in the k-context predictor. Therefore, every feasible weight-predictor pair for the optimization problem defining $\mathcal{I}_k(\rho)$ induces a feasible pair for the corresponding problem at context size m with the same weight channel. Thus, enlarging the context weakly expands the feasible set of the constrained optimization problem without increasing the task information stored in the weights. Taking the infimum over feasible solutions yields $\mathcal{I}_m(\rho) \leq \mathcal{I}_k(\rho)$. □

This is an *achievability* statement, showing that longer context *can* reduce the minimum amount of task information stored in the weights to achieve a target risk threshold.[3] If longer train-time context reduces the task information encoded in the weights, then removing or corrupting context leaves the predictor with less parametric knowledge to fall back on. This can cause performance to deteriorate when context is absent or misleading, yielding the behavioral signature of *context addiction* (§2).

5 Mechanisms Behind the Information Abundance Paradox

5.1 Solution Complexity

We conduct a controlled synthetic pretraining study to identify when longer context induces context addiction. By holding fixed the task, objective, and data-generating process while varying the number of in-context demonstrations, we test whether context addiction emerges selectively when additional demonstrations provide a lower complexity training trajectory.

Setup. We test this prediction in a synthetic pretraining setting with four tasks: (i) unary bitwise operations, (ii) string operations, (iii) mod10 arithmetic, and (iv) Caesar cipher. For each task, we train language models at three scales (0.3M, 1.5M, and 7.5M parameters), varying the number of in-context demonstrations while holding fixed the task, objective, and data-generating process. Figure 4 reports results for the largest model, with the full model-scale breakdown provided in App. H.2. Task definitions and training details are provided in App. C.3.

Evaluation. We evaluate each model under supporting and conflicting context, paralleling the setup used in §3.2. Test-time context contains the same number of in-context demonstrations as the model observed during training. In the supporting condition, in-context examples follow the correct task rule, while in the conflicting condition, they follow a consistent but incorrect rule. For example, in a bitwise-negation task, supporting examples map $\neg 0 \to 1$ and $\neg 1 \to 0$, whereas conflicting examples consistently map $\neg 0 \to 0$ and $\neg 1 \to 1$. *The supporting–conflicting gap therefore measures reliance on supplied context rather than on a parametrically internalized rule.*[4]

Results. Figure 4 (top row) shows that longer train-time context does not affect all tasks uniformly. For bitwise and string operations, the gap grows steadily with context length, indicating increasing dependence on the demonstrations as conflicting-context performance approaches the random baseline. In contrast, mod10 remains unchanged across context lengths, while Caesar cipher shows only a localized deviation at the longest context length.

[3] App. I gives the full proof.

[4] We omit no-context evaluation because models are trained only on final-query loss, making shorter test sequences out-of-distribution. See App. C.3 for details.

To distinguish these regimes, we use average training gradient norm as a comparative proxy for optimization path and learned function complexity, following connections between gradient magnitude, implicit regularization, and function complexity (Barrett & Dherin, 2022; Smith et al., 2021; Dherin et al., 2022). For context length k, with parameters $\theta_t^{(k)}$ at step t, we compute average gradient norm $G_k = \frac{1}{T}\sum_{t=1}^{T}\left\|\nabla_\theta \ell(\theta_t^{(k)}; B_t)\right\|_2$, where B_t is the training batch and ℓ is the final-query loss. At each step, we compute the global ℓ_2 norm before gradient clipping by concatenating the gradients of all trainable parameters.

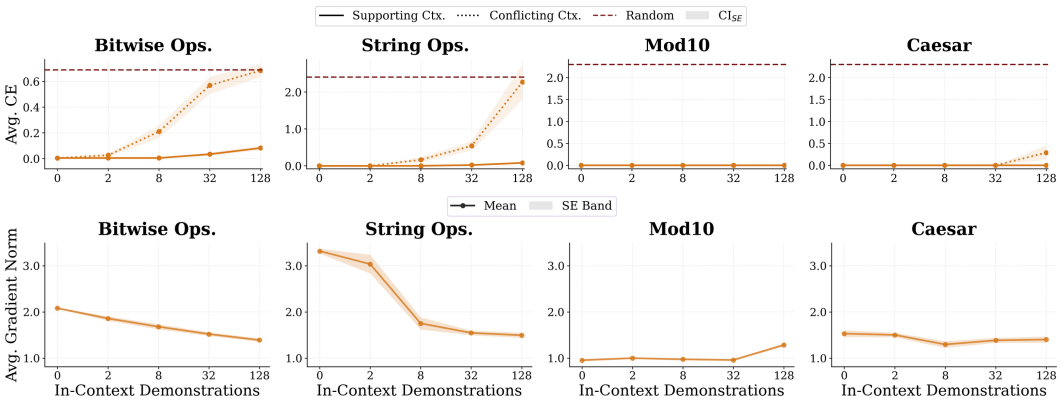


Figure 4: **Context addiction emerges when longer train-time context enables lower complexity solutions. Top row:** Supporting–conflicting gaps grow for bitwise and string tasks, but remain stable for mod10 arithmetic and Caesar cipher tasks, with each y-axis scaled to the task-specific random baseline. **Bottom row:** Average training gradient norms decrease only in the context-addicted tasks, consistent with longer context enabling simpler context-based solutions.

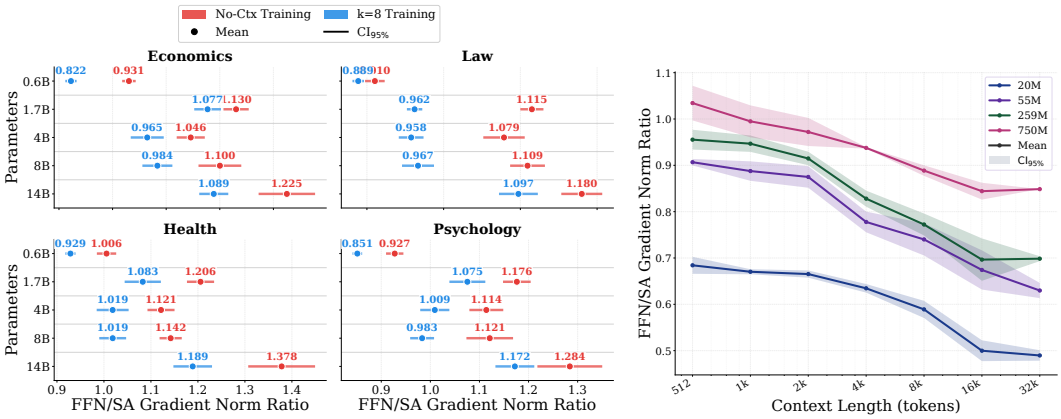
Figure 4 aligns the gradient norm proxy (bottom) with the behavioral results (top). Tasks with growing supporting–conflicting gaps also show decreasing average gradient norms as train-time context increases, whereas robust tasks show only noise-level changes or slight increases (see App. F.4 for significance tests). This pattern supports the interpretation that context addiction emerges when demonstrations provide an easier optimization path than parametric internalization of the task rule.

5.2 Module-Level Gradient Allocation

Prior work suggests that feed-forward networks (FFNs) and self-attention (SA) heads play distinct roles in transformer language models, associating FFNs with factual and task-relevant knowledge stored in parameters, and SA heads with the selection and routing of information from context (Geva et al., 2021; Dai et al., 2022; Meng et al., 2023). Motivated by this distinction and prior work that uses relative gradient magnitudes to characterize transformer optimization dynamics (Zhang et al., 2019; Liu et al., 2023a; Noci et al., 2022), we analyze module-level gradient allocation using the FFN-to-SA gradient ratio to measure whether training updates concentrate more strongly in FFNs or SA heads as context length increases.

Module-wise gradient ratios. For pretraining, we follow the setup in §3.1 and vary the context window size. For SFT, we follow the setup in §3.2 and compare training with $k = 8$ task-relevant documents against a no-context baseline that fine-tunes only on question-answer pairs. For each training step, we first average the gradient norm within each module type across layers, compute the FFN-to-SA ratio, and then average this ratio across training steps. Because the architecture and parameterization are fixed, the FFN and SA gradients have fixed dimensionality, and, therefore, changes in the ratio reflect changes in relative gradient magnitude. Accordingly, *it tracks the relative optimization pressure between FFN-mediated internalization and SA-mediated contextualization*.

Figure 5a first shows this shift in supervised fine-tuning. When models are trained with $k = 8$ target-domain documents, the FFN-to-SA gradient norm ratio is lower than in the no-context baseline across model sizes and domains. This indicates that adding task-relevant train-time context shifts relative gradient pressure away from FFNs and toward self-attention. Figure 5b shows the analogous



(a) **SFT with task-relevant context shifts gradient pressure toward attention.** With $k = 8$ target-domain documents, the FFN-to-SA gradient norm ratio is consistently lower than in no-context tuning across model sizes and domains.

(b) **Pretraining with longer context shifts gradient pressure toward attention.** The FFN-to-SA gradient norm ratio declines with context length.

Figure 5: Informative train-time context lowers the FFN-to-SA gradient norm ratio in both SFT and pretraining, consistent with increased contextualization.

pattern in pretraining. As the training context window increases, the FFN-to-SA gradient norm ratio decreases across model scales, with the largest reductions appearing at longer context windows. Thus, both the controlled SFT setting and the pretraining setting point to the same qualitative change in module-level optimization dynamics. Taken together, these results show that context-rich training shifts gradient pressure from FFN-mediated internalization toward SA-mediated contextualization, supporting the view that longer train-time context changes the model's mode of learning. These shifts are statistically significant in 19 out of 20 SFT model-domain comparisons and at every pretraining model scale (App. F.5).

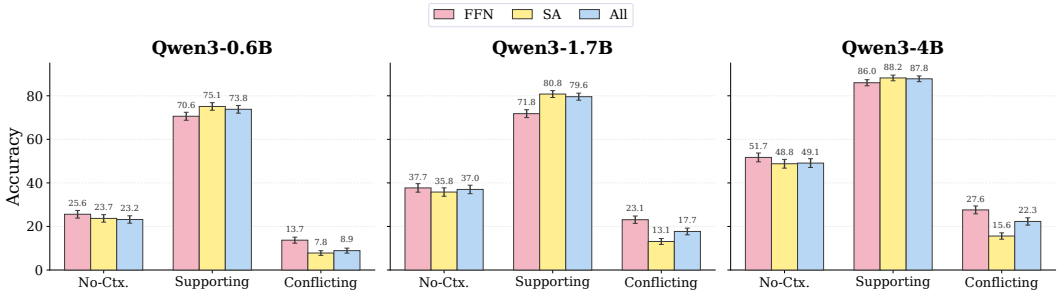


Figure 6: **FFN and SA updates causally control context reliance.** FFN-only tuning improves no-context robustness, while SA-only tuning strengthens supporting-context performance but increases sensitivity to conflicting context. Error bars denote 95% confidence intervals.

Module-restricted fine-tuning. We further test this interpretation with module-restricted fine-tuning, in which models are trained with $k = 8$ task-relevant documents while updating only FFN or only SA heads. To reduce compute, we perform these interventions on smaller Qwen3 models.

Figure 6 shows that FFN-only tuning yields stronger no-context performance and smaller degradation under conflicting context. In contrast, SA-only fine-tuning achieves significantly stronger performance with supporting context and significantly greater vulnerability to conflicting context across all three model sizes (App. F.6). This module-restricted intervention provides causal evidence for the gradient analysis, showing that FFN-directed updates improve parametric robustness, whereas SA-directed updates increase reliance on the supplied context. We further analyze whether this optimization shift is reflected at inference time through increased attention to context tokens in App. H.3.

6 Related Work

Long-context training. Recent work has made long-context language modeling increasingly practical by extending usable context through positional extrapolation and RoPE scaling (Press et al., 2022; Chen et al., 2023; Peng et al., 2026; Ding et al., 2024), sparse or recurrent attention (Dai et al., 2019; Beltagy et al., 2020; Child et al., 2019), and retrieval or memory augmentation (Borgeaud et al., 2022; Wu et al., 2022; Munkhdalai et al., 2024). Other work adapts pretrained models to longer windows using continued pretraining and long-document upsampling (Xiong et al., 2024), domain and length balanced data mixtures (Fu et al., 2024), and instruction tuning to shape long-context use (Gao et al., 2025b; Zhao et al., 2024). These methods primarily ask how models can acquire and use long-context capabilities while preserving existing performance. Short-to-long curricula recover long-context ability more efficiently than training at maximum length throughout (Jin et al., 2023; Pouransari et al., 2025; Zhu et al., 2025), suggesting that the training window may affect not only the attainable context length, but also the training dynamics used to acquire it. Relatedly, prior work explains non-monotonic context scaling as a tradeoff between the predictive value of additional context and the difficulty of learning to use it with limited data and model capacity (Shi et al., 2026). This account is complementary to ours, which focuses on how informative training context shifts learning from parametric knowledge toward context reliance.

Beyond effects on training efficiency and approximation difficulty, long-context adaptation can degrade short-context performance through representation drift and catastrophic forgetting (Dong et al., 2025), suggesting that naive context extension may trade one capability for another. A longer nominal window also does not by itself imply effective use of the added context, since nominal context length can overstate effective context length when long-range relative positions are undertrained (An et al., 2024). Relatedly, models systematically underuse information in the middle of long prompts (Liu et al., 2023b), and long-context benchmarks reveal persistent failures in retrieval, aggregation, and reasoning over extended inputs (Bai et al., 2024; Hsieh et al., 2024; Zhang et al., 2024b; Bianchi et al., 2025; Byerly & Khashabi, 2026). Prior work thus primarily asks how to obtain, preserve, or evaluate long-context capability, but does not characterize how the training context window shapes the model's mode of learning. We address this gap by studying whether the training context window governs the allocation of task information between parameters and context, thereby shifting models from parametric internalization toward context-dependent computation.

We discuss related work on *two modes of learning* and *language modeling as compression* in App. J.

7 Discussion and Conclusion

Implications. The Information Abundance Paradox challenges the assumption that scaling toward near-infinite context requires only more data. Our findings instead show that the context window can govern the model's mode of learning, with longer windows shifting models toward context-supplied information and away from reusable parametric knowledge. Whether this shift is beneficial depends on the application.

Limitations. Our pretraining experiments are limited up to 750M parameters in pretraining. Testing the same phenomenon at larger scale remains an important next step for determining how the location and severity of the observed inflection points change with model size, data scale, and compute.

Conclusion. Long-context processing is a powerful capability, but it is not a neutral scaling axis for language models. Our findings show that increasing train-time context can change the model's mode of learning, shifting it from parametric internalization toward contextualization. The takeaway is therefore not that long-context processing is undesirable, but that it should be understood through its role in mediating the tradeoff between context use and context independent competence.

AI use statement

We used generative AI tools to assist with polishing the writing, implementation and debugging, and early-stage discussions on how to formulate and approach the theoretical account presented in this work. All AI-assisted outputs were reviewed and revised by the authors as appropriate. The authors

take full responsibility for the content of the paper, including all text, code, analyses, and claims produced with the assistance of generative AI.

Ethics statement

We do not believe that our work raises specific ethical concerns requiring additional discussion under the ICLR Code of Ethics.

Reproducibility statement

We provide detailed descriptions of the experimental setups, model architectures, training procedures, hyperparameters, evaluation protocols, and statistical analyses in the main paper and appendices. Upon acceptance, we will publicly release the code, data used to construct our experiments, and trained model checkpoints necessary to reproduce the results presented in this work.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. Why does the effective context length of llms fall short?, 2024. URL https://arxiv.org/abs/2410.18745.

Suraj Anand, Michael A. Lepori, Jack Merullo, and Ellie Pavlick. Dual process learning: Controlling use of in-context vs. in-weights strategies with weight forgetting, 2025. URL https://arxiv.org/abs/2406.00053.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

David G. T. Barrett and Benoit Dherin. Implicit gradient regularization, 2022. URL https://arxiv.org/abs/2009.11162.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12119–12149, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.605. URL https://aclanthology.org/2025.naacl-long.605/.

Owen Bianchi, Mathew J. Koretsky, Maya Willey, Chelsea X. Alvarado, Tanay Nayak, Adi Asija, Nicole Kuznetsov, Mike A. Nalls, Faraz Faghri, and Daniel Khashabi. Hidden in the haystack: Smaller needles are more difficult for llms to find. *arXiv preprint arXiv:2505.18148*, abs/2505.18148, 2025. URL https://arxiv.org/abs/2505.18148.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Léonard Blier and Yann Ollivier. The description length of deep learning models, 2018. URL https://arxiv.org/abs/1802.07044.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022. URL https://arxiv.org/abs/2112.04426.

Jörg Bornschein, Yazhe Li, and Marcus Hutter. Sequential learning of neural networks for prequential mdl. *ArXiv*, abs/2210.07931, 2022. URL https://api.semanticscholar.org/CorpusID:252907410.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Adam Byerly and Daniel Khashabi. Self-consistency falls short! the adverse effects of positional bias on long-context problems. 2026. URL https://arxiv.org/abs/2411.01101.

Stephanie C. Y. Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K. Lampinen, and Felix Hill. Transformers generalize differently from information stored in context vs in weights, 2022. URL https://arxiv.org/abs/2210.05675.

Jianghao Chen, Junhong Wu, Yangyifan Xu, and Jiajun Zhang. Ladm: Long-context training data selection with attention-based dependency measurement for llms, 2025. URL https://arxiv.org/abs/2503.02502.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023. URL https://arxiv.org/abs/2306.15595.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. Dated data: Tracing knowledge cutoffs in large language models, 2024. URL https://arxiv.org/abs/2403.12958.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019. URL https://arxiv.org/abs/1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate, 2021. URL https://arxiv.org/abs/2006.16362.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc (eds.), *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers, 2022. URL https://arxiv.org/abs/2104.08696.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019. URL https://arxiv.org/abs/1901.02860.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression, 2024. URL https://arxiv.org/abs/2309.10668.

Benoit Dherin, Michael Munn, Mihaela Rosca, and David G. T. Barrett. Why neural networks find simple solutions: the many regularizers of geometric complexity, 2022. URL https://arxiv.org/abs/2209.13083.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens, 2024. URL https://arxiv.org/abs/2402.13753.

Zican Dong, Junyi Li, Jinhao Jiang, Mingyu Xu, Wayne Xin Zhao, Bingning Wang, and Weipeng Chen. Longred: Mitigating short-text degradation of long-context large language models via restoration distillation, 2025. URL https://arxiv.org/abs/2502.07365.

Eric Elmoznino, Tom Marty, Tejas Kasetty, Leo Gagnon, Sarthak Mittal, Mahan Fathi, Dhanya Sridhar, and Guillaume Lajoie. In-context learning and occam's razor, 2025. URL https://arxiv.org/abs/2410.14086.

Zhouxiang Fang, Aayush Mishra, Muhan Gao, Anqi Liu, and Daniel Khashabi. ICL CIPHERS: Quantifying "learning" in in-context learning via substitution ciphers. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25912–25933, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1316. URL https://aclanthology.org/2025.emnlp-main.1316/.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context, 2024. URL https://arxiv.org/abs/2402.10171.

Chaochen Gao, Xing Wu, Zijia Lin, Debing Zhang, and Songlin Hu. Nextlong: Toward effective long-context training without long documents, 2025a. URL https://arxiv.org/abs/2501.12766.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively), 2025b. URL https://arxiv.org/abs/2410.02660.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories, 2021. URL https://arxiv.org/abs/2012.14913.

Google DeepMind. Gemini 3.1 flash-lite. Model card, Google DeepMind, March 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/.

Peter Grünwald and Teemu Roos. Minimum description length revisited. *International Journal of Mathematics for Industry*, 11(01), December 2019. ISSN 2661-3344. doi: 10.1142/s2661335219300018. URL http://dx.doi.org/10.1142/S2661335219300018.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Annual Conference Computational Learning Theory*, 1993. URL https://api.semanticscholar.org/CorpusID:9346534.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly, 2024. URL https://arxiv.org/abs/2404.09937.

Marcus Hutter. The hutter prize. http://prize.hutter1.net, 2006. Accessed: 2026-04-30.

Nanjiang Jiang and Marie-Catherine de Marneffe. Evaluating BERT for natural language inference: A case study on the CommitmentBank. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *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. 6086–6091, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1630. URL https://aclanthology.org/D19-1630/.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020. URL https://arxiv.org/abs/2009.13081.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Chia-Yuan Chang, and Xia Hu. Growlength: Accelerating llms pretraining by progressively growing training length, 2023. URL https://arxiv.org/abs/2310.00576.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models, 2024. URL https://arxiv.org/abs/2402.18154.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, 2018.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

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, 2021. URL https://arxiv.org/abs/2005.11401.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning, 2024. URL https://arxiv.org/abs/2402.18819.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers, 2023a. URL https://arxiv.org/abs/2004.08249.

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, 2023b. URL https://arxiv.org/abs/2307.03172.

Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems, 2023c. URL https://arxiv.org/abs/2306.03091.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Matthew V. Mahoney. Text compression as a test for artificial intelligence. In *AAAI/IAAI*, 1999. URL https://api.semanticscholar.org/CorpusID:1023392.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018. URL https://arxiv.org/abs/1809.02789.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. URL https://arxiv.org/abs/2202.12837.

Fazal Mittu, Yihuan Bu, Akshat Gupta, Ashok Devireddy, Alp Eren Ozdarendeli, Anant Singh, and Gopala Anumanchipalli. Finezip : Pushing the limits of large language models for practical lossless text compression, 2024. URL https://arxiv.org/abs/2409.17141.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with infini-attention, 2024. URL https://arxiv.org/abs/2404.07143.

Aliakbar Nafar, Kristen Brent Venable, and Parisa Kordjamshidi. Learning vs retrieval: The role of in-context examples in regression with large language models, 2025. URL https://arxiv.org/abs/2409.04318.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse, 2022. URL https://arxiv.org/abs/2206.03126.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning, 2023. URL https://arxiv.org/abs/2305.09731.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context, 2016. URL https://arxiv.org/abs/1606.06031.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2026. URL https://arxiv.org/abs/2309.00071.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019. URL https://arxiv.org/abs/1909.01066.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations, 2019. URL https://arxiv.org/abs/1808.09121.

Hadi Pouransari, Chun-Liang Li, Jen-Hao Rick Chang, Pavan Kumar Anasosalu Vasu, Cem Koc, Vaishaal Shankar, and Oncel Tuzel. Dataset decomposition: Faster llm training with variable sequence length curriculum, 2025. URL https://arxiv.org/abs/2405.13226.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022. URL https://arxiv.org/abs/2108.12409.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1911.05507.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression, 2023. URL https://arxiv.org/abs/2306.15063.

Jorma Rissanen. Modeling by shortest data description*. *Autom.*, 14:465–471, 1978. URL https://api.semanticscholar.org/CorpusID:30140639.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model?, 2020. URL https://arxiv.org/abs/2002.08910.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pp. 90–95, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019a. URL https://arxiv.org/abs/1907.10641.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019b. URL https://arxiv.org/abs/1907.10641.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019. URL https://arxiv.org/abs/1904.09728.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948. URL http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Jingzhe Shi, Qinwei Ma, Hongyi Liu, Hang Zhao, Jeng-Neng Hwang, and Lei Li. Intrinsic entropy of context length scaling in llms, 2026. URL https://arxiv.org/abs/2502.01481.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent, 2021. URL https://arxiv.org/abs/2101.12176.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL https://arxiv.org/abs/1811.00937.

Nouamane Tazi, Ferdinand Mom, Haojun Zhao, Phuc Nguyen, Mohamed Mekkouri, Leandro Werra, and Thomas Wolf. The ultra-scale playbook: Training llms on gpu clusters, 2025.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline, 2019. URL https://arxiv.org/abs/1905.05950.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee,

Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Chandra Shekhara Kaushik Valmeekam, Krishna Narayanan, Dileep Kalathil, Jean-Francois Chamberland, and Srinivas Shakkottai. Llmzip: Lossless text compression using large language models, 2023. URL https://arxiv.org/abs/2306.04050.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length, 2020. URL https://arxiv.org/abs/2003.12298.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580/.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems, 2020. URL https://arxiv.org/abs/1905.00537.

Han Wang, Erfan Miahi, Martha White, Marlos C. Machado, Zaheer Abbas, Raksha Kumaraswamy, Vincent Liu, and Adam White. Investigating the properties of neural network representations in reinforcement learning, 2023. URL https://arxiv.org/abs/2203.15955.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Commun. ACM*, 30(6):520–540, June 1987. ISSN 0001-0782. doi: 10.1145/214762.214771. URL https://doi.org/10.1145/214762.214771.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality, 2024. URL https://arxiv.org/abs/2404.15574.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers, 2022. URL https://arxiv.org/abs/2203.08913.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. Effective long-context scaling of foundation models, 2023. URL https://arxiv.org/abs/2309.16039.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. Effective long-context scaling of foundation models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.260. URL https://aclanthology.org/2024.naacl-long.260/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. Entropy law: The story behind data compression and llm performance, 2024. URL https://arxiv.org/abs/2407.06645.

Bin Yu. The minimum description length principle in coding and modeling. *IEEE transactions on information theory*, 44(6):2743–2760, 1998.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Biao Zhang, Ivan Titov, and Rico Sennrich. Improving deep transformer with depth-scaled initialization and merged attention, 2019. URL https://arxiv.org/abs/1908.11365.

Junxuan Zhang, Zhengxue Cheng, Yan Zhao, Shihao Wang, Dajiang Zhou, Guo Lu, and Li Song. L3tc: Leveraging rwkv for learned lossless low-complexity text compression, 2024a. URL https://arxiv.org/abs/2412.16642.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension, 2018. URL https://arxiv.org/abs/1810.12885.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens, 2024b. URL https://arxiv.org/abs/2402.13718.

Liang Zhao, Tianwen Wei, Liang Zeng, Cheng Cheng, Liu Yang, Peng Cheng, Lijie Wang, Chenxia Li, Xuejie Wu, Bo Zhu, Yimeng Gan, Rui Hu, Shuicheng Yan, Han Fang, and Yahui Zhou. Longskywork: A training recipe for efficiently extending context length in large language models, 2024. URL https://arxiv.org/abs/2406.00605.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

Tongyao Zhu, Qian Liu, Haonan Wang, Shiqi Chen, Xiangming Gu, Tianyu Pang, and Min-Yen Kan. Skyladder: Better and faster pretraining via context window scheduling, 2025. URL https://arxiv.org/abs/2503.15450.

A EVALUATION OF PHI-3 AND OLMO 3 MODELS

We evaluate long and short-context variants from two model families, Phi-3 (Abdin et al., 2024) and OLMo 3 (Olmo et al., 2026). For Phi-3, we consider four instruction-tuned variants spanning two model scales, mini (3.8B parameters) and medium (14B parameters). Specifically, we evaluate `Phi-3-mini-4k-instruct`, `Phi-3-mini-128k-instruct`, `Phi-3-medium-4k-instruct`, and `Phi-3-medium-128k-instruct`. The short-context 4K variants correspond to the instruction-tuned checkpoints, whereas the long-context 128K variants are obtained through LongRoPE-based context extension with additional post-training (Abdin et al., 2024). For OLMo 3, we consider four base-model variants spanning two model scales, 7B and 32B parameters. Specifically, we evaluate the 8K and 65K variants for each scale. The short-context 8K variants correspond to checkpoints from the end of Stage 2 pretraining, whereas the long-context 65K variants correspond to checkpoints from the end of Stage 3 pretraining (Olmo et al., 2026). We evaluate these models on MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2022), and the suite of MCQA benchmarks described in App. D.3. We report accuracy as the evaluation metric for all tasks, along with the binomial standard errors.

We use a generation-based evaluation protocol for both few-shot and zero-shot settings using vLLM (Kwon et al., 2023). The model is prompted to generate an answer, and we extract the predicted option letter from the generated output. For few-shot evaluation, we follow the Phi-3 paper (Abdin et al., 2024) and use the same task-specific number of demonstrations: MMLU 5, BBH 3, ARC-Easy 10, ARC-Challenge 10, CommonsenseQA 10, BoolQ 0, OpenBookQA 10, PIQA 5, SocialIQA 5, SciQ 5, HellaSwag 5, MedQA 2, TruthfulQA 10, and WinoGrande 5. For zero-shot evaluation, we set $k = 0$ for all tasks. All evaluations use deterministic decoding.

We use the following multiple-choice prompt template:

```
Question: <question>
Options:
A. <choice A>
B. <choice B>
...
Answer:
```

B Positional Encoding Ablations

Our main pretraining experiments (§3.1) use RoPE (Su et al., 2023) for positional encoding. Because the choice of positional encoding may influence performance across context lengths, we test whether the observed trends depend on this architectural choice. To this end, we compare RoPE with ALiBi (Press et al., 2022) and LongRoPE (Ding et al., 2024) using the 259M parameter model under the pretraining setting of §3.1. For ALiBi, we replace RoPE with distance dependent attention biases while retaining the same training setup. For LongRoPE, we continue pretraining the 512 token RoPE checkpoints for 2.5B tokens at each extended context length using position dependent rescaling.

As shown in Figure 7, all three positional encoding schemes exhibit the same qualitative trend. Language modeling performance improves up to an intermediate context length and subsequently degrades, while SuperGLUE and MCQA peak at shorter context lengths before declining. ALiBi yields somewhat weaker absolute performance than RoPE, whereas LongRoPE closely follows the RoPE results. Taken together, these experiments show that the observed context length trend is robust to the choice of positional encoding and is not an artifact of RoPE.

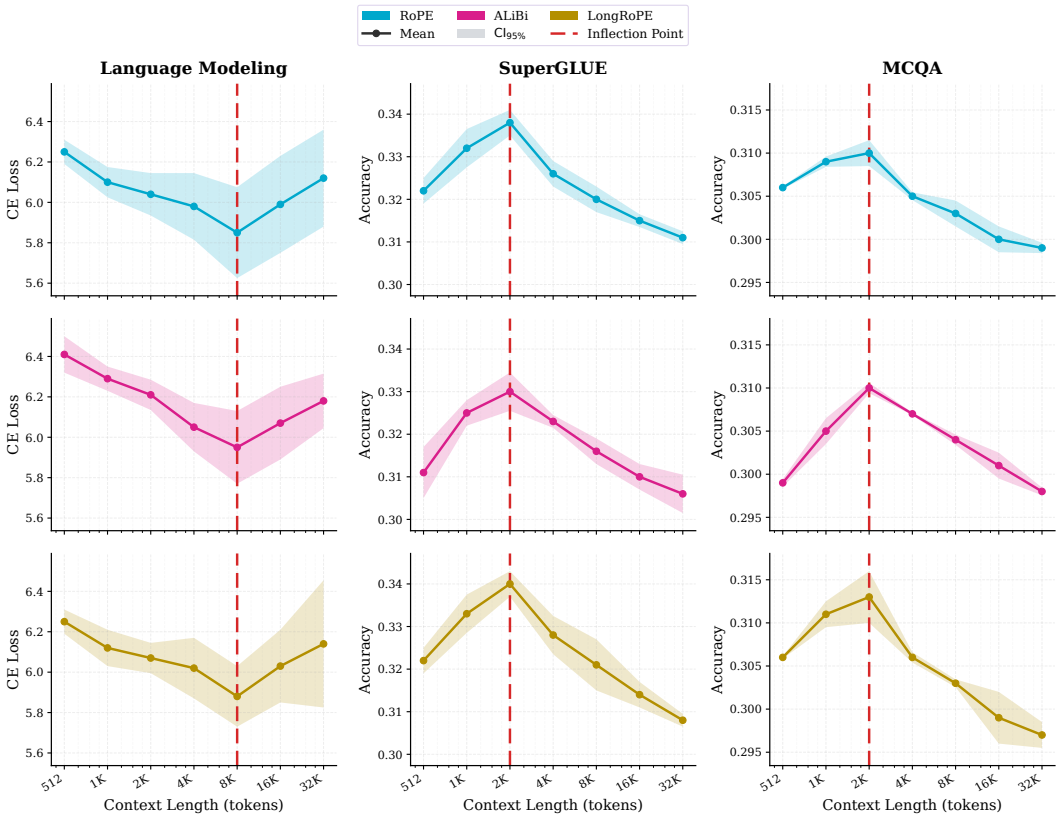


Figure 7: **The context length trend is robust to positional encoding choice.** Results for RoPE, ALiBi, and LongRoPE across language modeling, SuperGLUE, and closed-book MCQA. Each positional encoding exhibits the same qualitative pattern, where performance improves up to an intermediate context length and degrades as the training window increases further.

C Training Details

C.1 Natural Language Pretraining

This section provides training details for the natural language pretraining experiments in §3.1. All models follow a Llama-2 architecture (Touvron et al., 2023) with SwiGLU activations (Shazeer, 2020), RoPE positional embeddings (Su et al., 2023), RMSNorm (Zhang & Sennrich, 2019), and a shared vocabulary of 32000 tokens (Llama-2 tokenizer). The 259M and 750M models use grouped-query attention (Ainslie et al., 2023), while the 20M and 55M models use multi-head attention (Cordonnier et al., 2021). All models have untied input and output embeddings.

All models are trained for 9537 optimization steps on 10B tokens (4 epochs of a 2.5B-token corpus), with approximately 1.05M tokens per step. Optimization uses AdamW (Loshchilov & Hutter, 2019) with gradient clipping at 1.0, a linear warmup of 2000 steps, and cosine decay thereafter. All models are trained in `bfloat16` with FlashAttention-2 (Dao, 2023). The 20M, 55M, and 259M models use data parallelism across two NVIDIA A100 80GB GPUs, whereas the 750M models are trained on four NVIDIA H100 80GB GPUs. Remaining hyperparameters are provided in Table 1. For each configuration, we run multiple random seeds. We use five seeds for the 20M model and three seeds for the 55M, 259M, and 750M models. The seeds shared across all model scales are 42, 2026, and 1000. For the 20M model, we additionally use seeds 9999 and 12151. All reported evaluation results are averaged across seeds. Shaded standard error bands denote the standard error computed across seed-level means.

Table 1: Hyperparameters per model scale.

	20M	**55M**	**259M**	**750M**
Architecture				
Hidden size	256	512	1024	1536
Intermediate size	896	1792	3072	4608
Layers	4	6	16	24
Attn. heads (Q)	4	8	16	24
Attn. heads (KV)	4	8	4	6
Tied embeddings	×	×	×	×
Optimization				
β_1, β_2				0.9, 0.95
ϵ				10^{-8}
Weight decay				0.1
Peak LR	6×10^{-4}	4×10^{-4}	4×10^{-4}	3×10^{-4}
Min LR	6×10^{-5}	4×10^{-5}	4×10^{-5}	3×10^{-5}

Table 2 reports the average pretraining cost in GPU hours for each model scale and context length. Training cost is relatively stable at short context lengths but grows noticeably at longer contexts, particularly beyond 8K tokens. For instance, the 259M model requires 27.90 GPU hours at 512 tokens and 83.44 GPU hours at 32K tokens. Other models show the same qualitative trend. Because the 20M, 55M, and 259M models are trained on A100 GPUs whereas the 750M model is trained on H100 GPUs, absolute GPU hours should not be compared directly across model scales. These measurements make explicit the compute tradeoff associated with extending the context length during pretraining.

Table 2: Average pretraining cost in GPU hours across model sizes and context lengths.

	Context Length (tokens)						
Model	512	1K	2K	4K	8K	16K	32K
20M	4.77	4.74	4.91	5.12	5.52	6.40	8.17
55M	8.23	8.34	8.49	9.19	10.53	13.22	18.54
259M	27.90	28.66	30.13	33.57	40.72	55.04	83.44
750M	35.87	37.26	35.81	40.85	53.72	73.36	118.13

C.2 Supervised Fine-Tuning

This section provides training details for the SFT experiments in §3.2. We perform supervised fine-tuning using the `verl` infrastructure (Sheng et al., 2024). For each model and domain, we fine-tune for 5 epochs with a learning rate of 1×10^{-4}, cosine learning-rate decay, and gradient clipping at 1.0. We use a maximum sequence length of 1024 tokens and apply right truncation to examples exceeding this length.

We use LoRA with rank $r = 64$ and scaling parameter $\alpha = 128$. Training is performed in `bfloat16` precision with gradient checkpointing.

C.3 Synthetic Pretraining

This section provides training details for the synthetic pretraining experiments in §5.1.

Task format. Each example consists of a sequence of input-output demonstrations followed by a final query. Models are trained to predict only the output tokens of the final query. For a training context length k, the prompt contains k demonstrations sampled from the same task family, followed by one held-out query from that task.

A generic prompt has the form $(x_1, y_1), \ldots, (x_k, y_k), x_\star \mapsto y_\star$, where (x_i, y_i) are in-context demonstrations and $(x_\star, y_\star)$ is the final query. The supervised loss is computed only on $y_\star$. This setup makes the context useful for solving the final query while allowing us to vary the amount of useful context independently of the task family.

Task families. We use four deterministic task families.

1. **Unary bitwise operations.** Inputs are 16-bit binary strings. Each task applies a unary bitwise operation to the input string, such as NOT, which maps each bit to its complement.[5] The valid output tokens are `0` and `1`. We train each model for 25 epochs.
2. **String transformations.** Inputs are 8-letter strings over a fixed alphabet. Each task applies a deterministic string transformation, such as REVERSE.[6] The output is another string over the same alphabet. We train each model for 25 epochs.
3. **Digit-wise mod10 arithmetic.** Inputs are 5-digit strings. Each task applies a digit-wise arithmetic operation modulo 10. For example, a task may add a fixed digit-wise offset to each input digit, with all arithmetic performed modulo 10. We train each model for 10 epochs.
4. **Caesar cipher.** Inputs are 5-digit strings. Each task shifts every digit by a global offset modulo 10. Importantly, the same offset is applied to all positions. We train each model for 10 epochs.

Data generation. For each task family, datasets are generated deterministically from the corresponding input-output rule. We generate approximately 33K training examples and 10K test examples per task. Training and test inputs are disjoint. Each task uses a task-specific tokenizer with fewer than 30 tokens, including symbols for digits or letters, operation delimiters, separators, and special tokens. For each context length k, we construct examples by sampling k demonstrations and one final query from the same task. The final query does not appear among the demonstrations. The training set size is held fixed across context lengths so that changes in performance reflect the amount of useful context available per example rather than the number of optimization examples.

Table 3: Synthetic pretraining model configurations and hyperparameters.

	0.3M	**1.5M**	**7.5M**
Architecture			
Layers	4	6	8
Hidden size	64	128	256
Attn. heads (Q)	2	4	8
Attn. heads (KV)	1	2	4
Intermediate size	256	512	1024
Optimization			
Optimizer		AdamW	
Peak LR	1.0×10^{-4}	7.0×10^{-5}	5.0×10^{-5}
Min LR		1.0×10^{-5}	
Batch size	32	64	128
Training steps	25,000	12,500	6,250
Warmup steps	1,250	625	312
Weight decay		0.001	
Gradient clipping		1.0	

Architecture and optimization. We train decoder-only transformer language models at three scales: 0.3M, 1.5M, and 7.5M parameters. All models are trained from scratch on each task family separately. The architecture follows the same causal language modeling setup as the natural language pretraining experiments in §3.1, but uses smaller widths, depths, and task-specific vocabularies. All models are trained with a causal attention mask and next token prediction objective, but the loss is applied only to the final-query output tokens. We vary the number of in-context demonstrations available during training and hold the task family, data-generating process, optimizer, and model scale fixed.

Evaluation conditions. We evaluate each trained model under two context conditions.

1. **Supporting context.** The context contains demonstrations generated by the correct task rule. This condition measures performance when the model can rely on useful in-context evidence.

[5]The full operation list is IDENTITY, NOT, REVERSE, REVERSE_NOT, POPCOUNT, PARITY, LEADING_ZEROS, TRAILING_ZEROS, MSB, LSB, UPPER_HALF, and LOWER_HALF.

[6]The full operation list is IDENTITY, REVERSE, ROTATE_LEFT_1, ROTATE_RIGHT_1, ROTATE_LEFT_2, ROTATE_RIGHT_2, SWAP_HALVES, REVERSE_FIRST_HALF, REVERSE_SECOND_HALF, REVERSE_EACH_2BLOCK, REVERSE_EACH_4BLOCK, and INTERLEAVE_HALVES.

2. **Conflicting context.** The context contains demonstrations generated by a consistent but incorrect rule from the same task family. This condition measures whether the model follows the supplied context even when it conflicts with the parametrically correct rule.

We report the gap between supporting-context and conflicting-context performance as a measure of context addiction. A larger gap indicates that the model is more sensitive to the correctness of the provided context and therefore relies less robustly on a parametrically internalized task rule. We omit no-context evaluation in this synthetic setting because models are trained only on final-query losses following in-context demonstrations. Removing the demonstrations at test time changes the input format and sequence distribution, making no-context prompts out-of-distribution for these models.

D Evaluation Benchmarks

D.1 Language Modeling Suite

The language modeling suite consists of LAMBADA, Penn Treebank (PTB), and WikiSPAN. LAMBADA evaluates word prediction in passages where the target word depends on broad discourse context rather than only local syntax (Paperno et al., 2016). PTB provides a standard corpus-level benchmark for measuring next token prediction on natural text (Marcus et al., 1993). WikiSPAN evaluates language modeling over time-indexed Wikipedia documents, providing an additional testbed for factual and distributional variation in naturally occurring text (Cheng et al., 2024). These benchmarks are well suited to our evaluation because they directly measure the pretraining objective. They therefore test whether increasing the train-time context window improves objective-aligned compression or degrades general next token prediction beyond an intermediate optimum.

D.2 SuperGLUE

SuperGLUE is a suite of natural language understanding tasks that measure capabilities such as entailment, coreference, causal reasoning, and word-sense disambiguation (Wang et al., 2020). The suite aggregates several benchmarks, including BoolQ (Clark et al., 2019), CommitmentBank (Jiang & de Marneffe, 2019), COPA (Roemmele et al., 2011), MultiRC (Khashabi et al., 2018), ReCoRD (Zhang et al., 2018), RTE (Dagan et al., 2006), WiC (Pilehvar & Camacho-Collados, 2019), and WSC (Sakaguchi et al., 2019a). We include SuperGLUE as an intermediate evaluation between language modeling and closed-book multiple-choice question answering. Unlike the language modeling suite, SuperGLUE probes whether representations learned during pretraining transfer to structured language understanding tasks. Unlike the MCQA suite, it does not primarily target factual recall. This makes it well suited for testing whether train-time context length affects general linguistic and reasoning competence rather than only the model's ability to store task-specific knowledge.

D.3 Closed-Book MCQA Suite

The closed-book MCQA suite consists of ARC-Easy and ARC-Challenge (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2019), SocialIQA (Sap et al., 2019), MedQA (Jin et al., 2020), TruthfulQA (Lin et al., 2022), SciQ (Welbl et al., 2017), and WinoGrande (Sakaguchi et al., 2019b). These benchmarks cover complementary forms of knowledge and reasoning, including grade-school science, general commonsense, physical commonsense, social commonsense, medical knowledge, truthfulness, and commonsense coreference. We evaluate them in a closed-book setting, using only the question and answer choices without auxiliary retrieval or supporting documents. This protocol directly supports our goal of measuring parametric knowledge. If longer-context training shifts predictive structure from weights toward context-conditioned computation, the effect should be visible when the model must answer without context at test time.

E PER-DATASET EVALUATION RESULTS

We report evaluation results for each individual dataset underlying the aggregate results in Figure 2. This per-dataset analysis tests whether the observed *inverted-U* pattern is driven by a small number of benchmarks or is visible across the underlying tasks.

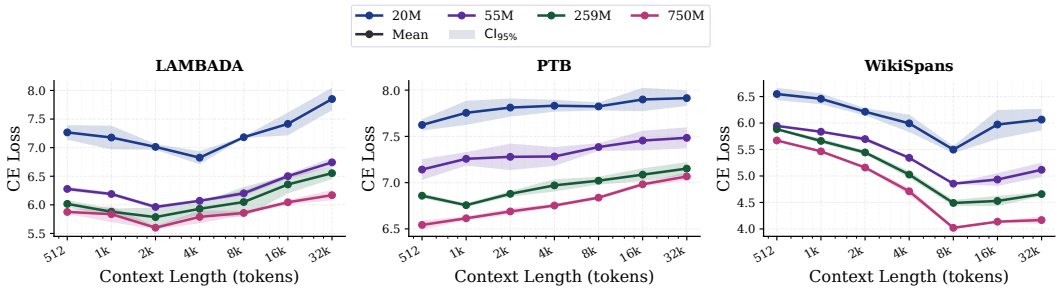


Figure 8: **Language modeling results by dataset.** Across LAMBADA, PTB, and WikiSPAN, longer pretraining windows initially improve next token prediction but eventually degrade performance at longer windows. The dataset-level trends mirror the aggregate language modeling curve in Figure 2, with the strongest performance generally occurring at intermediate context lengths.

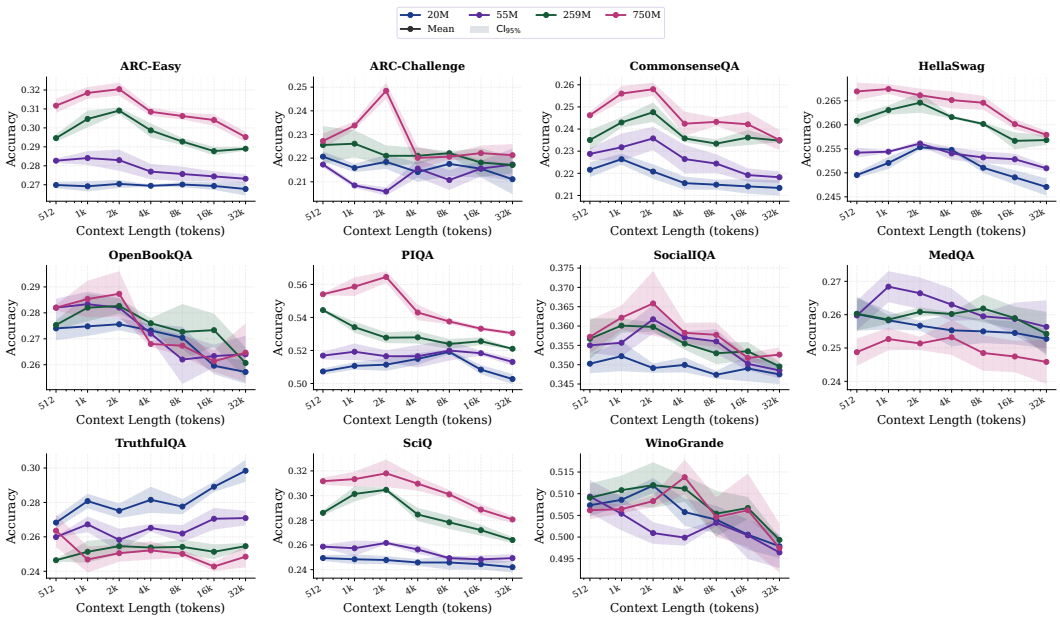


Figure 9: **Closed-book MCQA results by dataset.** Per-benchmark multiple-choice accuracy shows the same qualitative pattern as the aggregate MCQA result, where performance improves up to intermediate context windows and declines for longer ones. This indicates that the degradation is not an artifact of averaging, but appears across diverse forms of closed-book knowledge and reasoning.

This decomposition provides suggestive evidence that the preferred train-time context length partly tracks the effective length of the evaluation distribution. As shown in Table 4, PTB contains the shortest examples and attains its lowest loss at $W = 512$, LAMBADA is longer and attains its lowest loss at $W = 2048$, and WikiSPAN contains the longest examples and attains its lowest loss at $W = 8192$. This trend is consistent with the hypothesis that shorter evaluation tasks saturate at shorter training windows, while longer language modeling contexts benefit from longer train-time context before the long-context degradation appears. The downstream results in Table 5 follow the same broad pattern: MCQA and SuperGLUE have mean lengths of 65 and 135 tokens, respectively, and both exhibit an inflection point at $W = 2048$. Across all five benchmarks, the inflection point, therefore, is consistently on the order of 2^4–2^5 times the mean example length.

Table 4: **Longer language modeling examples favor longer pretraining context windows.** Token-length statistics for each language modeling dataset and the training window W yielding the lowest evaluation loss. Token counts are measured with the `Llama-2-7b` tokenizer.

Dataset	p_{05}	p_{25}	Mean	p_{75}	p_{95}	Inflection Point W
PTB	7	18	28	37	54	512
LAMBADA	71	78	88	95	113	2048
WikiSPAN	216	311	504	668	793	8192

For example, $512/28 \approx 18$, $2048/88 \approx 23$, $8192/504 \approx 16$, $2048/65 \approx 32$, and $2048/135 \approx 15$. Although this relationship is only approximate and the selected context windows are discretized, it suggests a simple scaling rule, where performance tends to saturate, or begin to degrade, once the train-time context window exceeds the mean evaluation length by roughly one to two orders of magnitude.

Table 5: **The trend observed for language modeling benchmarks continues on MCQA and SuperGLUE.** Token-length statistics for each benchmark and the training window W at which performance begins to degrade. Both MCQA and SuperGLUE statistics are weighted equally per benchmark.

Dataset	p_{05}	p_{25}	Mean	p_{75}	p_{95}	Inflection Point W
MCQA	35	48	65	77	110	2048
SuperGLUE	72	101	135	160	221	2048

Overall, the dataset-level decompositions (Figure 8 and Figure 9) support the conclusion that context length acts as a systematic training variable. While individual benchmarks vary in their sensitivity to window size, the broad trend remains consistent: increasing train-time context helps only up to an intermediate regime, after which longer windows reduce both objective-aligned language modeling performance and closed-book downstream accuracy.

F Statistical Significance Tests

This section reports statistical significance tests for the main empirical results in Figs. 1, 2, 3, 4, 5, 6, and 11. Throughout, we use a significance level of $\alpha = 0.05$.

F.1 Long and Short Context Phi-3 and OLMo 3 Variants (Figure 1)

We compare each long-context model with its corresponding short-context variant using a one-sided two-proportion score test in the hypothesized direction that the long-context variant performs worse. For Phi-3, we compare the 128K variants against their corresponding 4K variants; for OLMo 3, we compare the 65K variants against their corresponding 8K variants. We apply Holm correction within each model-family comparison set.

Table 6: **Significance tests for the comparisons in Figure 1.** We report p-values from one-sided two-proportion score tests comparing Phi-3 128K against 4K variants and OLMo 3 65K against 8K variants.

Model	Evaluation	MMLU p	BBH p	MCQA p
Phi-3				
3.8B	Few-shot	0.0048	0.1920	1.23×10^{-4}
3.8B	Zero-shot	4.39×10^{-9}	2.19×10^{-14}	5.86×10^{-7}
14B	Few-shot	0.0028	0.0155	1.72×10^{-5}
14B	Zero-shot	2.91×10^{-39}	2.16×10^{-11}	4.39×10^{-32}
OLMo 3				
7B	Few-shot	0.1204	6.83×10^{-4}	0.0189
7B	Zero-shot	0.0071	4.79×10^{-22}	4.66×10^{-5}
32B	Few-shot	0.0305	4.99×10^{-80}	0.2669
32B	Zero-shot	—	1.57×10^{-4}	6.21×10^{-3}

Phi-3. The 128K variants perform significantly worse than their corresponding 4K variants in 11 of the 12 comparisons.

OLMo 3. The 65K variants perform significantly worse than their corresponding 8K variants in 9 of the 11 comparisons. We do not conduct a test for 32B zero-shot MMLU comparison, since OLMo 3 65K performs better.

Taken together, the long-context variants are significantly worse in 20 of the 23 reported Phi-3 and OLMo 3 comparisons, providing statistical support for the motivating pattern in Figure 1.

F.2 Pretraining Context Length Inflection Tests (Figure 2)

We apply segmented regression tests to assess whether performance exhibits a statistically significant inflection as training context length increases. Degradation begins after 8K tokens for language modeling and after 2K tokens for SuperGLUE and MCQA, consistent with the intermediate optima observed in Figure 2. The inflection is statistically significant for all three evaluation suites at every model scale, indicating that the post-optimum degradation persists as model capacity increases.

Table 7: **Segmented regression inflection tests for Figure 2.** We report p-values for the corresponding inflection test.

Model	Language Modeling	SuperGLUE	MCQA
20M	0.016	0.041	0.022
55M	0.032	0.012	0.013
259M	0.043	0.003	0.005
750M	0.021	0.004	0.004

F.3 Ordered SFT Trends (Figure 3)

We use blocked ordered trend tests over train-time $k \in \{0, 4, 8\}$, treating the four domains as equally weighted repeated blocks. For no-context accuracy, the one-sided alternative is a decreasing trend with k; for the supporting–conflicting accuracy gap, the one-sided alternative is an increasing trend. The ordered effect is significant in 9 of 10 comparisons, as no-context accuracy decreases significantly for every model except Qwen3-8B, while the supporting–conflicting gap increases significantly for all five model sizes. These results show that the qualitative trends in Figure 3 are consistent across domains rather than being driven by a small subset of them.

Table 8: **Ordered trend tests for Figure 3.** Δ is the change from $k = 0$ to $k = 8$ for the corresponding evaluation quantity; p is the one-sided ordered trend test p-value.

Model	No Ctx.		Supporting Ctx. – Conflicting Ctx.	
	Δ	p	Δ	p
0.6B	−5.4	0.0074	+42.8	0.0067
1.7B	−3.6	0.0068	+25.6	0.0084
4B	−5.3	0.0097	+23.3	0.0087
8B	−1.7	0.0645	+23.6	0.0074
14B	−6.3	0.0062	+18.0	0.0096

F.4 Gradient Norm Trends in the Synthetic Study (Figure 4)

We apply a one-sided Kendall rank trend test to the mean gradient norm of each independent training run, testing whether gradient norm decreases as the number of in-context demonstrations increases. Gradient norms decrease significantly for bitwise operations and string operations, whereas mod10 and Caesar exhibit nonsignificant trends. Thus, significantly decreasing gradient norm is observed specifically for the tasks that exhibit context addiction.

Table 9: **Gradient norm trend tests for Figure 4.**

Task	Kendall τ	p
Bitwise Ops.	-0.409	0.0037
String Ops.	-0.641	6.26×10^{-10}
Mod10	$+0.260$	0.9912
Caesar	-0.159	0.0704

F.5 Module-Level Gradient Allocation (Figure 5)

Supervised fine-tuning (Figure 5a). At each model-domain pair, we match FFN-to-SA gradient norm ratios by training step and test the mean paired difference between $k = 8$ and no-context training using a two-sided normal test with Newey–West heteroskedasticity- and autocorrelation-consistent (HAC) standard errors. The $k = 8$ trajectory has a lower mean ratio in all 20 comparisons, and 19 of 20 comparisons are significant.

Table 10: **SFT gradient allocation tests for Figure 5a.** We report two-sided HAC-normal test p-values for the paired difference in FFN-to-SA gradient norm ratio between $k = 8$ and no-context training.

Model	Economics	Law	Health	Psychology
0.6B	1.42×10^{-51}	0.0706	0.0014	3.56×10^{-12}
1.7B	0.0195	3.60×10^{-30}	0.0155	4.57×10^{-4}
4B	0.0087	3.27×10^{-6}	0.0024	1.04×10^{-6}
8B	4.50×10^{-7}	9.13×10^{-19}	1.63×10^{-6}	5.95×10^{-12}
14B	0.0040	4.52×10^{-8}	0.0002	0.0093

Pretraining (Figure 5b). We compute the mean change in the FFN-to-SA gradient norm ratio per doubling of the training context length and test whether this slope is negative. The ratio decreases significantly with increasing context length at every model scale. This consistent negative trend indicates that the shift in gradient pressure toward self-attention persists as model capacity increases.

Table 11: **Pretraining gradient allocation trend tests for Figure 5b.** The slope is the mean change in FFN-to-SA gradient norm ratio per context length doubling.

Model	Mean Slope per Context Doubling	p
20M	-0.038	0.0013
55M	-0.047	0.0051
259M	-0.054	0.0016
750M	-0.045	0.0074

F.6 Module-Restricted Supervised Fine-Tuning (Figure 6)

For each model size and evaluation setting, we use a one-sided two-proportion score test in the hypothesized direction, using the independence approximation. The predicted direction holds in all nine comparisons, with seven of nine being significant. SA-only tuning is significantly better than FFN-only tuning under supporting context and significantly worse under conflicting context for all three models. Under no-context evaluation, SA-only tuning is significantly worse only for Qwen3-4B.

Table 12: **Module-restricted fine-tuning tests for Figure 6.** Δ denotes SA-only minus FFN-only accuracy, and p is the one-sided two-proportion score test p-value. Arrows indicate the hypothesized direction for SA-only relative to FFN-only.

Model	No Ctx. (↓)		Supporting Ctx. (↑)		Conflicting Ctx. (↓)	
	Δ	p	Δ	p	Δ	p
0.6B	-1.9	0.0733	$+4.5$	0.0036	-5.9	0.0005
1.7B	-1.9	0.0739	$+9.0$	0.0001	-10.0	3.17×10^{-6}
4B	-2.9	0.0015	$+2.2$	0.0135	-12.0	1.49×10^{-7}

F.7 Inference-Time Attention to Context (Figure 11)

For each domain, we use a one-sided cluster-based permutation test to determine whether $k = 8$ SFT increases attention to context tokens relative to no-context SFT. For each question, the layer-wise difference curve is sign-flipped during permutation, thereby preserving dependence across layers. Positive differences across contiguous layers are grouped into clusters, and the maximum cluster statistic controls for multiple comparisons over layers. The $k = 8$ checkpoint assigns significantly greater

Table 13: **Permutation tests for Figure 11.**

Domain	Layers	p
Law	4–16	1.4×10^{-4}
Health	4–16	0.0040
Psychology	4–16	5.4×10^{-4}
Economics	4–16	0.0180

attention mass to context tokens in layers 4–16 across all four domains.

G Context Construction for Supervised Fine-Tuning

We construct the supervised fine-tuning data from four MMLU-Pro domains, namely Health, Economics, Law, and Psychology. Since MMLU-Pro provides only a test split, we create an 80/20 split within each domain. The 80% portion is used for supervised fine-tuning, and the remaining 20% portion is reserved for held-out evaluation.

For each question, we generate two sets of documents. The first set is **supporting context**, which is intended to support the correct answer. The second set is **conflicting context**, which is intended to plausibly steer the model toward an incorrect answer. We generate both sets using `gemini-3.1-flash-lite-preview` (Google DeepMind, 2026) through the Gemini API. We use the model's default generation hyperparameters, with `temperature=1.0`, `topP=0.95`, `topK=64`, and `candidateCount=1`. The maximum output length is set by the model limit of 65536 tokens. The system prompt used for generation is provided below:

> **System prompt.**
>
> You are an expert question analyst. Given a multiple choice question, you will generate context and reasoning in JSON format only, with no extra text or markdown.
>
> Your output must be a JSON object with exactly these four keys.
>
> - `supporting_context`: Exactly 8 sentences of factual background information that directly supports arriving at the correct answer. Each sentence must be fully self-contained and independent. No sentence should reference, depend on, or follow logically from any other sentence. Avoid discourse connectives like "furthermore", "however", "therefore", and "this means".
> - `conflicting_context`: Exactly 8 sentences that sound plausible and relevant but subtly steer reasoning toward the second most likely incorrect answer, without explicitly mentioning any answer choice. Each sentence must be fully self-contained and independent. No sentence should reference, depend on, or follow logically from any other sentence. Avoid discourse connectives.
> - `trick_answer`: The single answer option letter, such as `A`, `B`, or `C`, that the `conflicting_context` is designed to steer toward.
> - `reasoning`: A concise 1 to 3 sentence explanation of why the correct answer is correct.

Table 14: **Illustration of the SFT context conditions.** Training uses supporting documents from the target or other domains. Testing uses target-domain documents that either support the correct answer or conflict with it. This allows us to vary the informativeness of context during training, and evaluate how it may impact models' dependency on context during testing.

Question: Which of the following is most likely to produce symptoms similar to anxiety?
Options: A) Hyperthyroidism B) Addison's disease

	Target Domain (Psychology)	Other Domain (e.g. Economics)
Supporting	*Hyperthyroidism involves an overactive thyroid gland that produces an excess of thyroid hormones.* **TRAIN + TEST**	*Price leadership models characterize industries where firms adopt the pricing strategy set by a dominant entity.* **TRAIN**
Conflicting	*Long-term systemic malaise may be mistaken for the hyper-arousal symptoms of a generalized anxiety state.* **TEST**	**UNUSED**

Table 14 summarizes these context conditions. During training, supporting documents may come from either the target domain or the paired source domain, whereas evaluation uses only target-domain documents and varies whether they support the correct answer or conflict with it. The conflicting other-domain condition is not used, since the evaluation is intended to isolate the effect of context correctness within the target domain.

Using these generated documents, we construct train-time context-relevance conditions under a fixed budget of eight documents. The variable $k \in \{0, 4, 8\}$ denotes the number of target-domain documents, while the remaining $8 - k$ documents are drawn from a paired source domain. We pair Law with Health and Economics with Psychology. Thus, Law uses Health as the source of irrelevant documents, Health uses Law, Economics uses Psychology, and Psychology uses Economics. The $k = 8$ condition contains only target-domain context, the $k = 4$ condition contains mixed context, and the $k = 0$ condition contains only irrelevant context. We also include a no-context condition in which all prepended documents are removed. All conditions use the same question set and answer supervision.

As shown in Table 15, supporting and conflicting contexts occupy the same few-hundred-token regime across domains and splits. On the held-out test split, the interquartile ranges largely overlap, where supporting contexts have medians between 170.5 and 201.0 tokens, while conflicting contexts have medians between 163.0 and 190.0 tokens. Supporting contexts are slightly longer overall, but their interquartile ranges remain comparable to those of conflicting contexts within each domain. Thus, the SFT comparison primarily varies the relevance and composition of a fixed eight-document context budget, rather than varying the context length.

Table 15: **SFT contexts have similar total token lengths.** Total tokens in the prepended context per question, measured with the `Llama-2-7b` tokenizer.

Split	Domain	Context	p_{25}	p_{50}	p_{75}
Train	Econ.	Supp.	152.0	170.5	186.0
	Econ.	Conf.	148.0	163.0	176.0
	Health	Supp.	174.2	201.0	225.0
	Health	Conf.	166.0	186.0	204.0
	Law	Supp.	173.0	198.0	219.0
	Law	Conf.	167.0	186.0	202.0
	Psych.	Supp.	157.0	176.0	192.2
	Psych.	Conf.	153.0	168.0	183.0
Test	Econ.	Supp.	158.0	170.5	186.2
	Econ.	Conf.	154.8	163.0	176.0
	Health	Supp.	181.0	201.0	218.0
	Health	Conf.	173.0	190.0	208.0
	Law	Supp.	182.0	201.0	216.5
	Law	Conf.	175.0	190.0	203.0
	Psych.	Supp.	161.5	179.0	193.5
	Psych.	Conf.	155.0	168.0	186.0

H Full Results

H.1 Supervised Fine-Tuning Full Results

Table 16 reports the complete supervised fine-tuning results underlying Figure 3. The table breaks down performance by domain, model size, train-time context condition, and test-time context condition. Across domains, the same qualitative pattern appears. Increasing the amount of target-domain context during fine-tuning improves accuracy when supporting context is available at test time. However, it generally reduces robustness when context is absent or when the supplied context conflicts with the correct answer. These domain-level results show that the context-reliance trend in Figure 3 is not driven by a single domain or model size. Instead, it appears consistently across Health, Economics, Law, and Psychology, as well as across the Qwen3 model scales.

Table 16: Performance across domains, context conditions, and retrieval settings for Qwen3 models. Average denotes the mean over Law, Health, Economics, and Psychology.

		Qwen3-0.6B					Qwen3-1.7B					Qwen3-4B					Qwen3-8B					Qwen3-14B				
Domain	**Eval ↓ – Train →**	No-SFT	No Ctx.	$k=0$	$k=4$	$k=8$	No-SFT	No Ctx.	$k=0$	$k=4$	$k=8$	No-SFT	No Ctx.	$k=0$	$k=4$	$k=8$	No-SFT	No Ctx.	$k=0$	$k=4$	$k=8$	No-SFT	No Ctx.	$k=0$	$k=4$	$k=8$
Law	No Ctx.	15.3	25.7	17.5	14.2	13.7	21.9	27.3	27.3	25.7	23.0	25.1	41.0	38.8	31.1	29.0	31.7	36.6	37.7	33.9	36.6	38.3	48.1	46.4	43.2	38.8
	Supporting Ctx.	41.0	36.1	35.5	65.0	71.0	57.9	49.7	51.9	72.1	75.4	65.0	65.6	68.3	80.9	83.1	73.8	75.4	76.5	84.2	85.8	76.5	78.7	77.6	85.8	86.3
	Conflicting Ctx.	10.9	25.1	19.7	8.2	6.0	13.1	21.3	22.4	6.6	4.9	14.2	21.9	23.0	4.4	4.9	14.2	17.5	20.2	4.4	3.8	13.1	21.3	24.0	5.5	4.4
Health	No Ctx.	20.6	25.5	24.8	23.4	16.3	39.7	36.9	37.6	36.9	35.5	54.6	61.0	58.2	56.7	53.9	63.1	63.1	62.4	63.8	61.0	73.0	68.8	68.1	68.8	66.7
	Supporting Ctx.	75.9	58.9	48.2	80.9	83.7	86.5	77.3	83.0	90.1	87.2	94.3	90.8	92.2	95.7	95.7	96.5	93.6	91.5	95.0	95.7	95.0	92.2	92.2	95.7	95.7
	Conflicting Ctx.	10.6	14.9	23.4	5.0	2.8	18.4	29.8	24.8	22.8	14.9	31.2	36.9	34.8	14.9	16.3	36.9	41.8	45.4	18.4	18.4	36.9	46.1	43.3	27.7	24.8
Economics	No Ctx.	30.3	34.9	36.8	31.6	30.3	51.3	47.4	48.0	44.1	43.4	61.8	58.6	57.2	55.9	55.9	66.4	63.8	63.2	63.2	60.5	74.3	72.4	73.0	70.4	67.1
	Supporting Ctx.	53.9	43.4	39.5	63.8	59.2	69.7	66.4	58.6	74.3	71.7	80.9	77.6	75.7	81.6	80.9	82.9	75.0	75.0	81.6	82.2	83.6	84.2	84.9	84.9	84.9
	Conflicting Ctx.	19.7	32.2	27.0	17.8	14.5	36.8	36.2	38.2	26.3	24.3	51.3	50.0	48.0	38.8	34.2	58.6	50.7	50.0	40.1	38.8	57.2	56.6	55.9	48.7	47.4
Psychology	No Ctx.	27.3	43.2	35.3	35.3	32.4	46.0	51.8	49.6	50.4	46.0	58.3	62.6	63.3	61.9	57.6	70.5	66.9	66.9	64.7	65.5	68.3	66.9	71.2	66.9	61.2
	Supporting Ctx.	71.9	60.4	67.6	82.0	81.3	81.3	79.9	80.6	88.5	84.2	88.5	91.4	88.5	92.1	91.4	89.2	89.9	93.5	94.2	95.0	86.3	91.4	90.6	92.8	92.1
	Conflicting Ctx.	20.1	35.3	32.4	12.9	12.2	31.7	41.7	43.2	34.5	26.6	41.7	51.1	50.4	30.9	33.8	39.6	48.9	50.4	31.7	33.1	40.3	49.6	48.9	38.8	37.4
Average	No Ctx.	23.4	32.3	28.6	26.1	23.2	39.7	40.8	40.6	39.3	37.0	50.0	55.8	54.4	51.4	49.1	57.9	57.6	57.6	56.4	55.9	63.5	64.1	64.7	62.3	58.4
	Supporting Ctx.	60.7	49.7	47.7	72.9	73.8	73.9	68.3	68.5	81.2	79.6	82.2	81.3	81.2	87.6	87.8	85.6	83.5	84.1	88.7	89.7	85.4	86.6	86.3	89.8	89.8
	Conflicting Ctx.	15.3	26.9	25.6	11.0	8.9	25.0	32.2	32.2	22.6	17.7	34.6	40.0	39.0	22.2	22.3	37.3	39.7	41.5	23.6	23.5	36.9	43.4	43.0	30.2	28.5

H.2 Solution Complexity Full Results

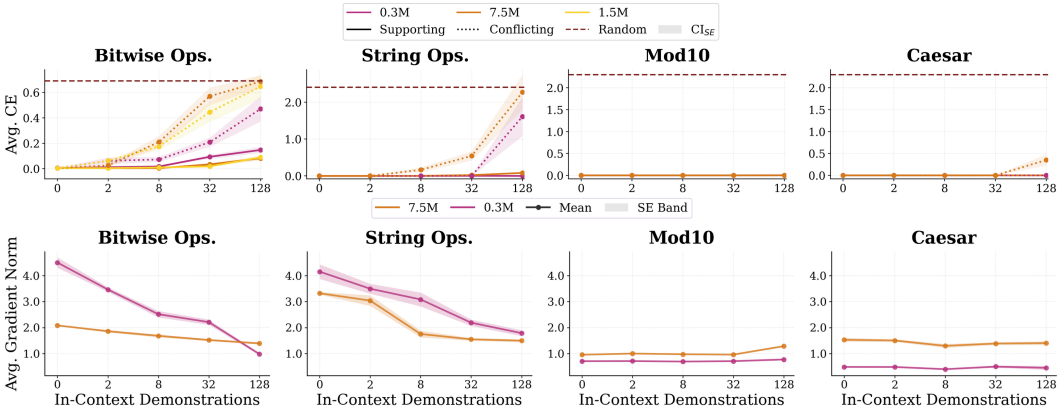


Figure 10: **Lower complexity solutions and context addiction co-occur across model sizes.** Across synthetic tasks and model scales, tasks with larger supporting–conflicting gaps also show lower average training gradient norms as the number of in-context demonstrations increases. This pattern is strongest for bitwise and string operations, while mod10 arithmetic and Caesar cipher remain comparatively stable. Context addiction becomes stronger as model size grows.

Figure 10 provides the full model-scale breakdown for the complexity analysis (§5.1). The same qualitative pattern holds across model sizes. Tasks that exhibit stronger context addiction also show decreasing average training gradient norms as train-time demonstrations increase. For bitwise and string operations, longer contexts produce lower gradient norm solutions and larger supporting–conflicting gaps. In contrast, mod10 arithmetic and Caesar cipher show comparatively stable gaps and no consistent decrease in the gradient norm proxy.

The effect also strengthens with model size. Larger models show larger growth in the supporting–conflicting gap under longer train-time context. This suggests that increased capacity does not prevent context addiction in these settings. Instead, when demonstrations provide a lower complexity route to

reducing loss, larger models appear even more able to exploit that contextual solution. These results support our conclusion that lower complexity solutions and context addiction emerge together, and show that this relationship is consistent across model scales.

H.3 Token-Level Attention Allocation

Prior work has shown that transformer layers exhibit functional specialization, where early layers encode local and syntactic features, middle layers support contextual integration and retrieval, and later layers refine representations for prediction (Tenney et al., 2019; Jin et al., 2024). Motivated by this layer-wise structure, we ask whether the train-time shift in gradient allocation (§5.2) is reflected in inference-time attention to context tokens. We compare the fine-tuned models analyzed in §5.2 under supporting-context evaluation. For a layer ℓ, head h, and answer-token query position q, let $A^{\ell,h}_{q,j}$ denote the normalized attention weight from query position q to key position j. We define the attention mass assigned to context tokens as $\sum_{j\in\mathcal{C}} A^{\ell,h}_{q,j}$, where $\mathcal{C}$ is the set of tokens in the prepended documents, excluding the question, answer choices, and generated answer tokens. For each layer and head, we average this quantity over generated answer-token positions and evaluation examples, and then report the maximum over heads within each layer. Our choice to report the maximum attention mass over heads is motivated by prior work using context-attention ratios to diagnose contextual grounding and showing that retrieval and in-context processing can be concentrated in sparse, specialized heads (Voita et al., 2019; Olsson et al., 2022; Chuang et al., 2024; Wu et al., 2024).

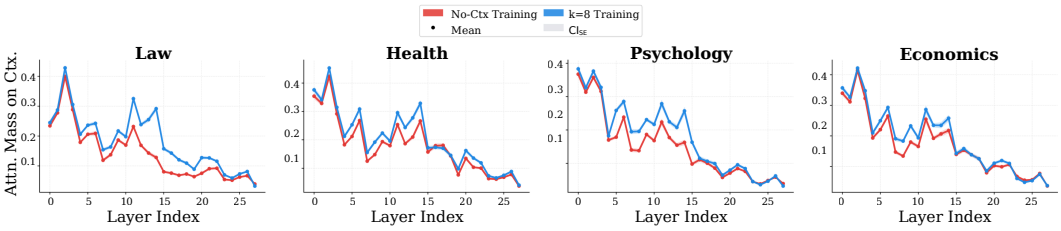


Figure 11: **SFT with task-relevant context increases inference-time attention to context.** Qwen3-1.7B fine-tuned with $k = 8$ target-domain documents assign more attention mass to context tokens.

Figure 11 reports results for `Qwen3-1.7B`, while the additional model results below show the same pattern. Models trained with task-relevant context allocate significantly more attention to context tokens at test time across all four domains (App. F.7). The shift is concentrated in middle layers, consistent with their role in contextual integration. This shows that increased train-time context availability changes the test-time computation, where the resulting models rely more heavily on the supplied context at prediction time.

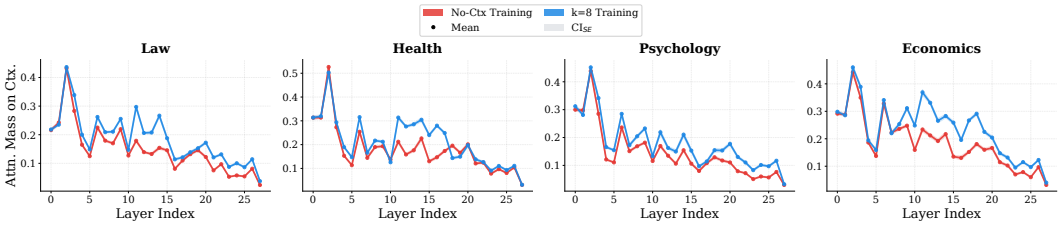


Figure 12: **Attention allocation for Qwen3-0.6B.** Fine-tuning with task-relevant context increases attention mass on context tokens, especially in middle layers.

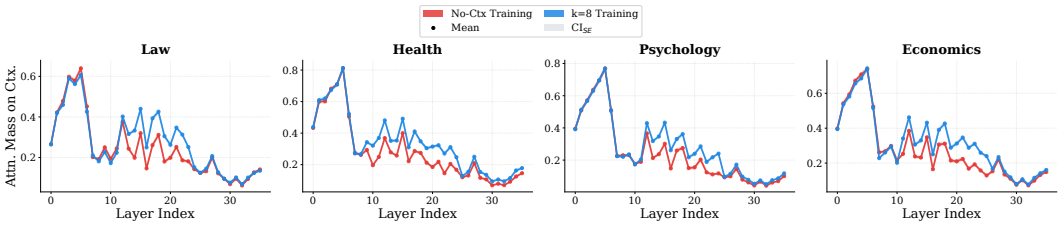


Figure 13: **Attention allocation for Qwen3-8B.** The context-trained model assigns more attention to context tokens than the no-context model under supporting-context evaluation.

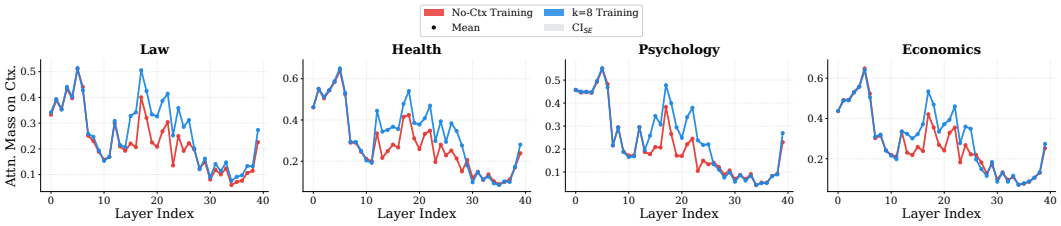


Figure 14: **Attention allocation for Qwen3-14B.** Task-relevant fine-tuning shifts inference-time attention toward supplied context across tasks, with strongest effects in middle layers.

I Proof of Parametric Information Monotonicity

Let ℓ denote the pointwise task loss and $\mathcal{R}_k(\Pi, q_k) = \mathbb{E}[\ell(Y, \hat{Y})]$ denote the corresponding population risk. The risk threshold ρ denotes an upper bound on this population risk. We reuse the notation from §4, where $X^{(k)}$ denotes the input available at context size k, $T_{k,m}$ is the measurable projection from a longer input $X^{(m)}$ to its aligned k-token subwindow, $\Pi(\cdot \mid \tau)$ is the task-conditional weight channel, and q_k is the predictor at context size k, inducing predictions $\hat{Y} \sim q_k(\cdot \mid X^{(k)}, W)$.

Proof. Fix $k < m$, and take any feasible pair (Π, q_k) for $\mathcal{I}_k(\rho)$, so that

$$\mathcal{R}_k(\Pi, q_k) \leq \rho.$$

We construct a feasible pair at context size m by using the same weight channel and defining the longer-context predictor by first projecting the input to the corresponding k-token subwindow:

$$q_m(\hat{y} \mid X^{(m)}, W) = q_k(\hat{y} \mid T_{k,m}(X^{(m)}), W).$$

By the nested-input assumption, $X^{(k)} = T_{k,m}(X^{(m)})$ almost surely. Therefore, under the constructed pair (Π, q_m), the induced joint law of $(X^{(k)}, Y, W, \hat{Y})$ is the same as under (Π, q_k). Hence the population risk is preserved:

$$\mathcal{R}_m(\Pi, q_m) = \mathcal{R}_k(\Pi, q_k) \leq \rho.$$

Moreover, because the weight channel is unchanged, the task information stored in the weights is also unchanged. Thus every feasible pair at context size k induces a feasible pair at context size m with the same parametric information. Taking the infimum over all feasible (Π, q_k) gives

$$\mathcal{I}_m(\rho) \leq \mathcal{I}_k(\rho).$$

□

J Additional Related Work

Two modes of learning with context. A growing body of work studies how language models balance parametric and contextual strategies for solving tasks. This distinction is often framed as *task retrieval*, in which demonstrations activate task structure already encoded in the model's weights (Pan et al., 2023; Lin & Lee, 2024), versus *task learning*, in which the model infers a new input-output rule from examples in context (Pan et al., 2023; Fang et al., 2025). These modes can coexist and compete during pretraining (Wang et al., 2023), and their relative use depends on the training process, task distribution, and informativeness of the context (Anand et al., 2025; Chan et al., 2022; Raventós et al., 2023). Empirical evidence further supports this view. Corrupted-label demonstrations can still improve performance, suggesting that demonstrations may identify a latent task rather than fully specify a new mapping (Min et al., 2022). Other probes measure when models retrieve internal knowledge versus learn from demonstrations in regression tasks (Nafar et al., 2025), substitution-cipher tasks (Fang et al., 2025), and many-shot prompting settings (Bertsch et al., 2025). These studies establish that language models can interpolate between parametric and contextual solutions. However, they primarily vary inference-time prompts, demonstration distributions, task families, or model scale. The role of the *training context window* in governing this bi-modal tradeoff remains largely unstudied, motivating our focus in this work.

Language modeling as compression. The idea that prediction, compression, and intelligence are linked predates modern LMs. Classical source coding connects probabilistic prediction to lossless coding (Shannon, 1948), while the MDL frames learning as compression without overfitting (Rissanen, 1978; Yu, 1998). This view also motivates compression-based evaluations of AI models, from text-compression tests as intelligence benchmarks (Mahoney, 1999) to the Hutter Prize's Wikipedia-compression objective (Hutter, 2006). In machine learning, description length and prequential coding have been used to measure what neural models learn (Hinton & van Camp, 1993; Blier & Ollivier, 2018) and how efficiently representations support downstream labels (Voita & Titov, 2020; Bornschein et al., 2022). Recent work revisits these ideas for modern language models. Delétang et al. (2024) demonstrate that autoregressive LMs can be paired with entropy coding to form strong general-purpose compressors, and use this lens to study scaling, tokenization, and in-context learning. Compression also serves as an empirical diagnostic, correlating with downstream performance (Huang et al., 2024)

and guiding data selection (Yin et al., 2024). Relatedly, Elmoznino et al. (2025) interprets in-context learning through prequential coding, showing that next token prediction can favor simple predictors inferred from examples in the prompt. Practical neural compressors similarly build on LM predictions, including LLMZip (Valmeekam et al., 2023), FineZip (Mittu et al., 2024), and low-complexity learned compressors such as L3TC (Zhang et al., 2024a). These works establish compression as an evaluation criterion, a data-selection signal, or a mechanism for building compressors. Our use of compression is different, since we ask how the availability of context during training changes the *location* of compressed task information. Rather than measuring whether a model compresses well overall, we study whether longer train-time contexts shift predictive structure away from parametric storage and toward context-conditioned computation.