AN INFORMATION THEORETIC PERSPECTIVE ON AGENTIC SYSTEM DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic language model (LM) systems have rapidly become central to modern workflows, powering applications like "Deep Research" and "Claude Code." As contexts grow beyond what even the largest frontier models can process effectively, multi-LM architectures have emerged to overcome context limitations. Beneath their apparent diversity lies a recurring pattern: smaller "compressor" LMs distill raw context into compact text that is then consumed by larger "predictor" LMs that interact with the user. Despite their popularity, the design of *compressor-predictor* systems remains largely ad hoc. Little guidance exists on how compressor and predictor choices shape downstream performance. In practice, attributing gains to compression versus prediction requires exhaustive pairwise sweeps, which is costly and task-specific. We argue that these agentic system design questions are, at root, information-theoretic. Viewing the compressor LM as a *noisy channel*, we introduce a simple estimator of the mutual information between the context and its compression to quantify compression quality in a task-independent way. We show that mutual information strongly predicts downstream performance, independent of any specific task. Through an information-theoretic framework, we perform a comprehensive empirical analysis across five datasets and three model families. Results reveal that larger compressors not only are more accurate, but also more token-efficient, conveying more bits of information per token. A 7B QWEN-2.5 compressor, for instance, is $1.6\times$ more accurate, $4.6\times$ more concise, and conveys $5.5\times$ more bits of mutual information per token. Across the datasets studied, scaling compressors is substantially more effective than scaling predictors, enabling larger on-device compressors to pair with smaller cloud predictors. When applied to a Deep Research system, these principles enable local compressors as small as 3B parameters to recover 99% of frontier-LM accuracy at 26% of API costs.

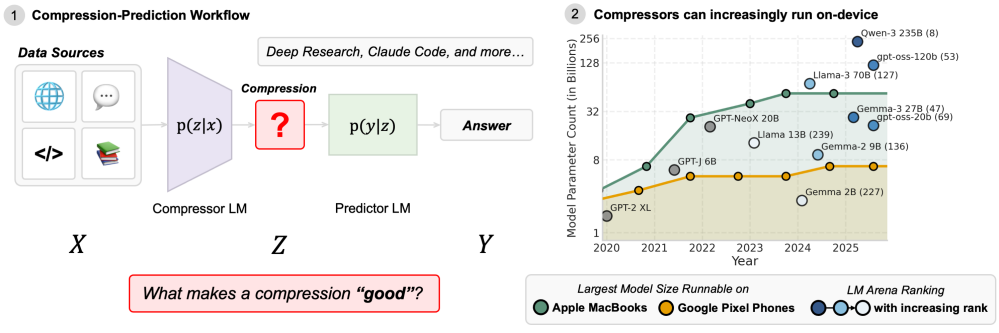

Figure 1: **Why compressors matter.** Many agentic LM systems rely on compressors, and personal devices are growing powerful enough to host them on-device. **(Left)** A compressor condenses a long input X into a shorter summary Z that a predictor expands into the final answer Y. **(Right)** Consumer hardware can now run increasingly large open-weight LMs, shown for Google Pixel phones and Apple MacBook laptops under FP16 precision with memory estimates from Modal (Lu, 2024). LM-Arena ranks indicate relative performance.

1 INTRODUCTION

Agentic language model (LM) systems have quickly become the backbone of modern AI workflows. From "Deep Research" systems (Hadfield et al., 2025) to Claude Code (Anthropic, 2025), millions of users now interact with pipelines where one model processes information and another builds on its outputs. Modern workflows commonly involve analyzing and generating more tokens than even the largest frontier models can handle effectively, degrading model performance—a failure mode referred to as *context rot* (Hong et al., 2025). Multi-LM systems coordinate multiple models to manage reasoning and memory beyond a single model's context window. While these architectures vary widely, a recurring pattern emerges across domains: smaller *compressor* models distill raw contexts into compact texts, which are then consumed by larger *predictor* models that output an answer and interact with the user (Figure 1) (Hadfield et al., 2025; Shen et al., 2023).

At present, however, designing *compressor-predictor* agentic systems remains largely trial-and-error. We lack a basic understanding of how the choice of compressor and predictor affects downstream performance. Specifically, we cannot determine whether credit belongs to the compressor's distillation or the predictor's reasoning—we lack task-agnostic methods to evaluate the compressor's outputs independently from downstream performance. This is because we are unable to measure how much of the original context the compressor actually preserves, which in turn determines how effectively the predictor can reason. This attribution problem has immediate practical consequences: as new models are released and practitioners swap components, they have no principled way to identify which module to improve without sweeping across the compound system from scratch.

To address this gap, we take an information-theoretic perspective, viewing the compressor as a *noisy channel* between the raw data and the predictor model. This framing allows us to evaluate communication between the two models rather than treat it heuristically. We propose using *mutual information* (MI) between the raw context and its compression as a task-agnostic measure of compressor efficacy—analogous to how perplexity serves as a task-agnostic predictor of downstream performance (Hoffmann et al., 2022; Kaplan et al., 2020). We then conduct a *rate-distortion analysis* to measure how downstream task performance varies with the degree of compression. While it is intractable to calculate MI between two token sequences linked via a nonlinear model, we develop a simple, unbiased estimator that can be computed via modern inference servers without requiring full vocabulary log probabilities.

With this new information-theoretic lens, we perform extensive empirical studies on five datasets (LONGHEALTH (Adams et al., 2024), FINANCEBENCH (Islam et al., 2023), QASPER (Dasigi et al., 2021), WILDCHAT (Zhao et al., 2024), and FINEWEB (Penedo et al., 2024)) to answer the following questions:

1. *Should you spend compute on the compressor or predictor?* We find that compressor quality overwhelmingly governs performance: scaling a QWEN-2.5 compressor from 1B to 7B improves accuracy by 60% whereas scaling the predictor from 70B to 405B yields only a 12% improvement on LONGHEALTH. This establishes a simple design principle: "front-load" compute into compressors, perhaps running on-device, to reduce dependence on massive cloud-hosted predictors. (Section 3.1)
2. *Which compressors are more token-efficient?* We find that larger compressors emit fewer output tokens while maintaining quality: in many model families, scaling compressor size not only improves accuracy but also produces compressions that are up to $4.6\times$ more concise. This token-efficiency yields sublinear scaling of FLOPs-per-generation as a function of model size. Strikingly, increasing QWEN-2.5 compressor from 1.5B to 7B, only adds 1.3% more FLOPs-per-generation. (Section 3.1)
3. *Which factors determine compression quality and how do they relate to downstream performance?* We find that compressors' outputs carry up to $5.4\times$ more MI about the context (Section 3.2). Rate-distortion analysis reveals that information rate (MI per token) correlates strongly with downstream performance and perplexity ($r = -0.84$, $R^2 = 0.71$), providing a practical proxy for predicting system performance without full end-to-end evaluation (Section 3.3).
4. *With so many knobs to turn, which factors should you focus on for agentic system design?* We perform a meta-analysis across model families, sizes, and datasets, exposing a clear hierarchy of importance: compressor model family $>$ compressor size $>$ predictor size. (Section 3.4)

As a practical demonstration, we apply our findings to a simplified Deep Research pipeline, where a single predictor aggregates outputs from multiple compressors. This system achieves 99% of frontier-LM accuracy on the DEEPRESEARCH BENCH benchmark (Du et al., 2025) using local compressor models as small as 3B, reducing API costs by 74% (Section 3.5).

2 PRELIMINARIES

2.1 RELATED WORK

Agents have long been a concept in computer science (Franklin & Graesser, 1997; Masterman et al., 2024; Park et al., 2023; Shoham, 1993). In the age of AI, agentic architectures have become a practical way to improve cost-efficiency and accuracy on complex tasks (Belcak et al., 2025; Chen et al., 2023; Du et al., 2025; Narayan et al., 2025; Shen et al., 2023). Most prior work reports end-to-end utility such as accuracy, latency, and cost, while overlooking the communication channel itself. We instead analyze the intermediate communication, focusing on asymmetric compressor-predictor setups and their scaling. Design choices studied in the literature include model size (Narayan et al., 2025; Wang et al., 2024), number of agents and communication rounds (Chen et al., 2024; Schluntz & Zhang, 2025), research depth (Zhang et al., 2025), and planning or decomposition strategies (Chen et al., 2023; Erdogan et al., 2025; Saad-Falcon et al., 2024; Yao et al., 2023). Overall, they find that performance improves with larger models, more agents, and additional rounds. Other research focuses on learned or automated optimization of agentic architectures (Hu et al., 2024; Saad-Falcon et al., 2024) and agentic memory systems (Han et al., 2025).

The next section reviews prior work on information-theoretic approaches in deep learning and formalizes the setup underlying our analysis.

2.2 INFORMATION THEORETIC PROBLEM SETUP

Information theory has provided a formal lens for characterizing how neural networks encode and transform information in deep learning. Researchers have used it to analyze intermediate representations in neural networks (Farquhar et al., 2024; Kawaguchi et al., 2023; Saxe et al., 2019; Tishby & Zaslavsky, 2015), understand information dynamics of of model learning and generalization (Saxe et al., 2019; Westphal et al., 2025), evaluate natural language generations including summaries and token sequences (Arda & Yener, 2025; Darrin et al., 2024; Fränken et al., 2024; Shani et al., 2025), and define training objectives and regularization terms (Chen et al., 2016; Goldfeld & Polyanskiy, 2020; Hjelm et al., 2018; Kirsch et al., 2020; Oord et al., 2018; Wang et al., 2020). We extend this framework beyond single-model analysis to study communication between two LMs.

We start with a simple compressor-predictor system, including one compressor LM, and one predictor LM. Let X be the input context and Y the answer to that query. We consider a two-stage process

$$X \xrightarrow{p(z|x)} Z \xrightarrow{p(y|z)} Y.$$

The compressor $p(z \mid x)$ is modeled as a noisy channel, which compresses the context into a summary Z. The predictor $p(y \mid z)$ then uses this summary to generate the answer Y.

We proceed to define our MI estimator.

Estimating mutual information We want to measure the amount of information Z contains about X, denoted as $I(X;Z)$. Larger MI values $I(X;Z)$ indicate that the compression retains more information about the original context. Mutual information estimation is a well-studied problem in statistics and machine learning (Murphy, 2023; Poole et al., 2019; Paninski, 2003; McAllester & Stratos, 2020; Belghazi et al., 2018). However, many existing estimators are not practical in our setting. While many variational bounds require access to the underlying distribution or the training of an auxiliary model, we want to directly use the log probabilities exposed by LM inference engines.

To estimate MI, we start with the KL divergence (Kullback & Leibler, 1951) representation:

$$\begin{aligned} I(X;Z) &= D_{\mathrm{KL}}\big(p(x,z) \,\|\, p(x)\,p(z)\big) \\ &= \mathbb{E}_{x,z\sim p(x,z)}\left[\log \frac{p(z|x)}{p(z)}\right]. \end{aligned}$$

While computing $p(z)$ is intractable, we can sample from and evaluate our encoder $p(z|x)$, and take samples from the data distribution $p(x)$,

$$I(X;Z) = \mathbb{E}_{x,z\sim p(x,z)}\left[\log \frac{p(z|x)}{\mathbb{E}_{x'}[p(z|x')]}\right],$$
$$\approx \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\log p(z_{ij}|x_i) - \log\left(\frac{1}{N}\sum_{l=1}^{N} p(z_{ij}|x_l)\right)\right] \equiv \hat{I}(X;Z),$$

where $z_{ij} \sim p(z|x_i),\ i = 1,\ldots,N,\ j = 1,\ldots,M$ and $x_l \sim p(x),\ l = 1,\ldots,N$.

Note that $\hat{I}(X;Z) \leq \log(N)$, where the maximum is achieved when $p(z_{ij}|x_i) = 1$ for all i, j and $p(z_{ij}|x_l) = 0$ for all $l \neq i$. We do not claim theoretical optimality with this estimator. Instead, our aim is a practical estimator that can be be computed directly using modern inference engines. With this estimator, we do not need access to the full probability distribution over the vocabulary, which allows us to use accelerated inference engines such as SGLang (Zheng et al., 2024).

In our experiments, each compression Z is generated conditioned on a query Q, so we estimate $I(X;Z \mid Q)$, which simply requires conditioning all terms on Q. While $I(X;Z \mid Q) \geq 0$, in practice, our Monte Carlo estimate can produce small negative values due to finite-sample variance. We correct these artifacts by clipping MI to zero. Furthermore, we find that LMs at 1–3B could assign high likelihoods to nonsensical token sequences, indicating miscalibration. Thus, we evaluated the log probabilities using a proxy model at the 7–8B scale. To mitigate biases, the proxy model was selected from a different family. We acknowledge that this could introduce variance and biases, and investigate their effects through ablation studies in Appendix E.1.4.

Rate-distortion theory quantifies the trade-off between *rate*—i.e., the amount of information the compression carries about the input—and *distortion*—the error in the prediction. We define *rate* (or *bit-efficiency*) as

$$R = \frac{I(X;Z \mid Q)}{L},$$

for L output tokens (measured in bits of mutual information per token). For simplicity, we define distortion as $D = 1 - \text{ACC}(Z)$, by using the accuracy $0 \leq ACC(Z) \leq 1$. See the R-D curve in Figure 19 (left). As rate increases, we expect distortion to converge towards a lower bound (irreducible error). See Appendix B.5 for further details.

3 Results

We evaluate the compressor-predictor system as an information bottleneck across different tasks and domains. Beginning with a comprehensive scaling analysis of compressor and predictor model family and sizes, we find that larger compressors are not only more accurate but also more concise, leading to sublinear FLOPs-per-generation growth relative to model size. We conclude that scaling compressors is more effective than scaling predictors. Building on this, we show that mutual information rate closely tracks downstream accuracy and perplexity, providing a task-agnostic signal of compression quality. A meta-analysis highlights compressor model family and compressor size outweighing predictor size as the most important factors. Finally, we validate these principles in a Deep Research pipeline, where local compressors deliver frontier-level accuracy at a fraction of the cost.

Datasets We study our setup on five datasets, two synthetic, three non-synthetic: (a) LongHealth, a set of synthetic clinical reports and 20 patient histories (Adams et al., 2024). We hide the multiple-choice options from the LMs to treat it as a question-answering (QA) dataset. (b) FinanceBench, a collection of 150 10-K filings and reports paired with QA tasks (Islam et al., 2023). (c) QASPER, a QA dataset of 5,049 scientific research papers (Dasigi et al., 2021). (d) WildChat, a large-scale LM conversation dataset (Zhao et al., 2024). (e) FineWeb, a dataset of processed web pages from CommonCrawl (Penedo et al., 2024). See Appendix D.1 for more details on datasets.

Evaluation procedure We run each experiment with $S = 5$ random seeds. We evaluate prediction quality using accuracy for LongHealth, FinanceBench, and QASPER, assessing correctness of the predictions against the ground-truth using a GPT-4o-mini judge. We use perplexity for WildChat and FineWeb, evaluating the log probabilities of a Llama-3.1-8B model.

Compressor model As compressors, we use smaller open-source LMs of the model families LLAMA-3 (Grattafiori et al., 2024), QWEN-2.5 (Qwen et al., 2025), and GEMMA-3 (Team et al., 2025). In our analysis, we choose to focus on GPT-style non-reasoning architectures. We additionally conduct preliminary experiments on reasoning and mixture-of-experts (MoE) LMs of the QWEN-3 family (Yang et al., 2025). See Appendix D.2 for further details.

Predictor model We evaluate larger frontier models GPT-4O (OpenAI et al., 2024) as well as LMs of the LLAMA-3 (1B, 8B, 70B, 405B) and QWEN-2.5 (72B) families as predictors.

3.1 WHERE DO YOU NEED THE FLOPS: IN COMPRESSORS OR PREDICTORS?

We ask: should we scale compressors, which distill large amount of information into concise summaries, or predictors, which reason over the provided summaries to solve complex tasks?

In the following section, we show that scaling the compressor LM yields more significant gains. We vary the compressor model size and study its effects on downstream accuracy, the length of the compressed summaries, and the number of overall FLOPs-per-generation. We also establish scaling laws linking compressor model size to downstream performance.

First, we examine question-answering (QA) accuracy when scaling both compressor and predictor model size.

Downstream performance is a function of compressor size. We illustrate how downstream accuracy increases as model size increases (Figure 2, left). On LONGHEALTH, 7–8B models are up to $3.1\times$ more accurate than 1–1.5B models and surpass the GPT-4O-only baseline by 4pp. On FINANCEBENCH, 7–8B models are up to $2.6\times$ more accurate than 1–1.5B models and are able to recover 97% of the GPT-4O-only baseline accuracy. The same scaling behavior holds for GEMMA-3 models.

Larger compressors are more concise. In this section, we study the number of compression output tokens as a function of compressor size. We find that larger compressors are more concise (Figure 2) without sacrificing accuracy. Specifically, 7–12B compressors are up to $4.6\times$ more token-efficient than their 1–1.5B counterparts within the same model family. QWEN-2.5 models tend to be more concise than LLAMA and GEMMA-3, suggesting that models can significantly vary in their communication profiles.

Compression compute cost scales sublinearly with compressor size. We combine the number of parameters with output token counts to estimate FLOPs-per-generation for each model family and size, i.e., the *actual* compute cost (Appendix B.1). Because larger compressors generate fewer tokens while maintaining accuracy, FLOPs-per-generation scale sublinearly with model size. Different model families exhibit distinct scaling behaviors (Figure 2): QWEN-2.5 compressors can scale from 1.5B to 7B parameters with only a 1.3% increase in FLOPs-per-generation on LONGHEALTH.

Scaling compressors is more effective than scaling predictors. Figure 3 shows that scaling the predictor LM provides only marginal improvements in accuracy once a baseline predictor capacity ($\approx$ 8B–70B) is reached. The gains in accuracy by increasing predictor size from 70B to 405B are within 12% (LONGHEALTH) and 1% (FINANCEBENCH). In contrast, scaling the compressor LM for both families leads to steeper increases in performance for fewer FLOPs-per-generation spent. For the QWEN-2.5 compressor family, FLOPs-per-generation meaningfully increase only when transitioning from 7B to 14B (models up to 7B all have roughly constant FLOPs-per-generation).

You can trade local for remote compute. As shown in Figure 1 (right), powerful models up to 27B can run without aggressive quantization on current-generation laptops. We anticipate the trends to continue and that even bigger models could run locally, and for free. Our results motivate "front-loading" FLOPs into local compressors to reduce cloud costs for serving the predictor (Figure 3).

Analysis of compressor errors. Errors in the compression step can be characterized into one of three categories: (a) the compression contains an incorrect answer (36.3% of compressor errors); (b) the compression contains no answer (33.3% of compressor errors); and (c) the compression omits

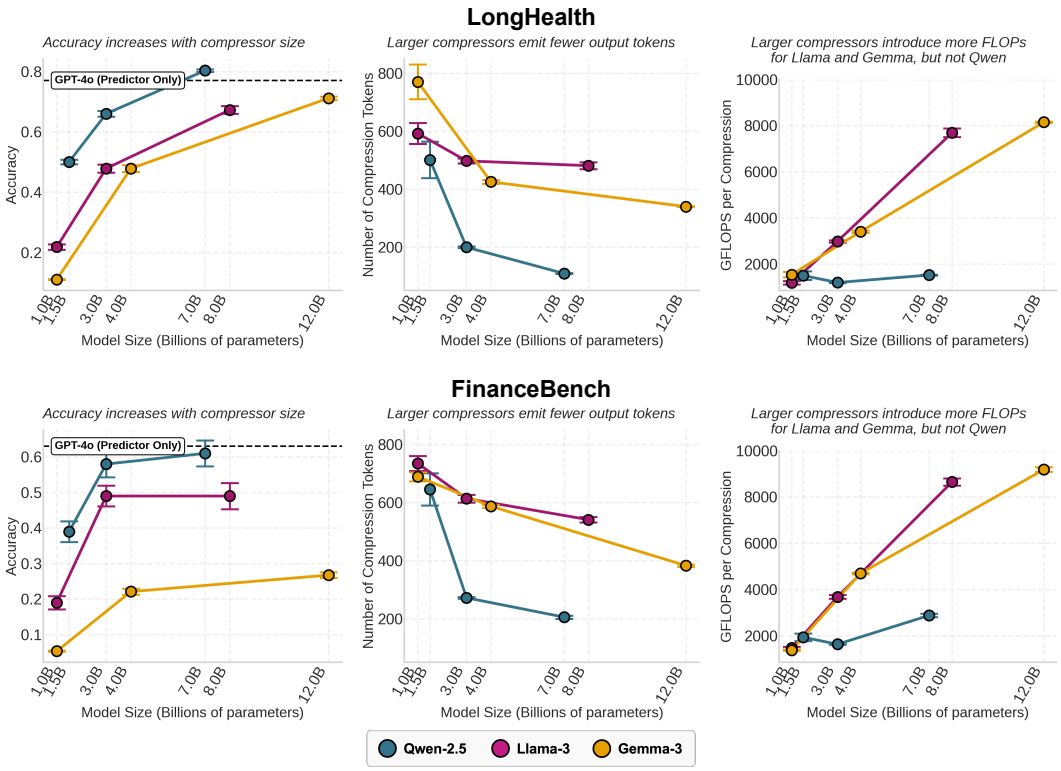

Figure 2: **Downstream accuracy, compression length, and compute cost scale with compressor size (Top: LONGHEALTH; Bottom: FINANCEBENCH).** We scale compressor model size and reports a different metric on the y-axis of each column: **(Left)** accuracy, with the black dotted line showing the GPT-4o model baseline. **(Middle)** compression length, **(Right)** GFLOPs-per-compression. Vertical bars denote standard errors. Larger compressors produce shorter outputs with higher downstream accuracy. Similar trends hold on QASPER (Appendix E.1.1), WILDCHAT (Appendix E.1.2) and FINEWEB (Appendix E.1.3).

details or parts of the information necessary for the answer (30.4% of compressor errors). For more details on compressor errors, refer to Appendix D.7.

3.2 WHICH COMPRESSORS MAXIMIZE COMMUNICATION EFFICIENCY?

We want to select compressors that provide maximal task-relevant information, ideally communicated in as few tokens as possible. The downstream QA accuracy and compression length do not fully capture compression quality. Instead, we turn to a information-theoretic framing: we estimate the mutual information $I(X; Z \mid Q)$ between context X and generated compression Z conditioned on query Q for each compressor model in our scaling analysis, using the Monte Carlo estimator described in Section 2.2.

Larger compressors retain more mutual information and are more bit efficient. We observe that $I(X; Z \mid Q)$ increases as compressor size increases (Figure 4). Larger, more expressive compressor models carry more mutual information between the original document and the compression into the summary.

On LONGHEALTH, while LLAMA compressors are far from the theoretical maximum, we find that QWEN-2.5 and GEMMA-3 models produce compressions that saturate in mutual information at the largest model sizes. By contrast, on FINANCEBENCH, mutual information saturates already at the 3B scale. We observe this saturation behavior primarily on datasets with a highly heterogeneous corpus of context documents.

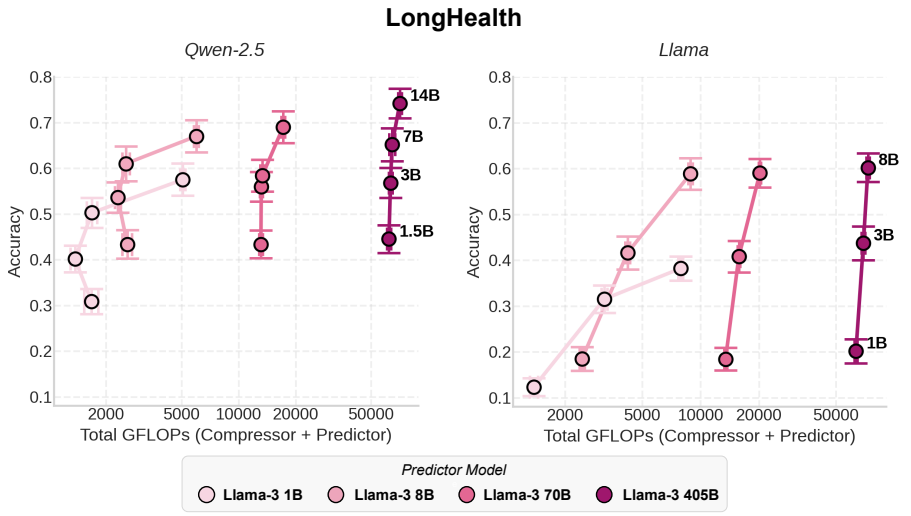

Figure 3: **Scaling compressors is more effective than scaling predictors on LONGHEALTH.** The y-axis reports accuracy and x-axis shows total compute cost in FLOPs-per-generation (log-scale). We compare compressor LMs from two families: **(Left)** QWEN-2.5, **(Right)** LLAMA-3. We scale predictor (marker color) and compressor (marker label) sizes and measure the total FLOPs-per-generation and downstream accuracy on QA tasks. Appendix E.1.6 shows consistent trends on FINANCEBENCH.

Combining the scaling effects of mutual information with the observation that larger compressors omit fewer tokens, we find that larger compressors are more bit efficient (Figure 4).

We ablate across multiple proxy model choices and find that the mutual information scaling trends remain consistent across different proxy choices (Appendix B.5). Furthermore, we estimate mutual information without a proxy model for compressors of the QWEN-3 family and observe the same scaling behavior (Appendix E.1.5).

Compressor scaling effects are consistent across prompt conditions. A natural concern is whether our scaling results depend on specific prompt formatting. To test robustness, we instructed compressor models to output 3, 6, or 9 sentences, varying conciseness levels. Scaling behavior in accuracy, compression output size, FLOPs-per-generation, MI, and bit efficiency remained consistent across all conciseness instructions on both LONGHEALTH and FINANCEBENCH (Figures 5, 16). The relative improvements from larger compressors persist regardless of prompted compression output length, confirming that model capacity drives the observed efficiency gains.

3.3 INFORMATION RATE CORRELATES STRONGLY WITH DOWNSTREAM PERFORMANCE

Mutual information and bit-efficiency are proxies for system performance. Information rate (bit-efficiency) is closely related to distortion (1 − accuracy). Motivated by the classical form of the rate-distortion function for a independent Gaussian source X, we fit decaying exponential functions to the rate-distortion data (Appendix B.5). This fit characterizes the correlation between information rate and distortion and corroborates our previous finding that scaling predictors beyond 70B yields only marginal improvements in distortion (Figure 6, left).

Furthermore, our results in Figure 6 (right) reveal that mutual information is also strongly correlated with perplexity ($r = -0.84$, $R^2 = 0.71$) for extractive tasks on FINEWEB (setup detailed in Appendix D.1.5).

Predictors do not prefer compressors of the same family. Further rate-distortion analysis across QWEN-2.5 and LLAMA models reveal that distortion is primarily dependent on model family and size. Crucially, predictors do not perform better when paired with compressors of the same family (Figure 19).

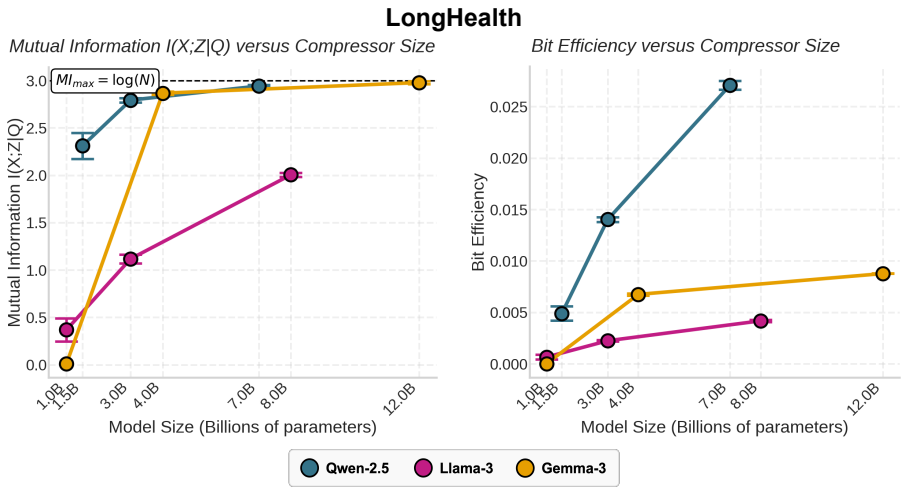

Figure 4: **Larger compressors generate outputs that carry more information about their inputs (conditioned on the query) on LONGHEALTH.** We scale compressor model size and estimate the **(Left)** mutual information, and **(Right)** bit efficiency (bits of mutual information per token; higher is better) carried by their outputs. Larger compressor model sizes compress documents with higher mutual information and bit efficiency. The black dotted line represents the theoretical maximum of the mutual information estimator at the natural logarithm $\log(N)$, where N is the number of documents mutual information is computed across. We find consistent trends on FINANCEBENCH (Appendix E.1.6) and QASPER (Appendix E.1.1).

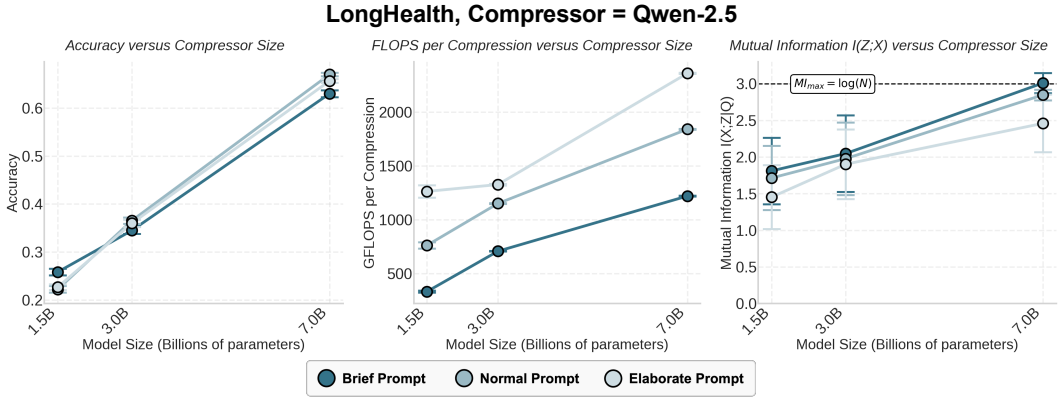

Figure 5: **Scaling behavior of compressor model size hold across instructed conciseness (COMPRESSOR = QWEN-2.5).** We ablate over different levels of compression conciseness by varying the compression prompt instructions. We measure **(Left)** accuracy, **(Middle)** GFLOPs-per-generation, and estimate **(Right)** mutual information. We find that accuracy and mutual information are largely unaffected by conciseness instructions. Compressors instructed to be more concise are more token-efficient, and thus compute-efficient. Trends in accuracy, compute cost, and mutual information as we scale compressor hold across conciseness constraints. Appendix E.1.7 shows analogous results on FINANCEBENCH.

3.4 WHICH KNOBS TO TURN?

To guide practical system design, we analyze which components of the compression-prediction pipeline most strongly drive downstream QA accuracy. We fit a logistic regression predicting binary correctness on LONGHEALTH and FINANCEBENCH using the features specified in Appendix D.4.

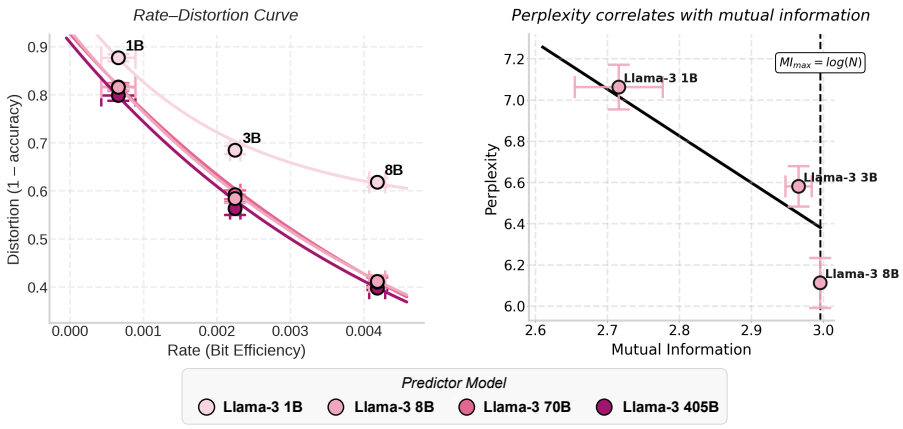

Figure 6: **Mutual information and bit efficiency correlate strongly with downstream performance. (Left)** We vary both predictor and compressor model in the compression-prediction workflow and measure the distortion on the y-axis and estimate the rate on the x-axis. We plot the resulting rate-distortion curves across predictor sizes 1B, 8B, 70B, and 405B for LLAMA compressors on LONGHEALTH. The colored lines show fitted exponential-decay functions. **(Right)** We measure perplexity and mutual information on compressions generated by LLAMA compressors on FINEWEB. The black line shows a fitted linear function ($r = -0.84$, $R^2 = 0.71$).

The compressors we consider are QWEN-2.5 and LLAMA models and predictors are LLAMA models of sizes 1B, 8B, 70B, and 405B.

Our analysis reveals that compressor model family is the most important factor (Figure 17) with QWEN-2.5 compressors outperforming LLAMA. Additionally, scaling the compressor LM matters substantially more than scaling the predictor LM, confirming previous findings in Section 3.1.

3.5 SCALING DEEP RESEARCH

We evaluate our compression-prediction framework on open-domain "Deep Research" workflows, where a predictor LM decomposes research tasks into subtasks and aggregates compressor outputs into final reports. We use DEEPRESEARCH BENCH (Du et al., 2025), which assesses system performance across four dimensions: Comprehensiveness, Depth, Instruction-following, and Readability. These four dimensions form a quantitative *RACE* (Reference-based Adaptive Criteria Evaluation) score. We measure cost per task based on current API prices (as of August 2025), which are time-sensitive, but serve as good approximations for relative cost differences between models. We vary predictor sizes across the LLAMA family and compressor sizes across the QWEN-2.5 family. Full experimental details are in Appendix D.6.

Larger predictor models consistently improve *RACE* scores, while larger compressors provide substantial performance gains at minimal additional API costs (Figure 7).

As a baseline, we evaluate the results of providing uncompressed web search data to a GPT-4O predictor. A QWEN-2.5-14B compressor paired with a GPT-4O predictor achieves 2.3% higher RACE scores at only 28.1% of the API cost compared to the uncompressed baseline. We detail further findings in our scaling experiments in Appendix E.3.

4 DISCUSSION

We establish an information-theoretic framework for compressor-predictor systems to determine how model selection and scaling affect compute efficiency. Our findings come with important limitations: at the 1–3B model scale, our MI estimator relies on proxy models and log probabilities, introducing potential variance and biases. Furthermore, we primarily focus on GPT-style non-reasoning models with single-round communication, limiting generalizability to reasoning-augmented models or iterative multi-agent workflows. While we provide initial results for reasoning and mixture-

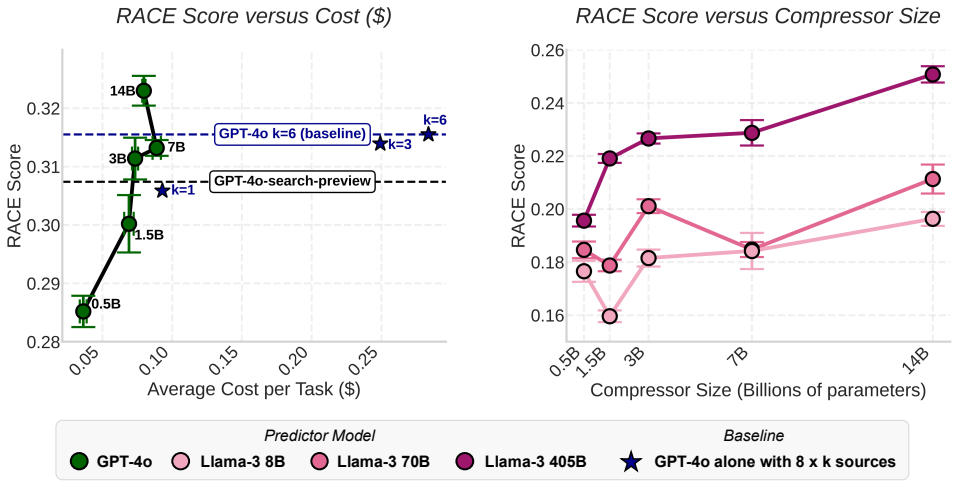

Figure 7: **Deep Research Scaling Results. (Left)** RACE score versus average task cost when using GPT-4O as a predictor with QWEN-2.5 compressors of varying sizes. Costs are based on GPT-4O API rates (August 2025: $2.50/1M input tokens, $10.00/1M output tokens). Larger compressors improve performance with minimal cost increases. For reference, we include GPT-4O results without compression and also for the GPT-4O-SEARCH-PREVIEW model. **(Right)** RACE scores for different QWEN-2.5 compressor sizes (0.5B–14B) under three LLAMA predictors (8B, 70B, 405B).

of-experts models, future work should extend the information-theoretic analysis to a wider range of model families and evaluate reasoning traces in a more principled manner.

Several research directions warrant investigation. Mutual information estimation for LM outputs remains challenging, though alternative estimators like INFONCE (Aitchison & Ganev, 2021) offer promising solutions. Information-theoretic principles could guide compressor routing strategies and fallback decisions for remote full-context processing. Training objectives based on rate-distortion analysis represent another avenue to optimize compressor-predictor communication. While we define compression as summarization in this work, compression can also be found in other agentic-system workflows, such as structured extraction and function-call generation. FLOPs-per-generation is an intuitive measurement of compute cost. However, device-specific efficiency optimizations are crucial in real-world deployment settings and warrant further analysis. Finally, mixture-of-experts (MoE) models (Fedus et al., 2022) may exhibit different scaling behaviors since their compute cost depends on activated experts rather than total parameter count.

Overall, we distill our findings into four principles for agentic system design:

Principles for Agentic System Design

- **Compressors can be scaled at a sublinear computational cost.** Larger compressor LMs are more accurate, concise, and information-efficient. Since larger models are more concise, FLOPs-per-generation scale sublinearly as a function of model size.
- **"Front-load" compute into local compressors to reduce remote costs.** Scaling compressors is more effective than scaling predictors. By running larger compressors on-device, we can reduce predictor serving costs on the cloud.
- **Optimize for information density.** Mutual information serves as a task-agnostic indicator of compression quality and is tightly linked to downstream performance and perplexity.
- **Expect model family to differ in scaling trends.** Choice of compressor and predictor model family yields offsets in rate-distortion curves and scaling effects. QWEN-2.5 compressors scale more compute-efficiently than LLAMA and GEMMA-3. QWEN-2.5 predictors yield higher accuracies than LLAMA.

We utilized AI tools to assist with code implementation and manuscript proofreading.

ETHICS STATEMENT

Regarding fairness and accessibility, our recommendation to "front-load" computation into local compressors may create barriers for researchers with limited hardware resources, potentially exacerbating inequalities in AI access despite reducing cloud API costs by 74%. The compression techniques process documents through multiple model stages, raising privacy concerns about information retention in compressed representations, especially when handling sensitive data. Our efficiency improvements could accelerate broader deployment of agentic systems with both beneficial and harmful applications, while the environmental impact of encouraging larger local model deployment (up to 27B parameters) requires careful consideration against potential increases in aggregate energy consumption. We encourage practitioners to implement appropriate privacy safeguards and consider the dual-use implications of these compression-prediction architectures as they become more prevalent.

REPRODUCIBILITY STATEMENT

We provide comprehensive implementation details and experimental specifications throughout the paper and appendices. Section 2.2 contains the complete derivation and implementation of our mutual information estimator, while Appendix B.1 details the FLOPs computation methodology for dense transformer models. All experimental configurations, including model selections, hyperparameters, and prompt templates, are specified in Appendices B and D. Dataset construction procedures are documented in Appendix D.1, with specific sampling criteria for each of the four datasets, as well as prompt templates to construct synthetic QA and generation tasks. All models used to generate synthetic tasks and answers are detailed in Appendix D.1. The Deep Research experimental setup is fully described in Appendix D.6, including the complete workflow implementation and evaluation framework, as well as prompt templates for compressor and predictor LMs. All experiments run on $S = 5$ random seeds with reported standard errors, and we specify the exact model versions, inference parameters, and evaluation protocols used across all experiments. Rate-distortion analysis parameters and fitting procedures are detailed in Appendix B.5, while the generalized linear model specifications for meta-analysis are provided in Appendix D.2.

REFERENCES

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. Longhealth: A question answering benchmark with long clinical documents. *arXiv preprint arXiv:2401.14490*, 2024.

Laurence Aitchison and Stoil Ganev. Infonce is variational inference in a recognition parameterised model. *arXiv preprint arXiv:2107.02495*, 2021.

Anthropic. Claude 3.7 sonnet and claude code. https://www.anthropic.com/news/claude-3-7-sonnet, February 2025. Accessed: 2025-09-17.

Enes Arda and Aylin Yener. A rate-distortion framework for summarization. *arXiv preprint arXiv:2501.13100*, 2025.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more llm calls all you need? towards scaling laws of compound inference systems. *arXiv preprint arXiv:2403.02419*, 2024.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Thomas M. Cover and Joy A. Thomas. *Rate Distortion Theory*, chapter 10, pp. 301–346. John Wiley Sons, Ltd, 2005. ISBN 9780471748823. doi: https://doi.org/10.1002/047174882X.ch10. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/047174882X.ch10.

Maxime Darrin, Philippe Formont, Jackie Cheung, and Pablo Piantanida. COSMIC: Mutual information for task-agnostic summarization evaluation. 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. 12696–12717, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.686. URL https://aclanthology.org/2024.acl-long.686/.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. 2021.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint*, 2025.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, et al. Cartridges: Lightweight and general-purpose long context representations via self-study. *arXiv preprint arXiv:2506.06266*, 2025.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625 – 630, 2024. URL https://api.semanticscholar.org/CorpusID:270615909.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Jan-Philipp Fränken, Eric Zelikman, Rafael Rafailov, Kanishk Gandhi, Tobias Gerstenberg, and Noah Goodman. Self-supervised alignment with mutual information: Learning to follow principles without preference labels. *Advances in Neural Information Processing Systems*, 37:61328–61371, 2024.

Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings (eds.), *Intelligent Agents III Agent Theories, Architectures, and Languages*, pp. 21–35, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. ISBN 978-3-540-68057-4.

Ziv Goldfeld and Yury Polyanskiy. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1):19–38, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279, 2023.

Jeremy Hadfield, Barry Zhang, Kenneth Lien, Florian Scholz, Jeremy Fox, and Daniel Ford. How we built our multi-agent research system, 2025. URL https://www.anthropic.com/engineering/built-multi-agent-research-system. Accessed July 24, 2025.

Dongge Han, Camille Couturier, Daniel Madrigal Diaz, Xuchao Zhang, Victor Rühle, and Saravan Rajmohan. Legomem: Modular procedural memory for multi-agent llm systems for workflow automation, 2025. URL https://arxiv.org/abs/2510.04851.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. *Advances in neural information processing systems*, 35:30016–30030, 2022.

Kelly Hong, Anton Troynikov, and Jeff Huber. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma, July 2025. URL https://research.trychroma.com/context-rot.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In *International conference on machine learning*, pp. 16049–16096. PMLR, 2023.

Andreas Kirsch, Clare Lyle, and Yarin Gal. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning. *arXiv preprint arXiv:2003.12537*, 2020.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Yiren Lu. How much vram do i need for llm inference? *Modal Blog*, September 2024. URL https://modal.com/blog/how-much-vram-need-inference. Online; accessed 24 September 2025.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*, 2024.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pp. 875–884. PMLR, 2020.

Kevin P Murphy. *Probabilistic machine learning: Advanced topics*. MIT press, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Avanika Narayan, Dan Biderman, Sabri Eyuboglu, Avner May, Scott Linderman, James Zou, and Christopher Re. Minions: Cost-efficient collaboration between on-device and cloud language models. *arXiv preprint arXiv:2502.15964*, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, et al. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 06 2003. ISSN 0899-7667. doi: 10.1162/089976603321780272. URL https://doi.org/10.1162/089976603321780272.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pp. 5171–5180. PMLR, 2019.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Guha, E Kelly Buchanan, Mayee Chen, Neel Guha, Christopher Ré, et al. Archon: An architecture search framework for inference-time techniques. *arXiv preprint arXiv:2409.15254*, 2024.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning*. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, dec 2019. doi: 10.1088/1742-5468/ab3985. URL https://doi.org/10.1088/1742-5468/ab3985.

Erik Schluntz and Barry Zhang. Building effective ai agents, 2025. URL https://www.anthropic.com/engineering/building-effective-agents. Accessed July 24, 2025.

Chen Shani, Dan Jurafsky, Yann LeCun, and Ravid Shwartz-Ziv. From tokens to thoughts: How llms and humans trade compression for meaning. *arXiv preprint arXiv:2505.17117*, 2025.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face, 2023. URL https://arxiv.org/abs/2303.17580.

Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(93)90034-9. URL https://www.sciencedirect.com/science/article/pii/0004370293900349.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pp. 1–5. Ieee, 2015.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329*, 2020.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024.

Charles Westphal, Stephen Hailes, and Mirco Musolesi. A generalized information bottleneck theory of deep learning. *arXiv preprint arXiv:2509.26327*, 2025.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Weizhi Zhang, Yangning Li, Yuanchen Bei, Junyu Luo, Guancheng Wan, Liangwei Yang, Chenxuan Xie, Yuyao Yang, Wei-Chieh Huang, Chunyu Miao, et al. From web search towards agentic deep research: Incentivizing search with reasoning agents. *arXiv preprint arXiv:2506.18959*, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37: 62557–62583, 2024.

A Extended Related Work

Deep Research Over the past year, large-scale Deep Research systems have been popularized and adopted by frontier labs in industry such as OpenAI, Anthropic, and xAI. These systems commonly have an asymmetric setup, where a high-capacity *predictor* LM decomposes the user query into subtasks that are executed by *compressor* models in parallel (Schluntz & Zhang, 2025). The results of these subtasks are synthesized into one answer that is commonly presented to the user as a comprehensive *research report*. In practice, compressor models can range from frontier models (Hadfield et al., 2025; Schluntz & Zhang, 2025) to local small LMs (Narayan et al., 2025). Recent works have been centered around establishing evaluation benchmarks to compare different agentic *Deep Research* systems (Du et al., 2025; Guha et al., 2023; Nakano et al., 2021). These benchmarks focus on measuring the output quality and downstream utility of generated research reports.

B Extended Description of Methods

In this section, we provide a more in-depth explanation and derivation of our information-theoretic approach and problem setup.

B.1 Compute Cost of Dense LMs

We measure compute cost of each compressor/predictor LM call by the number of FLOPs per token in each forward pass through our dense transformer-based LMs as

$$C_{\text{dense}} \approx 2N_{\text{params}} + 2n_{\text{layer}}n_{\text{ctx}}d_{\text{attn}},$$

with model size N_{params}, number of input context tokens n_{ctx}, number of layers n_{layer}, and number of attention heads per layer d_{attn} (Kaplan et al., 2020). We observe that FLOPs-per-token-generated for dense models grows roughly linearly with model size.

B.2 Theoretical Analysis: Lower Bound on $I(X;Y)$ via Fano's Inequality

In addition to estimating $I(X;Z)$, we provide a lower bound on $I(X;Y)$ for categorical Y variables (e.g., multiple-choice QA in LongHealth). Using Fano's inequality, we are able to connect downstream prediction accuracy to $I(X;Y)$ without the need for any LM logits.

Let $|\mathcal{Y}|$ denote the number of possible answer choices. For classification tasks, Fano's inequality says that

$$H(Y \mid X) \leq H_b(e) + P(e) + \log(|\mathcal{Y}| - 1),$$

where $P(e) = 1 - P(\hat{Y} = Y)$ is the probability of error under the true joint distribution of (X, Y). $H_b(e)$ is the binary entropy function:

$$H_b(e) = -P(e)\log P(e) - (1 - P(e))\log(1 - P(e))$$

Since

$$I(X;Y) = H(Y) - H(Y \mid X),$$

we find the lower bound:

$$I(X;Y) \geq H(Y) - H_b(e) - P(e)\log(|\mathcal{Y}| - 1),$$

If the ground-truth multiple-choice answers are uniformly distributed, we can approximate

$$H(Y) \approx \log|\mathcal{Y}|.$$

This estimator allows us to compute the lower bound to $I(X;Y)$ for categorical Y without needing to access LM logits, offering a more practical approach to evaluating $I(X;Y)$ given only the error rate.

B.3 Theoretical Analysis: Bounds of Mutual Information

Theorem 1. *Mutual information between X and Z is upper bounded by*

$$I(X;Z) \leq \log \min\left(|X|, |Z|\right).$$

Proof. Recall the definition of mutual information

$$\begin{aligned} I(X;Z) &= H(X) - H(X|Z) \\ &= H(Z) - H(Z|X), \end{aligned}$$

where $H(X)$ and $H(Z)$ are Shannon entropies of input documents X and output compressions Z. Starting from the first identity and the definition of conditional entropy,

$$\begin{aligned} I(X;Z) &= H(X) - H(X|Z), \qquad H(X|Z) \geq 0 \\ &\leq H(X). \end{aligned}$$

Similarly, starting from the second identity,

$$\begin{aligned} I(X;Z) &= H(Z) - H(Z|X), \qquad H(Z|X) \geq 0 \\ &\leq H(Z). \end{aligned}$$

We combine the two upper bounds of mutual information between X and Z

$$I(X;Z) \leq min\{H(X), H(Z)\} \leq \log \min\left(|X|, |Z|\right).$$

□

B.4 Theoretical Analysis: Bounds of Monte Carlo Estimator

Theorem 2. *The Monte-Carlo estimator of mutual information between X and Z is upper bounded by*

$$\hat{I}(X;Z) \leq \log N,$$

where N is defined as the number of contexts sampled from X.

Proof. Consider the Monte-Carlo estimator of mutual information

$$\begin{aligned} \hat{I}(X;Z) &= \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\log p(z_{ij}|x_i) - \log \left(\frac{1}{N} \sum_{l=1}^{N} p(z_{ij}|x_l) \right) \right] \\ &= \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\log p(z_{ij}|x_i) + \log N - \log \left(\sum_{l=1}^{N} p(z_{ij}|x_l) \right) \right]. \end{aligned}$$

For any fixed summary z_{ij} and context x_l, we have

$$\begin{aligned} \sum_{l=1}^{N} p(z_{ij}|x_l) &\geq \max_{l} \; p(z_{ij}|x_l) \\ &\geq p(z_{ij}|x_i). \end{aligned}$$

Plugging this into the estimator bounds yields the upper bound

$$\begin{aligned} \hat{I}(X;Z) &= \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\log p(z_{ij}|x_i) + \log N - \log \left(\sum_{l=1}^{N} p(z_{ij}|x_l) \right) \right] \\ &\leq \log N. \end{aligned}$$

To see when the bound is tight at $p(z_{ij}|x_i) \gg p(z_{ij}|x_l)\ \forall l \neq i$, write the denominator using a constant c_{ij}

$$\sum_{l=1}^{N} p(z_{ij}|x_l) = e^{c_{ij}} \sum_{l=1}^{N} e^{\log p(z_{ij}|x_l) - c_{ij}}.$$

Choose $c_{ij} = \log p(z_{ij}|x_i)$, this gives us

$$\hat{I}(X;Z) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\log p(z_{ij}|x_i) + \log N - c_{ij} - \log\left(\sum_{l=1}^{N} e^{\log p(z_{ij}|x_l) - c_{ij}}\right)\right]$$
$$= \log N - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\log\left(\sum_{l=1}^{N} e^{\log p(z_{ij}|x_l) - \log p(z_{ij}|x_i)}\right)\right],$$

Since $p(z_{ij}|x_i) \gg p(z_{ij}|x_l), \forall l \neq i$,

$$\hat{I}(X;Z) \approx \log N + \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}[-\log(1+\varepsilon)] \qquad (\varepsilon \approx 0)$$
$$\approx \log N.$$

Thus $\log N$ is a tight upper bound. □

B.5 Rate-Distortion-Theory

Assume X to be an independent Gaussian random variable with variance $\sigma^2(X)$, the canonical rate-distortion function is

$$R(D) = \begin{cases} \frac{1}{2}\log\left(\frac{\sigma^2(X)}{D}\right), & 0 \leq D \leq \sigma^2(X) \\ 0, & D > \sigma^2(X). \end{cases}$$

We illustrate rate-distortion curves as distortion D (how much accuracy is lost in communication) versus rate R (how many bits spent encoding data). Inverting the expression for the rate-distortion function gives (Cover & Thomas, 2005)

$$D_{Gaussian}(R) = \sigma^2\, 2^{-2R}$$
$$= \sigma^2\, e^{-\,2\ln 2\, R}$$
$$= C\, e^{-bR}, \qquad \text{with} \quad C = \sigma^2,\ b = 2\ln(2).$$

The Gaussian rate-distortion function does not describe LM data distributions. However, it serves as a closed-form model that captures the qualitative shape of typical rate-distortion curves. We use the Gaussian rate-distortion function as a simplified model to qualitatively compare different predictors.

In practice, we treat C and b as function parameters to account for unknown variance and modeling noise. LM compression-prediction systems often exhibit a non-zero distortion floor (e.g. imperfect LM judge, label noise, predictor expressive power), which we account for through offset D_0. D_0 is a lower bound of the distortion in the system as rate (bit efficiency) increases,

$$D(R) = C\, e^{-bR} + D_0.$$

We fit exponential decay functions to the rate-distortion curves based on the least-squares estimates $(\hat{C}, \hat{b}, \hat{D}_0)$.

C Prompts

C.1 Compressor Model Prompts

We use the following prompt templates to compress the raw context documents on LongHealth, FinanceBench, QASPER, FineWeb, and each chat conversation on WildChat:

Query-Specific Base Compression Prompt Template

```
Summarize the following text to include ONLY information needed to answer the question.
Extract the key points relevant to the question.
DO NOT ANSWER THE QUESTION DIRECTLY.
```

```
Question:
{query}

Text:
{text}

Your summary (make sure to include all important details / background information related to the
*question*.  **DO NOT ANSWER THE QUESTION**)
```

Memory Construction/Compression Prompt Template (WildChat)

```
You are a memory compression assistant, tasked with summarizing a chat conversation.
Produce a summary that preserves all details that could be useful as memory for a language model.  DO NOT
invent any information.

CHAT:
{conversation}

Your summary (Just plain text, no formatting.)
```

Query-Agnostic Compression Prompt Template (FineWeb)

```
Summarize the following text and produce a summary that preserves all details that could be needed to
answer likely questions about the text.  Do NOT invent facts.

Do NOT answer any question; just summarize potential answer-bearing info.

Text:
{text}

Your summary (make sure to include all important details / background information related.  Just plain
text, no formatting.)
```

C.2 Predictor Model Prompts

Given the compressor output, we answer extractive QA tasks on LongHealth, FinanceBench, QASPER, WildChat, and FineWeb, and creative tasks on FineWeb using the following prompt templates:

Base Prediction Prompt Template

```
Please answer the following question based on the provided summary.
Question:
{query}

Summary:
{summary}

Please respond in the following JSON format:  <briefly think about the information you have and the
question you need to answer>

{{
   "explanation":  "<brief explanation of the answer.  explain how you arrived at the answer.  1-2
sentences>",
   "answer":  "<your final answer>"
}}

Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

WildChat Prediction Prompt Template

```
Please answer the following question based on the provided chat memory.

Question:
{query}

Memory:
{memory}

Please respond in the following JSON format:  <briefly think about the information you have and the
question you need to answer>

{{
   "answer":  "<your final answer>"
}}

Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

FineWeb Prediction Prompt Template (Extractive)

```
Please answer the following question based on the provided {context_type}.

Question:
{query}

{context_type}:
{summary}

Please respond in the following JSON format:
<briefly think about the information you have and the question you need to answer>

{{
   "answer": "<your final answer>"
}}

Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

FineWeb Prediction Prompt Template (Creative)

```
Please do the following based on the provided {context_type}.

Task: {query}

{context_type}:
{summary}

Please respond in the following JSON format:
<briefly think about the information you have and the question you need to answer>

{{
   "answer": "<your final answer>"
}}

Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

C.3 DEEPRESEARCH PROMPTS

The following prompt templates were used sequentially as the backbone for our compressor-predictor Deep Research workflow.

DeepResearch Query Generation Prompt Template (Predictor)

```
You are a research supervisor tasked with comprehensively exploring a research topic. Use a strategic,
top-down approach to design your research.

Research Topic: {query}

**PHASE 1: RESEARCH PLANNING**
First, analyze this research topic and create a comprehensive research plan. Consider:
- What are the key areas that must be investigated to fully understand this topic?
- What specific objectives will guide your research?
- How do different aspects of this topic relate to each other?
- What types of information will be most valuable for a complete analysis?
- What is the logical flow for presenting findings?

**PHASE 2: STRATEGIC QUERY GENERATION**
Based on your research plan, generate EXACTLY 8 different search queries that together will provide
comprehensive coverage of this topic. Each query should serve a specific strategic purpose in your
overall research architecture.

For each search query, provide a specific sub-task/question that explains how it serves your research
plan.

Return your response in this exact JSON format:
{{
   "research_plan": "Your comprehensive research architecture and strategic objectives for investigating
this topic. Explain the key areas to investigate, how they relate, and the logical structure for
analysis.",
   "queries": [
      {{
         "search_query": "specific search terms optimized for Google",
         "sub_task": "What specific question does this query address and how does it serve the research
plan?"
      }},
      {{
         "search_query": "second strategic search query",
         "sub_task": "What does this query aim to discover and how does it fit the research
   architecture?"
      }},
      {{
         "search_query": "third targeted search query",
         "sub_task": "What aspect does this explore and why is it essential to the research plan?"
      }},
```

```
      {{
         "search_query":  "fourth strategic search query",
         "sub_task":  "What question does this answer and how does it complement other queries?"
      }},
      {{
         "search_query":  "fifth focused search query",
         "sub_task":  "What aspect does this cover and how does it build on previous queries?"
      }},
      {{
         "search_query":  "sixth comprehensive search query",
         "sub_task":  "What additional dimension does this explore and why is it crucial?"
      }},
      {{
         "search_query":  "seventh strategic search query",
         "sub_task":  "What specific gap does this fill in the research architecture?"
      }},
       {{
         "search_query":  "eighth concluding search query",
         "sub_task":  "What final aspect does this cover and how does it complete the comprehensive
research?"
      }}
   ],
   "synthesis_strategy":  "Detailed strategy for combining findings from all 8 queries based on your
research plan.  Explain how the information will be structured, what relationships will be highlighted,
and how the final analysis will be organized to maximize comprehensiveness and insight."
}}

**Strategic Guidelines:**
1.  Each search query should be 3-8 well-chosen keywords targeted for your specific research objectives
2.  Design queries to serve complementary roles in your research architecture (not just generic
dimensions)
3.  Ensure queries are strategically coordinated to provide comprehensive topic coverage
4.  Each sub-task should explain how the query serves your overall research plan
5.  Create a synthesis strategy that reflects your planned research structure

**Research Focus Areas to Consider:**
- Foundational understanding and current state
- Key challenges, problems, or limitations
- Solutions, methodologies, and best practices
- Evidence, data, and empirical findings
- Future trends, developments, and implications
- Multiple perspectives and stakeholder viewpoints

CRITICAL: You must return ONLY the JSON object.  Do NOT format it as a code block with ```json``` or
any other markdown formatting.  Return the raw JSON object directly.
```

DeepResearch Synthesis Prompt Template (Predictor)

```
You are tasked with creating a comprehensive, high-quality research report for a DeepResearch task.  You
have extensive research findings below - use ALL of them to create a detailed, thorough analysis.

**Original Research Task:** {original_task}

**Research Plan:** {research_plan}

**Research Findings:**
{qa_pairs}

**Synthesis Strategy:** {synthesis_strategy}

**COMPREHENSIVE INFORMATION UTILIZATION - ALL SOURCES REQUIRED:**
You must systematically work through ALL the provided research findings above.  Do not selectively use
only some information - your report must demonstrate that you have reviewed and integrated ALL relevant
details, data points, examples, and perspectives from every query and source provided.

**REPORT STRUCTURE AND REQUIREMENTS:**
1.  **Detailed Background Context** - Provide extensive background and context
2.  **Comprehensive Analysis** - Multiple detailed sections covering all aspects
3.  **Extensive Evidence Integration** - Use specific examples, data, quotes from ALL sources
4.  **Thorough Implications Discussion** - Detailed analysis of implications and significance
5.  **Complete Conclusions** - Comprehensive conclusions and future research directions

**WRITING REQUIREMENTS FOR HIGH QUALITY:**
- Write detailed explanations, not brief summaries
- Include extensive examples and case studies from the research
- Provide comprehensive background and context for every major point
- Use all statistical data, quotes, and specific details from the research findings
- Elaborate on implications, significance, and broader connections
- Include detailed analysis of methodologies, approaches, and frameworks mentioned
- Discuss limitations, challenges, and areas for further research extensively

Create a thorough academic research report that:
- Uses extensive detail and comprehensive analysis throughout
- Integrates ALL findings with detailed explanations and context
- Provides comprehensive coverage with extensive supporting evidence
- Includes detailed discussion of all relevant aspects and implications
- Demonstrates mastery of the subject through thorough, detailed analysis

**FINAL REQUIREMENT:**
Your response must be substantial and comprehensive.  Write extensively with exhaustive detail,
```

```
comprehensive analysis, and complete utilization of all research findings.  Provide truly comprehensive
coverage of the topic that demonstrates thorough understanding and integration of all available research.
```

DeepResearch Source Summarization Prompt Template (Compressor)

```
Your job is to extract detailed, specific information from the following content to support comprehensive
research analysis.

**Main Research Query:** {query}

**Specific Sub-task/Question:** {sub_task}

## Content
{content}

**EXTRACTION REQUIREMENTS: Provide a detailed and comprehensive extraction that captures:**

**Factual Information:**
- Specific numbers, statistics, percentages, and quantitative data
- Dates, timelines, and chronological information
- Names of people, organizations, companies, and institutions
- Geographic locations, regions, and jurisdictions
- Technical specifications, measurements, and benchmarks

**Detailed Examples and Evidence:**
- Concrete case studies and real-world examples
- Specific research findings and study results
- Direct quotes and expert opinions
- Policy details and regulatory information
- Implementation details and methodologies

**Comprehensive Coverage:**
- Key facts directly relevant to both the main query AND the specific sub-task
- Important concepts, definitions, and explanations
- Cause-and-effect relationships and underlying mechanisms
- Trends, patterns, and developments over time
- Challenges, limitations, and problem areas identified

**Analytical Insights:**
- Implications and significance of the information
- Relationships between different data points
- Comparative information and benchmarks
- Future projections and forecasted trends
- Expert assessments and professional evaluations

Focus on depth and specificity while maintaining clarity.  Extract comprehensive, specific information
with extensive detail, numbers, examples, and evidence.  Do not provide brief summaries - ensure your
extraction is thorough and substantial.  Extract information that would be valuable for creating a
comprehensive research report.  Pay special attention to information that directly addresses the sub-task
question.

Return your extraction in JSON format with these fields:
- "explanation":  Your detailed extraction of specific information, facts, data, examples, and evidence
with extensive detail
- "answer":  "relevant" if this content contains information relevant to the query and sub-task, "not
relevant" otherwise

CRITICAL JSON FORMATTING RULES:
- Replace all double quotes (") inside text with single quotes (')
- Replace all newlines with spaces
- Ensure the JSON is valid and parseable
- Do NOT use line breaks within the JSON fields

Example format:
{{"explanation":  "Your detailed extraction with specific facts, numbers, examples, and evidence using
single quotes for any nested quotes", "answer":  "relevant"}}

CRITICAL: You must return ONLY the JSON object.  Do NOT format it as a code block with ```json``` or
any other markdown formatting.  Return the raw JSON object directly.
```

D Extended Experimental Setup

Here, we further explain the construction of our datasets, choice of compressor and predictor models, and Deep Research experimental setup.

D.1 Datasets

D.1.1 LongHealth

LongHealth is a QA benchmark composed of 20 patient cases and clinical documents. Each of the 20 patients has a set of 20 multiple-choice questions about their personal records each ranging from 5,090 to 6,754 words (Adams et al., 2024). The original LongHealth benchmark is a multiple-

choice QA task. To more closely mirror our QA setups in the remaining three datasets, we remove the multiple-choice options in the prediction step. We subsample $N = 20$ documents and queries and generate $M = 20$ compressions for each of the problem contexts.

D.1.2 FINANCEBENCH

FINANCEBENCH is a long-context QA benchmark on 150 financial reports. Each financial report ranges from 1,923 to 517,224 tokens, with an average length of 119,968 tokens (Islam et al., 2023). We filter the original FINANCEBENCH dataset to only include samples with answer evidence at one location in the text. We slice a text segment of 21,500 tokens centered around the evidence as the raw document context. We subsample $N = 20$ problems and generate $M = 20$ compressions for each of the problem contexts.

D.1.3 QASPER

QASPER is a non-synthetic QA benchmark consisting of 1,585 scientific research papers in Natural Language Processing and 5,049 human-written questions about the content of the paper. Each scientific paper has up to 16,000 tokens (Dasigi et al., 2021). The questions are written by an NLP practitioner prior to reading the full paper, so QA evidence can be dispersed across multiple parts of the document. We subsample $N = 20$ documents and queries and generate $M = 20$ compressions for each of the documents. All experiments are run with $S = 3$ random seeds.

D.1.4 WILDCHAT

Our motivation in constructing a chat memory dataset is to simulate real-world memory systems that require models to integrate information across multiple previous interactions. Queries could build upon multiple previous exchanges, or individual isolated chats. In the original WILDCHAT dataset consisting of 837,989 multi-turn ChatGPT chats, each chat conversation exists as a standalone sample. We subsample $D = 1000$ chat conversations with between 4 and 8 turns to construct our dataset. The dataset construction process is as follows:

1. **User Construction:** We construct synthetic users by grouping 10 chat samples to each user (total $N = 100$ users).
2. **QA Generation:** We format each of the 10 chat conversations and provide GPT-4O-MINI with all full chat conversations along with the QA prompt to generate a question unique to each user that has not appeared in its chat history.

QA Prompt Template

```
You are a data generation assistant, tasked with building a benchmark that evaluates the memory
capabilities of a language model.
You will be provided a list of previous chat conversations.  Your goal is to generate a new synthetic
query that has not appeared in previous chats, but nevertheless benefits from the information in previous
chats.

CHATS:
{chats}

Generate a new synthetic query that has not appeared in previous chats, but nevertheless benefits from
the information that has appeared in previous chats.
Do not generate a RAG query about existing data in the chats, but rather a new query that could leverage
existing chat information as **memory**.

Please respond in the following JSON format:  <briefly think about the information you have and the
question you can generate from it>

{{
   "question":  "<question>",
   "answer":  "<answer>"
}}
Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

D.1.5 FINEWEB

The FINEWEB dataset contains an extensive set of web pages since 2013. At the time of writing, the dataset includes 25.9 billion entries spanning from 2013 to 2025. To construct our subset of document and QA pairings, we collect $N = 100$ samples with between 15,000 and 28,000 tokens,

and ask GPT-4O-MINI to synthetically generate 2 extractive and 3 creative QAs based on the cleaned web data and QA prompt.

QA Prompt Template

```
You are generating synthetic question-answer (QA) pairs from a source text.

SOURCE_TEXT:
{context}

Use only information from SOURCE_TEXT. No hallucinated facts.
Generate five questions and answers:
- Question 1:  What is {{topic}} and why is it important?  (type = "qa")
- Question 2:  What is {{topic}} and how does it work?  (type = "qa")
- Question 3:  Write an email to a colleague summarizing the findings and take-aways.  (type =
"generation")
- Question 4:  Generate rap lyrics that teach the core concepts.  (type = "generation")
- Question 5:  Generate a poem about the topic.  (type = "generation")

Please respond in the following JSON format:  <briefly think about the information you have and questions
you can generate from it>

{{
   "questions":  [
      {{
         "topic":  "<topic 1>",
         "question":  "<question 1>",
         "answer":  "<answer 1>",
         "type":  "qa"
      }},
      {{
         "topic":  "<topic 2>",
         "question":  "<question 2>",
         "answer":  "<answer 2>",
         "type":  "qa"
      }},
      {{
         "topic":  "<topic 3>",
         "question":  "<question 3>",
         "answer":  "<answer 3>",
         "type":  "generation"
      }},
      {{
         "topic":  "<topic 4>",
         "question":  "<question 4>",
         "answer":  "<answer 4>",
         "type":  "generation"
      }},
      {{
         "topic":  "<topic 5>",
         "question":  "<question 5>",
         "answer":  "<answer 5>",
         "type":  "generation"
      }}
   ]
}}

Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

D.2 COMPRESSOR MODEL DETAILS

For the LLAMA-3 family, we use the models LLAMA-3.2-1B-INSTRUCT, LLAMA-3.2-3B-INSTRUCT, LLAMA-3.1-8B-INSTRUCT. For the QWEN-2.5 family, we use the models QWEN-2.5-1.5B-INSTRUCT, QWEN-2.5-3B-INSTRUCT, QWEN-2.5-7B-INSTRUCT. For the GEMMA-3 family, we use the models GEMMA-3-1B-IT, GEMMA-3-4B-IT, and GEMMA-3-12B-IT. Additionally, we evaluate QWEN-2.5-14B-INSTRUCT as compressor LM on WILDCHAT and FINEWEB.

For reasoning compression models, we use the models QWEN-3-4B and QWEN-3-8B, and for mixture-of-experts models, we use QWEN-3-30B-A3B.

All compressor model families are fine-tuned for instruction following. Compression outputs of at most 4096 tokens are generated with temperature of 0.7 for LLAMA-3, QWEN-2.5, and QWEN-3, and 1.0 for GEMMA-3.

D.3 PREDICTOR MODEL DETAILS

As predictor models we use GPT-4O, LLAMA-3.1-8B-INSTRUCT, LLAMA-3.3-70B-INSTRUCT, and LLAMA-3.1-405B-INSTRUCT. Predictor models generate with a temperature of 0.6 across all benchmarks and experiments.

D.4 Generalized Linear Model Analysis Setup

We fit a logistic regression that predicts binary correctness of a compression-prediction output on:

- Z-score normalized lengths of the input document, prediction output, and compression output,
- Z-score normalized predictor and compressor model size,
- Indicator $\mathbf{1}\{\text{Compressor=Qwen}\}$ for the compressor model family,

where the predictors are LLAMA-3 models of sizes 1B, 8B, 70B, and 405B.

D.5 Mutual Information Proxy Model Details

We choose a QWEN-2.5-7B proxy model when evaluating LLAMA compressors and a LLAMA-3.1-8B proxy model when evaluating QWEN-2.5 and GEMMA-3 compressors. We directly use internal log probabilities to estimate mutual information for QWEN-3 compressors.

D.6 Deep Research Setup Details

For our experiments, we randomly sample $N = 20$ English research tasks from the DEEPRESEARCH BENCH test set to ensure a representative evaluation across diverse research domains. We conduct 5 independent runs for each experimental configuration. This allows us to report mean performance with standard error bars, providing a robust assessment.

D.6.1 Full Deep Research Workflow Setup

In our Deep Research system setting, a predictor LM decomposes each research task into a collection of *(Query, Subtask)* pairs. Each pair consists of a targeted web search query with a natural language instruction that specifies how the retrieved evidence should be analyzed. The predictor then distributes these pairs to compressor LMs, which independently perform the searches in parallel. Compressor LMs process the retrieved content according to the subtask, and compress the results into summaries. The predictor then aggregates these summaries into a comprehensive research report. This setup is illustrated in Figure 8.

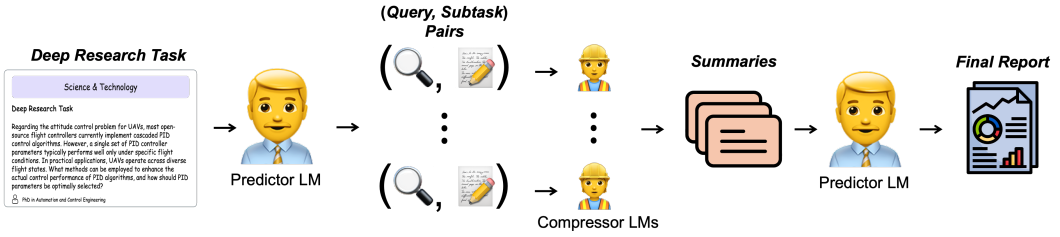

Figure 8: **Deep Research workflow.** A predictor LM decomposes a Deep Research task into *(Query, Subtask)* pairs, where each pair specifies a targeted web search and an associated analysis instruction. Compressor LMs work in parallel to retrieve evidence, process it according to the subtask, and compress the findings into concise summaries, which the predictor then aggregates into a final report.

We evaluate our system using the DEEPRESEARCH BENCH framework (Du et al., 2025), which assesses agent performance through four dimensions: Comprehensiveness, Depth, Instruction-following, and Readability. More specifically, we use *RACE* (Reference-based Adaptive Criteria Evaluation) scores to study the impact of model scale. The costs used in Figure 7 are based on GPT-4O API rates (Aug 2025: $2.50/1M input tokens, $10.00/1M output tokens). In addition, there is a constant SerpAPI web search cost of $0.12 for every task, which is not included in the figure.

D.6.2 Deep Research Compressor Model Details

We employ the Qwen-2.5-Instruct family of models as compressor LMs, ranging from 0.5B to 14B parameters. These models are hosted on Modal Labs using the SGLang inference framework, enabling free, high-throughput parallel inference. All compressor models use a temperature of 0.7 and a maximum output token limit of 2,000 tokens per response. The specific compressor models used are:

- Qwen-2.5-0.5B-Instruct: Smallest model for minimal compression overhead
- Qwen-2.5-1.5B-Instruct: Balance between efficiency and capability
- Qwen-2.5-3B-Instruct: Mid-range compression quality
- Qwen-2.5-7B-Instruct: Strong comprehension with moderate compute
- Qwen-2.5-14B-Instruct: Highest quality compression in our experiments

Each compressor independently processes the search results for its assigned *(Query, Subtask)* pair, extracting and compressing the relevant information according to the predictor's instructions. The compressed summaries from all compressors are then aggregated by the predictor into the final research report.

D.6.3 Deep Research Predictor Model Details

We evaluate four predictor models spanning different scales and providers:

- Llama-3.1-8B-Instruct: Entry-level predictor with basic task decomposition capabilities. Temperature set to 0.6, maximum output tokens of 4,000.
- Llama-3.1-70B-Instruct: Mid-tier predictor with improved reasoning and task planning. Temperature set to 0.6, maximum output tokens of 4,000.
- Llama-3.1-405B-Instruct: Large-scale predictor with advanced multi-step reasoning capabilities. Temperature set to 0.6, maximum output tokens of 4,000.
- GPT-4o: State-of-the-art commercial predictor serving as our performance upper bound. Temperature set to 0.6, maximum output tokens of 16,000 to accommodate comprehensive report generation.

All predictors use a slightly lower temperature (0.6) compared to compressors to ensure more consistent and structured task decomposition and report synthesis. The predictor is responsible for: (1) decomposing the research question into targeted queries, (2) formulating specific subtasks for each query, (3) distributing work to the compressor pool, and (4) synthesizing compressor outputs into a coherent final report.

A notable limitation is that the Llama family predictors are constrained to 4,000 output tokens, which can limit the comprehensiveness of their final research reports compared to GPT-4o's 16,000 token capacity. This constraint particularly affects the synthesis phase where the predictor must compile information from multiple compressor summaries into a cohesive report.

D.7 Compression Failure Modes

We attempt to understand how scaling compressor size qualitatively increases downstream QA accuracy. To do so, we categorize compressor errors into one of the three recurring failure modes.

Category 1: Incorrect answer. The compression contains an erroneous number or finding in an attempt to answer the question, often due to confusion between reports at different time points. Larger compressors reduce this due to their capacity to better model temporal dependencies.

Error Category 1: Incorrect Answer

Query: When was Mr. Alan Fisher first diagnosed with a heart disease?
Summary: Mr. Alan Fisher was first diagnosed with coronary artery disease on 10/1995, during a routine renal transplant biopsy due to progressive deterioration of kidney function. The initial biopsy showed isometric tubular epithelial vacuolization, which was associated with isometric microvesicular cytoplasmic transformation, exceeding 80%. This finding was highly suggestive of Chronic Allograft Nephropathy (CAN).
Prediction: 10/1995

Correct Answer: 1989
Evidence in context: **Other Diagnoses** [...] Coronary artery disease, CAD-3: - Previous anterior wall infarction in 1989, treated with [...]
Error diagnosis: Other diagnoses were made in 10/1995, but not coronary artery disease.

Category 2: No answer. The compression contains no information relevant to the query. All facts and findings in the compression are unrelated, leaving the question unaddressed. The compressor model fails to include any answer-bearing content.

Error Category 2: No answer

Query: What significant cardiac intervention did Mr. Fisher undergo in May 1995?
Summary: Mr. Alan Fisher was admitted on March 14, 2020, and underwent a urethrotomy interna and calculated intravenous antibiotic therapy with Meropenem. He was discharged with a urinary catheter inserted. However, he still experienced recurrent urinary tract infections. An additional treatment plan was developed to include antifungal therapy with Fluconazole. The patient was discharged on March 19, 2020.
Prediction: None
Correct Answer: PTCA + stent in RIVA
Evidence in context: **Other Diagnoses** [...] PTCA + stent in the right coronary artery (RIVA) in 05/1995
Error diagnosis: The cardiac intervention in May 1995 was embedded as background information in the section "Other Diagnoses" and not directly in the main medical findings. The model failed to understand the importance of historical interventions to answering the query.

Category 3: Details or parts of the findings are omitted from compression. The compression may correctly identify all medical findings or records required to answer the query, however not in enough detail to adequately answer the question. The compression may also capture only part of the medical findings required to answer the query, while omitting certain facts/information that are crucial to answering the full question. We often observe this when the compression contains many different patient/treatment facts that are irrelevant for answering the query, such that the critical findings are glossed over and "overly compressed".

Error Category 3: Omission of information/over-compression

Query: Where did Mrs. Done show metastases?
Summary: Mrs. Jane Done, born on 01.01.1966, had a history of metastatic malignant melanoma, with the most recent metastases being detected on 11/01/2020. Her treatment included a microsurgical resection of a right frontal tumor in 2018, a 2020 radiation therapy course, and immunotherapy with Nivolumab and Ipilimumab. In 2021, she was diagnosed with cervical lymph node metastasis, and underwent surgery followed by adjuvant therapy with pembrolizumab. In 2022, she presented with multiple metastases in her CT scan, which were confirmed on PET-CT. Her most recent treatments include six cycles of Vemurafenib, Cobimetinib, and Pembrolizumab.
Prediction: Right frontal tumor, cervical lymph nodes, and multiple unspecified locations
Correct Answer: Brain, lungs, liver, heart, lymph nodes, muscles, bone
Evidence in context: Microsurgical resection right frontal tumor [...] hemorrhaged right frontal metastasis from previously diagnosed malignant melanoma [...] multiple roundish subsolid nodules found bipulmonary [...] multiple hypodense lesions throughout both lobes, indicative of metastatic spread [...] concerning 2 cm mass abutting the lateral wall of the left ventricle raising the suspicion for cardiac metastasis [...] Cervical lymph node metastasis [...] a 2.5 cm mass identified within the left psoas muscle, consistent with muscular metastasis [...] lytic lesions involving the sternum and right 4th rib, consistent with osseous metastatic disease
Error diagnosis: The compressor selectively included only frequent metastasis mentions explicitly (brain, lymph nodes) in its summary while compressing numerous organ-specific findings in other parts of the context (lungs, liver, heart, muscles, bone) as "multiple metastasis".
This suggests that the compressor model was successful in identifying further metastasis. However, the compressor model did not provide all details necessary for answer completeness and was overly aggressive in compressing sites mentioned less frequently in the context.

E Extended Results

In this section, we present extended results and ablations of key design choices in our compression-prediction setup.

E.1 Extended Results on Scaling Laws of Compressor Models

We extend our analysis by constructing synthetic QA tasks on three further datasets (two synthetic, one non-synthetic) and evaluate the accuracy and perplexity of the compressions across different compressor model sizes. We measure perplexity by evaluating the log probabilities of a Llama-3.1-8B model on the target answer.

E.1.1 Summarizing Scientific Papers on QASPER

On QASPER, we extend our compressor scaling analysis to compression-prediction workflows on non-synthetic scientific papers. We vary compressor size for non-reasoning Llama and Qwen-2.5 and reasoning Qwen-3 compressors.

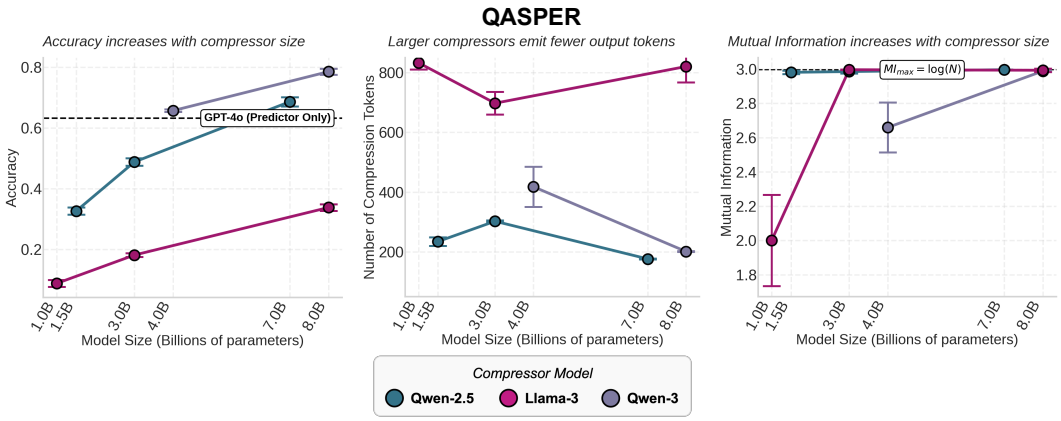

Figure 9: **Scaling behavior holds for reasoning and mixture-of-experts compressor models.** We scale compressor model size and reports a different metric of the compression step on the y-axis of each column: **(Left)** accuracy, **(Middle)** compression length, **(Right)** mutual information. Mutual information is estimated using the log probabilities of a proxy model for QWEN-2.5 and LLAMA compressors, and using internal log probabilities for QWEN-3 compressors. Larger compressors produce shorter outputs with higher downstream accuracy and higher mutual information.

Larger compressors are more accurate. We find that compressions generated by larger compressor models are more accurate across all three model families. Compressors at the 8B scale outperform the GPT-4O-only baseline (Figure 9).

Larger compressors retain more mutual information. Scaling trends observed on LONGHEALTH and FINANCEBENCH continue to hold on the QASPER dataset. Larger compressors output summaries of the scientific papers that carry more mutual information, with models at the 8B scale yielding up to $1.5\times$ more mutual information.

E.1.2 CONSTRUCTING CHAT MEMORY ON WILDCHAT

In our experiments, a compressor model summarizes long contexts with regard to context-specific questions. In practice, long context lengths also pose a major challenge in recalling information from past LM chat conversations (Eyuboglu et al., 2025). Modern LLM chatbots construct internal memory about a user's past chat histories, which serve as context for future conversations. Instead of generating query-specific summaries, we generate chat memories for each user by summarizing each chat interaction of a user using a compressor LM. The predictor then attempts to answer synthetic queries posed by the user based on the chat memory. Again, we vary the compressor model size and examine its effects on downstream perplexity, compression size, and compute cost in FLOPs-per-generation.

Larger compressors yield lower perplexity. As expected, we find in Figure 10 that chat memories generated by larger compressor models yield lower perplexity across model families. Query-agnostic summaries of chat conversations output by the largest compressor model of each model family yield up to $1.14\times$ lower log probabilities as compared to the 1B model sizes.

FLOPs-per-generation scaling holds on WILDCHAT. The scaling of compression output length on holds for QWEN-2.5 and GEMMA-3 compressors, resulting in FLOPs-per-generation scaling sublinearly with model size (Figure 10). In contrast, larger LLAMA compressors generate longer compressions, resulting in steeper scaling of FLOPs-per-generation. However, we observe consistent trends in scaling of FLOPs-per-generation between compressor model families: It is significantly more compute-efficient to scale QWEN-2.5 and GEMMA-3 compressors than LLAMA compressors.

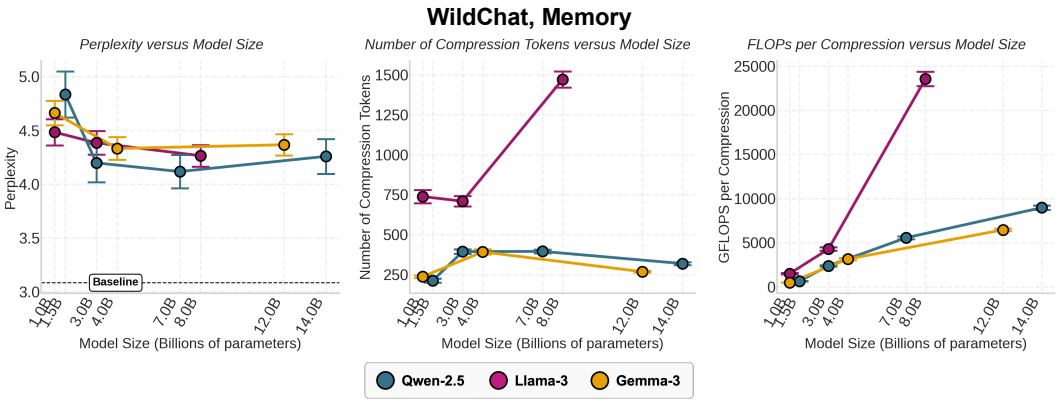

Figure 10: **Perplexity, compression length, and compute cost scale with compressor size (WILDCHAT).** We scale compressor model size and report a different metric on the y-axis of each column: **(Left)** Perplexity, with the black dashed line showing baseline perplexity given all 10 full chat conversations, **(Middle)** compression length of each set of chat conversations, **(Right)** GFLOPs-per-compression. Larger compressors produce shorter outputs with lower perplexity.

E.1.3 VARYING TASK TYPE ON FINEWEB

We extend our analysis of scaling compressor model size to a fourth dataset. On FINEWEB, we further ablate by task type: **extractive** tasks, which require the predictor model to identify and reproduce information explicit in the context—e.g. factual QA—and **creative** tasks which require the predictor model to generate longer, open-ended outputs that is not verbatim in the context—e.g., paraphrasing, format-change). We examine the compressor scaling behavior of both query-specific (Figure 11) and query-agnostic (Figure 12) summaries.

Larger compressors yield lower perplexity. As expected, we find that increasing compressor size consistently reduces perplexity for both extractive and creative tasks, on both query-specific and query-agnostic summaries. Larger compressors approach the baseline performance of giving the predictor direct access to the full uncompressed context, rather than a lossy compression thereof (Figures 11, 12).

We observe that perplexity differs in magnitude for different task types. Extractive tasks show lower perplexity values, as answers are explicitly present in the context, while creative tasks are more challenging. Query-agnostic summaries tend to achieve lower perplexity on creative tasks than query-specific summaries, which suggests that broader, more general compressions capture stylistic and semantic cues that are key to creative, open-ended generation tasks.

FLOPs-per-generation scaling holds on FINEWEB. We continue to observe predictable scaling of compute cost. As we increase compressor size, the amount of FLOPs-per-generation increases at different rates consistent with our findings on LONGHEALTH, FINANCEBENCH, and WILDCHAT. Scaling QWEN-2.5 and GEMMA-3 compressor model size comes at a cheaper compute cost than for LLAMA compressors. We find identical trends across different natures of the task (extractive vs. creative) and types of summary (query-specific vs. query-agnostic).

E.1.4 EFFECT OF PROXY MODEL ON MUTUAL INFORMATION ESTIMATION

As discussed in Section 2.2, proxy models are used to evaluate the log probabilities of compressions generated by small LMs when estimating mutual information. This is necessary when the compressor is insufficiently calibrated.

To understand the effect of the choice of proxy model in our mutual information estimator, we compare three distinct proxy LMs at the 7–8B scale: QWEN-2.5-7B, QWEN-3-8B, and LLAMA-3.1-8B (Figure 13). We find that the choice of proxy model introduces a fixed vertical offset in the

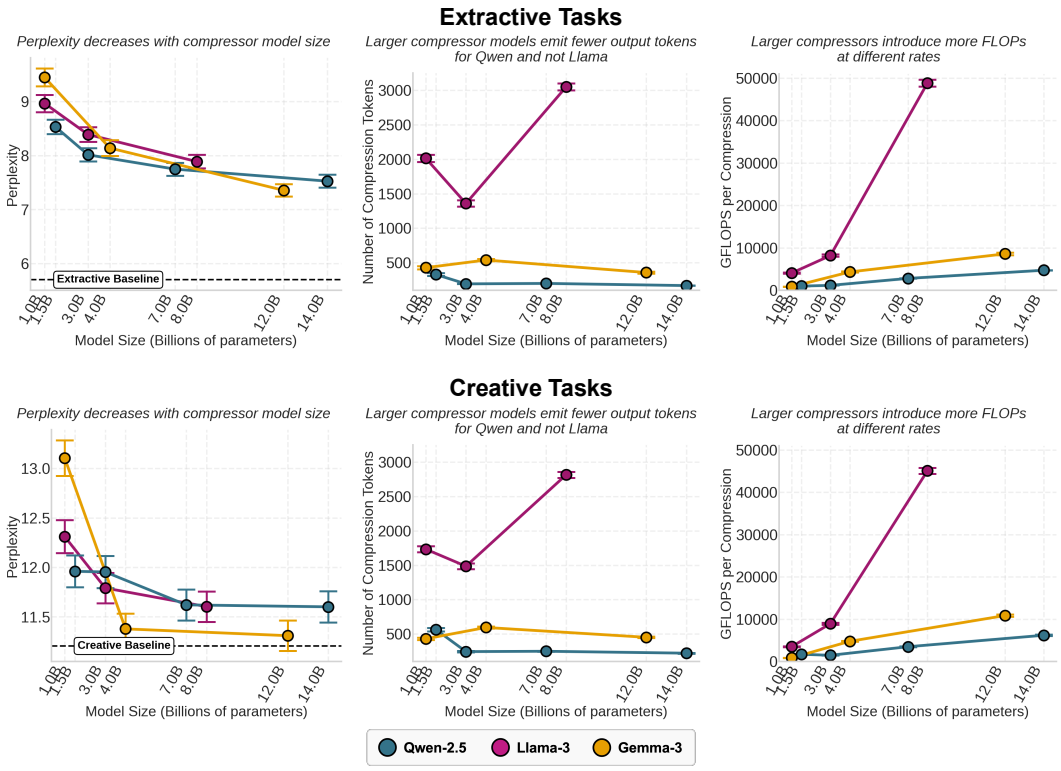

Figure 11: **Perplexity, compression length, and compute cost scale with compressor size (FINEWEB; Top: Extractive Tasks; Bottom: Creative Tasks).** We scale compressor model size and report a different metric on the y-axis of each column: **(Left)** perplexity, with the black dashed line showing baseline perplexity given the full context, **(Middle)** compression length, **(Right)** GFLOPs-per-compression. Larger compressors produce shorter outputs with lower perplexity.

MI curves that is consistent across estimates. However, it does not affect the scaling rates or any of the previous conclusions drawn.

E.1.5 EXTENDED RESULTS ON REASONING AND MIXTURE-OF-EXPERTS COMPRESSORS

We present initial results for reasoning and mixture-of-experts compressor models. Specifically, we compare how dense non-reasoning (QWEN-2.5), dense reasoning (QWEN-3), and MoE reasoning (QWEN-3) compressors scale across accuracy, compression length, and mutual information. We observe similar compressor scaling trends across reasoning and non-reasoning compressors. Interestingly, the mixture-of-experts model outperforms dense models the same scale in accuracy, producing more concise compressions with higher mutual information (Figure 14).

E.1.6 SCALING OF MUTUAL INFORMATION AND BIT EFFICIENCY ON FINANCEBENCH

To understand whether our information-theoretic findings generalize beyond LONGHEALTH, we examine mutual information and bit efficiency scaling on FINANCEBENCH. Figure 15 shows that the scaling behavior remains consistent: larger compressors retain more information about the original document while compressing more efficiently.

E.1.7 WHAT ARE THE EFFECTS OF CONCISENESS INSTRUCTIONS?

Naturally, we ask whether explicitly instructing the compressor to different levels of conciseness changes the scaling behaviors that we observe. We vary the prompt to instruct the compressor LM

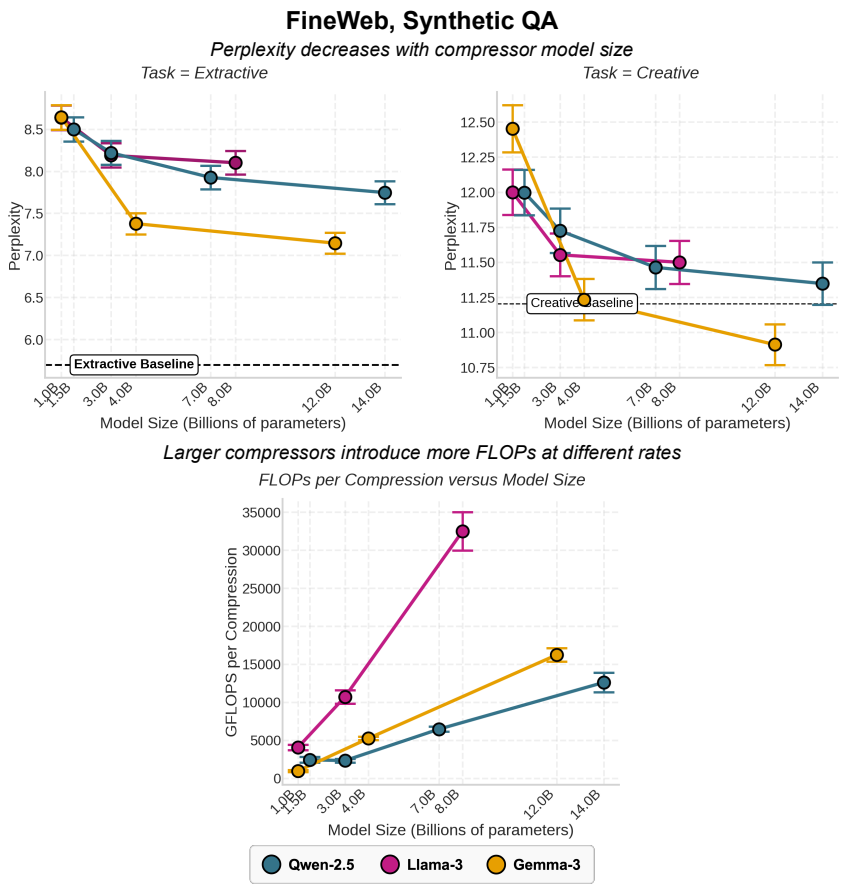

Figure 12: **Perplexity, compression length, and compute cost scale with compressor size for query-agnostic compressions (FINEWEB).** We scale compressor model size on different types of tasks (extractive and creative) and report a different metric on the y-axis of each subplot: **(Top Left)** perplexity evaluated on extractive tasks, **(Top Right)** perplexity evaluated on creative tasks, **(Bottom)** compression length. Larger compressors produce query-agnostic compressions with lower perplexity across both types of tasks.

to be *concise* (3 sentences), *normal* (6 sentences), and *elaborate* (9 sentences). We find in Figure 5 and 16 that accuracy and MI are unaffected by instructed conciseness on both LONGHEALTH and FINANCEBENCH. While prompting shifts compression output size and compute cost by an absolute offset, the compressor scaling trends hold across different conciseness constraints, showing that our scaling results are driven by compressor capacity.

E.1.8 MULTI-TURN INTERACTIONS

We extend our analysis beyond the single-turn compression-prediction setting to evaluate multi-turn workflows on LONGHEALTH.

In our setup, the predictor (LLAMA-3.1-405B) is allowed to query the compressor (LLAMA-3.2-3B) for additional information for three rounds. At each turn, the predictor integrates the compression it has constructed so far with the information it has received this turn, and issues a targeted follow-up query asking for the most relevant information, effectively separating data and control plane.

We show in Figure 18 that the amount of mutual information the compressions carry at each turn increases when given the opportunity to query more information. However, additional turns past two rounds do not provide further improvements.

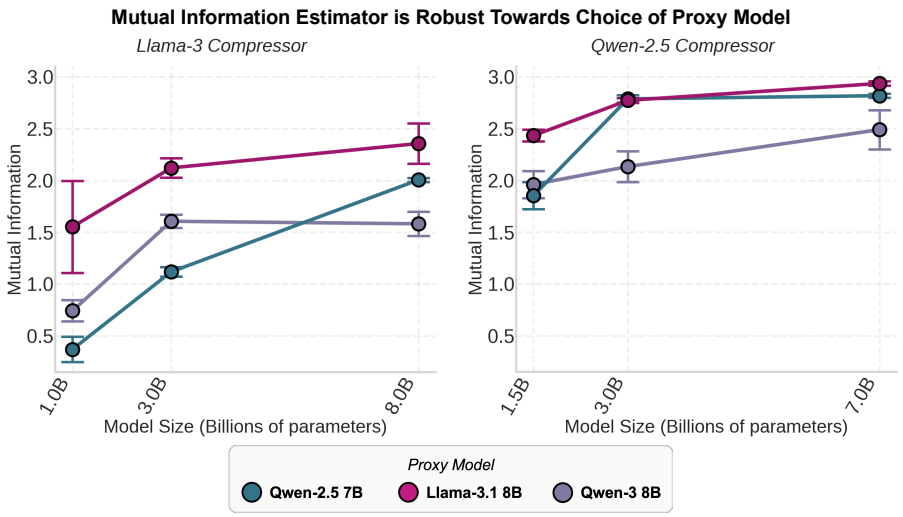

Figure 13: **Monte Carlo mutual information estimator is robust towards choice of proxy model.** The y-axis shows the mutual information estimate. Larger models display lower perplexity and similar behavior in compute scaling as on LONGHEALTH, FINANCEBENCH, and FINEWEB.

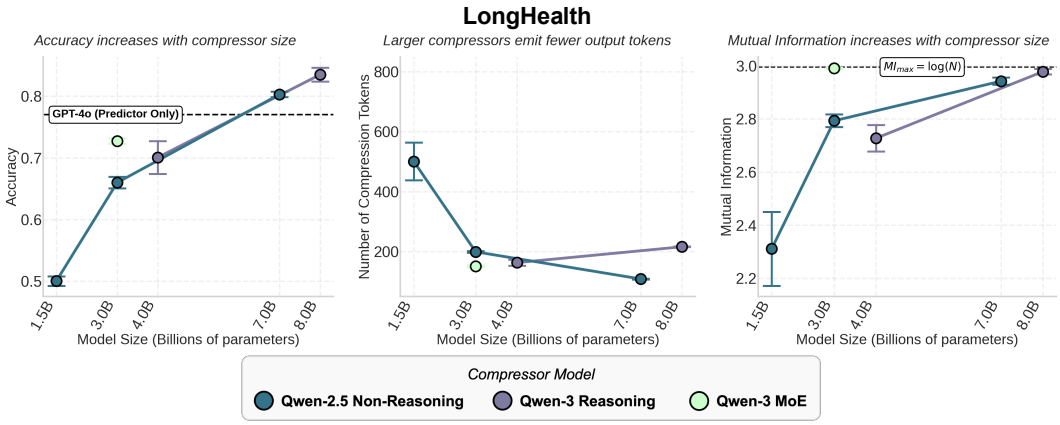

Figure 14: **Compressor scaling behavior holds for reasoning and mixture-of-experts models.** We scale compressor model size and reports a different metric of the compression step on the y-axis of each column: **(Left)** accuracy, **(Middle)** compression length, **(Right)** mutual information. Mutual information is estimated using the log probabilities of a proxy model for non-reasoning QWEN-2.5 compressors and using internal log probabilities for dense and MoE reasoning QWEN-3 compressors. Scaling trends are consistent with our observations for non-reasoning dense models (blue), where larger compressors yield higher accuracy with shorter compressions and higher mutual information. Interestingly, at the same scale (3B), the mixture-of-experts model (green) outperforms the dense models in downstream accuracy, compression conciseness, and mutual information.

E.2 EXTENDED RESULTS OF RATE-DISTORTION ANALYSIS

We aim to establish rules of thumb for design decisions around choice of compressor and predictor models based on rate-distortion theoretic concepts introduced in Section 2.2 and Appendix B.5. We further investigate the fidelity, compute cost, and communication efficiency of different compressor-predictor pairings. We examine light-weight QWEN-2.5 and LLAMA compressor models, and QWEN-2.5 and LLAMA predictor models stretching from 1B to 405B parameters.

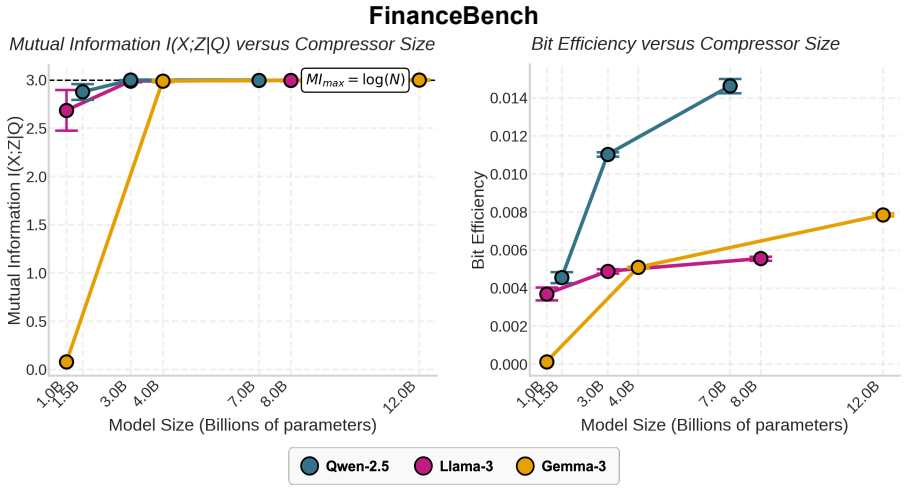

Figure 15: **Larger compressors generate outputs that carry more information about their inputs (conditioned on the query) on FINANCEBENCH.** We scale compressor model size and estimate the **(Left)** mutual information, and **(Right)** bit efficiency (bits of mutual information per token; higher is better) carried by their outputs. Larger compressor model sizes compress documents with higher mutual information and bit efficiency.

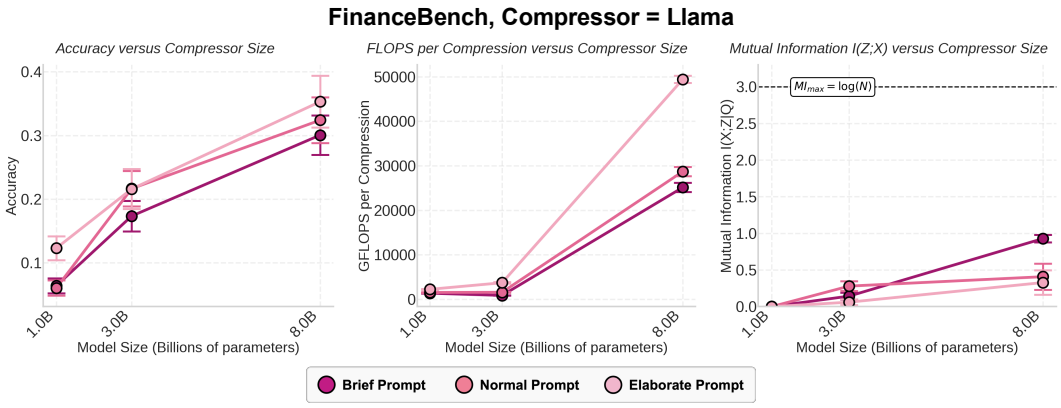

Figure 16: **Scaling behavior of compressor model size hold across instructed conciseness (COMPRESSOR = LLAMA).** We ablate over different levels of compression conciseness by varying the compression prompt instructions. We measure **(Left)** accuracy, **(Middle)** GFLOPs-per-generation, and estimate **(Right)** mutual information. We find that accuracy and mutual information are largely unaffected by conciseness instructions. Compressors instructed to be more concise are more token-efficient, and thus compute-efficient. Trends in accuracy, compute cost, and mutual information as we scale compressor hold across conciseness constraints.

E.2.1 SCALING PREDICTOR MODEL SIZE

In a compression-prediction system, the compressor model acts as a bottleneck on information about the document X. If we fix that information bottleneck and the amount of information passed through, how does the capacity to generate predictions affect downstream QA performance? We fix the compressor model to be either QWEN-2.5 or LLAMA, and vary the predictor to be LLAMA models of sizes 1B, 8B, 70B, 405B on the datasets LONGHEALTH and FINANCEBENCH (Figure 20). By plotting accuracy against bit efficiency in an alternative representation of rate-distortion analysis, downstream QA accuracy improves when moving from small predictors at 1B to larger 8–70B predictors, but then saturates with further scaling to a massive 405B predictor (Figure 21).

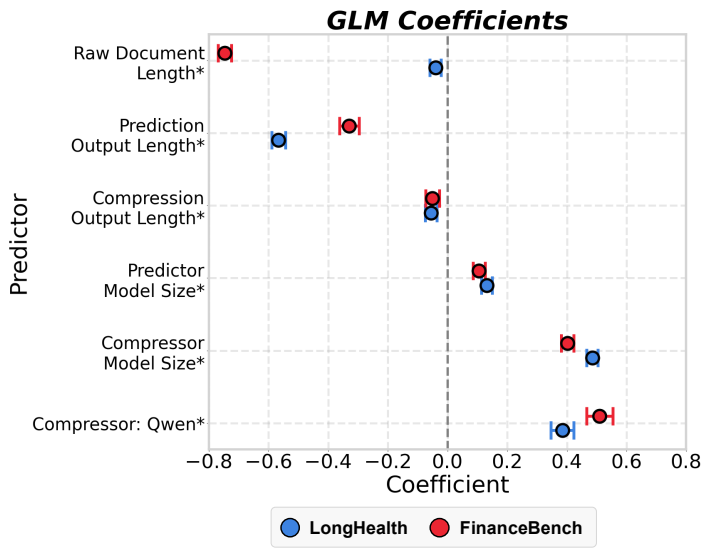

Figure 17: **Generalized Linear Model (GLM) Coefficients.** We conduct regression analysis on a GLM predicting QA correctness (0/1) on **(Blue)** LONGHEALTH, **(Red)** FINANCEBENCH. The y-axis shows coefficient estimates for each variable, horizontal bars are 95% confidence intervals, asterisks mark variables that are significant at $p < 0.05$ on both datasets. For more details on our GLM setup, refer to Appendix D.4.

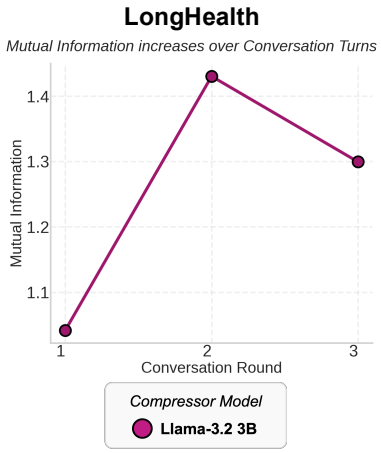

Figure 18: **Mutual information in multi-turn workflows on LONGHEALTH.** We allow a LLAMA-3.1-405B predictor to repeatedly query a LLAMA-3.2-3B compressor for additional information. Increasing the compression-prediction workflow to two turns yields an increase in mutual information, but we observe no additional improvement when extending to a third turn.

E.2.2 ALTERNATIVE MEASUREMENTS OF DISTORTION

In addition to our primary measurement of distortion as 1–accuracy, we evaluate an alternative notion of distortion based on semantic similarity. We embed both prediction and target answer using the OpenAI TEXT-EMBEDDING-3-SMALL embedding model, and measure distortion as 1–cosine similarity between the semantic embeddings.

We observe characteristic rate-distortion curves with the same ordering of predictor model sizes (Figure 23).

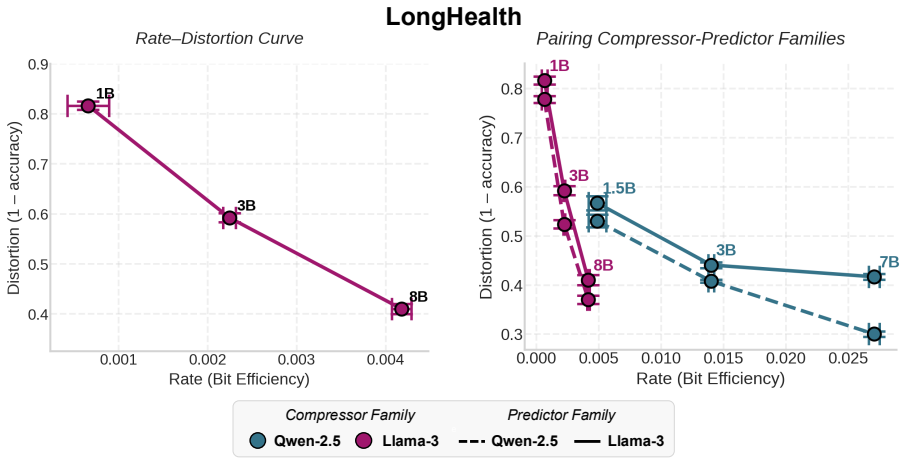

Figure 19: **Exploring the trade-off between compression and fidelity loss: rate-distortion curve.** We vary predictor and compressor model in the compression-prediction workflow and measure the distortion on the y-axis and estimate the rate on the x-axis. **(Left)** We examine a single compressor-predictor LM pairing, COMPRESSOR=LLAMA-3 and PREDICTOR=LLAMA-3.3-70B. **(Right)** We compare different compressor-predictor LM pairings, where the predictor model is QWEN-2.5-72B ("Qwen-2.5") or LLAMA-3.3-70B ("Llama"). Markers indicate compressor sizes (1B, 3B, 8B) in the LLAMA-3 compressor model family.

E.3 EXTENDED RESULTS OF DEEP RESEARCH ANALYSIS

In our Deep Research scaling experiments, we additionally ablate the effect of search result quantity by providing GPT-4O with the top $k = 1, 3, 6$ search results from each of the 8 predictor queries directly, bypassing the compression step. This analysis in Figure 7 reveals that even with maximal context utilization ($k = 6$, totaling 48 sources), the uncompressed approach achieves a similar level of *RACE* scores at significantly higher API costs compared to our compression-based strategy.

We also compute the compute cost in FLOPs-per-generation of the system, as shown in Figure 24. We observe an increase in the number of total FLOPs-per-generation used by the Deep Research system as predictor size increases. We generally observe that larger compressors extract and provide more tokens of information in their compressions.

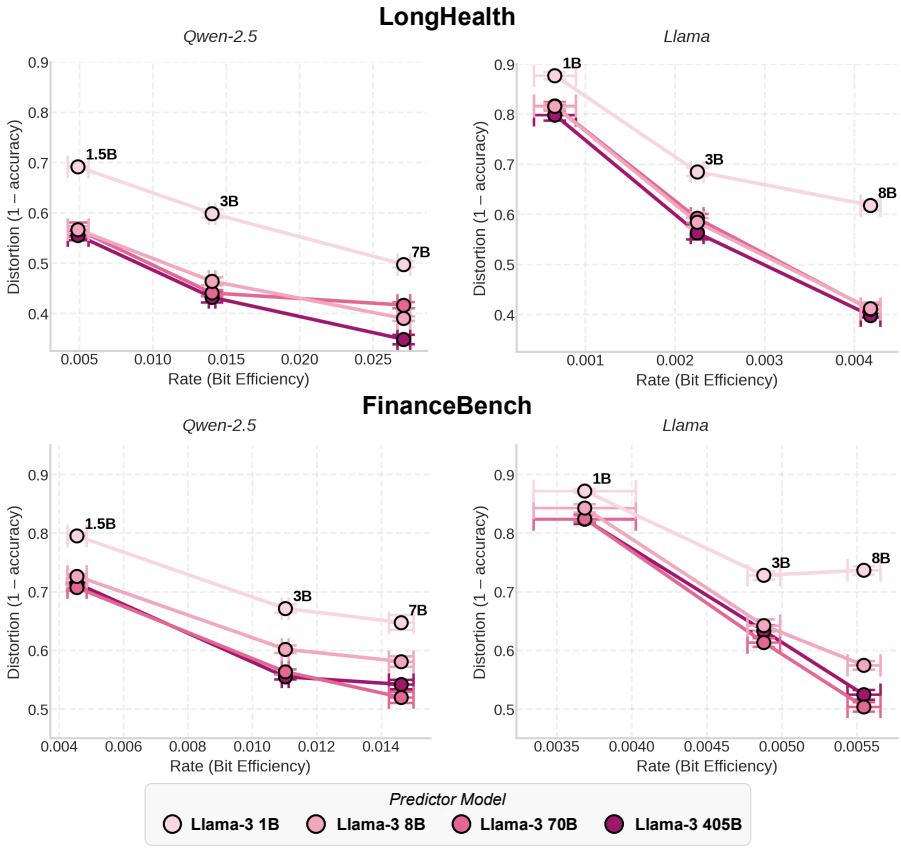

Figure 20: **Exploring the trade-off between compression and fidelity loss: rate-distortion curve.** We vary both predictor and compressor model in the compression-prediction workflow and measure the distortion on the y-axis and estimate the rate on the x-axis. We plot the resulting rate-distortion curves across predictor sizes 1B, 8B, 70B, and 405B for **(Left)** QWEN-2.5 and **(Right)** LLAMA compressors. We evaluate the rate-distortion curves for two datasets: **(Top)** LONGHEALTH, **(Bottom)** FINANCEBENCH.

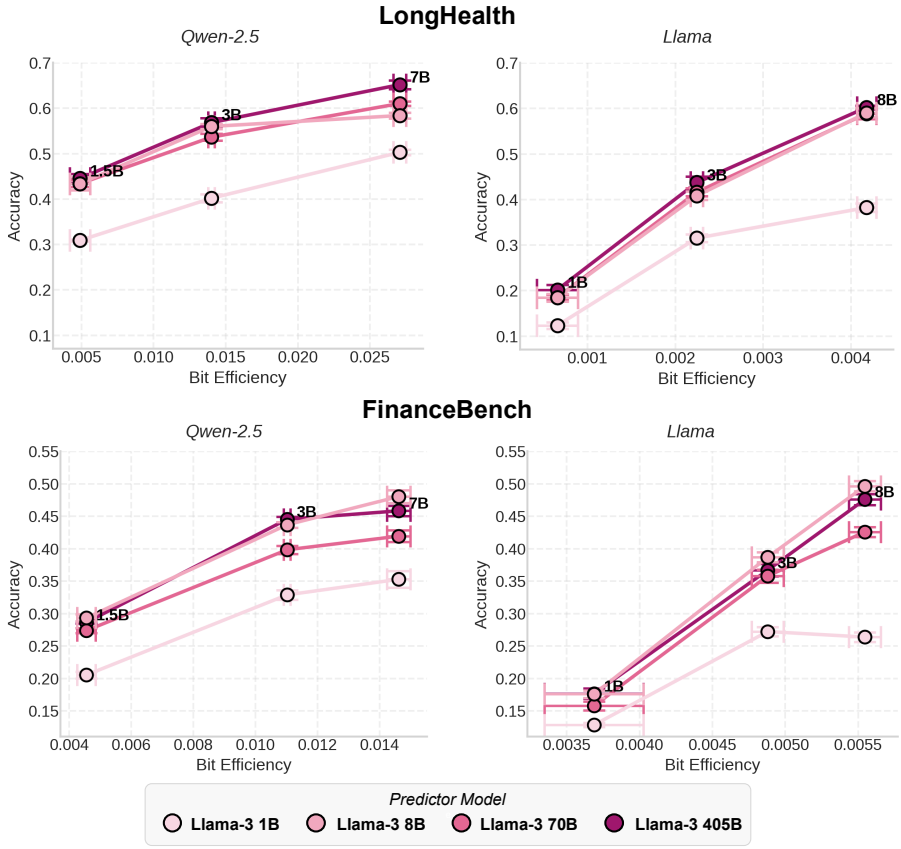

Figure 21: **Exploring the relationship between compression and accuracy.** The y-axis depicts accuracy and the x-axis shows bit efficiency. *Bit efficiency* is defined as the bits of mutual information encoded in each compression token. Markers indicate compressor sizes in the QWEN-2.5 (1.5B, 3B, 7B) and LLAMA (1B, 3B, 8B) compressor model family; vertical and horizontal bars denote standard errors.

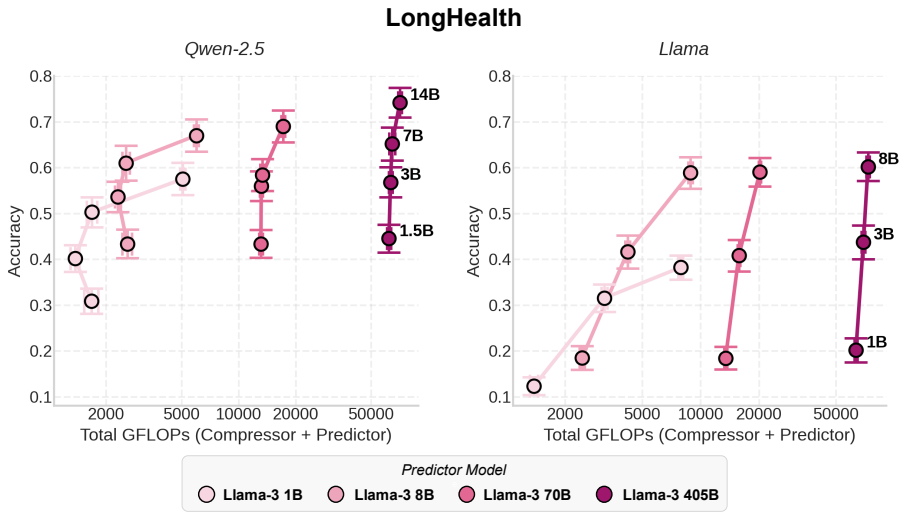

Figure 22: **QA Accuracy versus total compute cost on FINANCEBENCH.** In each panel, the y-axis shows the accuracy and the x-axis plots total compute cost in FLOPs-per-generation on a log-scale for **(Left)** QWEN-2.5, **(Right)** LLAMA-3 compressor LMs. Markers indicate compressor sizes in the QWEN-2.5 (1.5B, 3B, 7B) and LLAMA-3 (1B, 3B, 8B) compressor model family; vertical and horizontal bars denote standard errors.

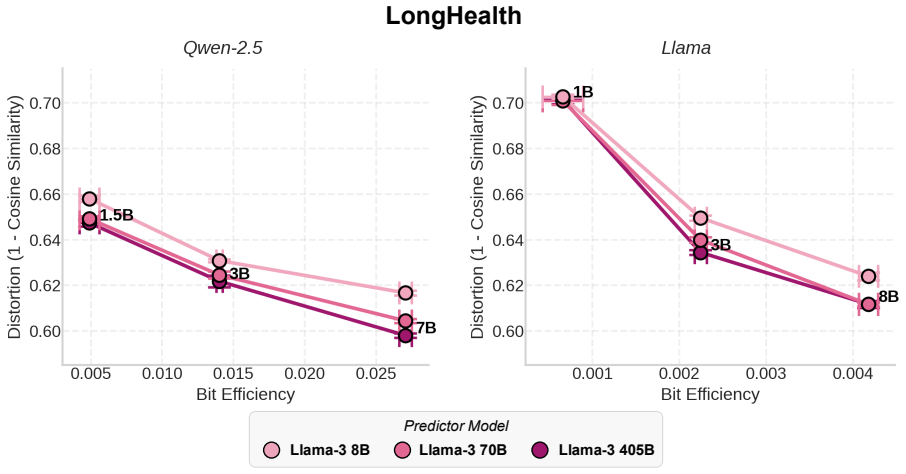

Figure 23: **Exploring the trade-off between compression and fidelity loss: alternative definition of distortion (LONGHEALTH).** We vary both predictor and compressor model in the compression-prediction workflow and measure the distortion on the y-axis as 1–cosine difference between the semantic embedding of the prediction and target answer. We estimate the rate on the x-axis. We plot the resulting rate-distortion curves across predictor sizes 8B, 70B, and 405B for **(Left)** QWEN-2.5 and **(Right)** LLAMA compressors.

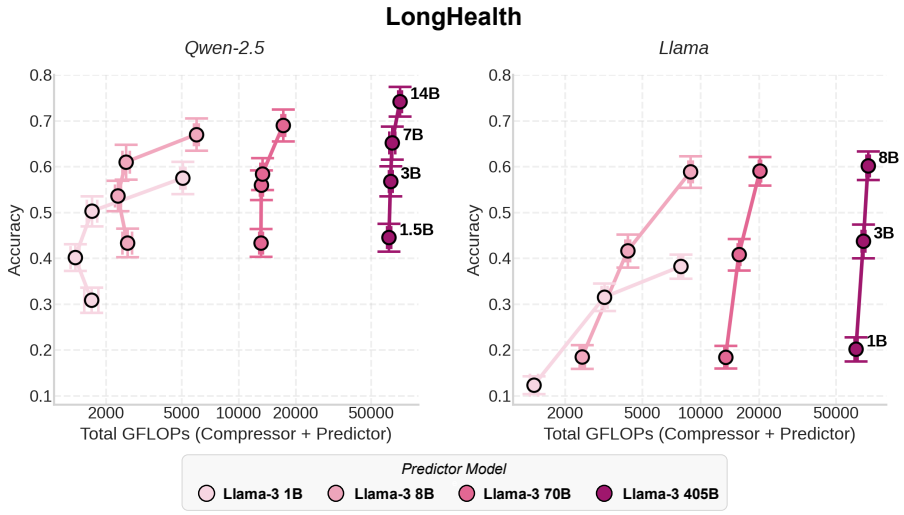

Figure 24: **Compressor model size versus compute and token usage.** In each panel, the x-axis shows the Qwen compressor size (in billions of parameters). **(Left)** Total GFLOPs per task grows with compressor model size, with larger predictors (Llama 405B, 70B, 8B) amplifying compute cost. **(Right)** Compressor output tokens per task, which remain relatively stable across predictors (GPT-4o, Llama 405B, 70B, 8B), increase moderately with larger compressors.