AN INFORMATION THEORETIC PERSPECTIVE ON AGENTIC SYSTEM DESIGN

Anonymous authors

000

001

002003004

006

008

010 011

012

013

014

015

016

017

018

019

021

023

024

025

026

027

028

029

031

033

037

040 041 042

043

044

045

047

048

051

052

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." 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 compressorpredictor systems remains largely ad-hoc. Little guidance exists on how compressor and predictor choices shape downstream performance. Attributing gains to compression versus prediction typically requires exhaustive pairwise sweeps. 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. Using a rate-distortion analysis, we show that mutual information strongly predicts downstream performance. With this toolkit, we perform a comprehensive empirical analysis across four datasets and three model families. Results reveal that larger compressors are both more accurate and more token-efficient, conveying more bits of mutual 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.

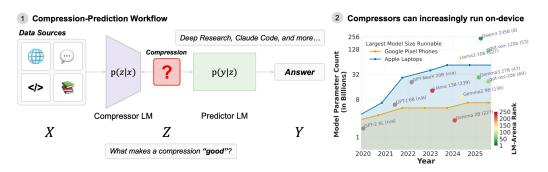


Figure 1: Why compressors matter. Many agentic LM systems rely on compressors, and evaluating their quality in a task-independent way is essential as personal devices grow powerful enough to host them. (Left) A compressor condenses a long input X into a shorter summary Z that a predictor expands into the final answer Y. (Right) Consumer hardware can now run increasingly large openweight LMs, shown for Google Pixel phones and Apple MacBook laptops (2020-2025) under FP16 precision with memory estimates from Modal (Lu, 2024). LM-Arena ranks (in parenthesis, lower is better) 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. While multi-LM architectures vary widely, this 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).

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 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.

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. 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 (Kaplan et al., 2020; Hoffmann et al., 2022). 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 four datasets (LongHealth (Adams et al., 2024), FinanceBench (Islam et al., 2023), 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 communication-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

Agentic architectures have become a practical way to improve cost-efficiency and accuracy on complex tasks (Du et al., 2025; Narayan et al., 2025). 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 (Saad-Falcon et al., 2024; Erdogan et al., 2025). Overall, they find that performance improves with larger models, more agents, and additional rounds.

2.2 Information Theoretic Problem Setup

Information theory has been applied to deep learning in several ways, primarily within single models. Researchers have used it to analyze intermediate representations in neural networks (Kawaguchi et al., 2023; Tishby & Zaslavsky, 2015), evaluate natural language generations including summaries and token sequences (Arda & Yener, 2025; Darrin et al., 2024; Shani et al., 2025), and define training objectives (Kirsch et al., 2020; Goldfeld & Polyanskiy, 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 LM compressor, and one LM predictor ("Deep Research" will later extend to multiple compressors). Let X be the input context, Q be the query, 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. To estimate MI, we start with the KL divergence (Kullback & Leibler, 1951) representation:

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

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),

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

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

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 $R=\frac{I(X;Z|Q)}{L}$, for L output tokens (measured in bits of mutual information per token). For simplicity, we define distortion as D=1-ACC(Z), by using the accuracy $0 \le ACC(Z) \le 1$. See the R-D curve in Figure 15 (left). As rate increases, we expect distortion to converge towards a lower bound (irreducible error). See Appendix B.2 for further details.

3 RESULTS

We evaluate the compressor-predictor system as an information bottleneck across different tasks and domains. We begin with a comprehensive scaling analysis of compressor and predictor model family and sizes, finding that larger compressors are not only more accurate but also more concise, leading to sublinear FLOPs 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 four datasets: (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) WILDCHAT is a large-scale LM conversation dataset (Zhao et al., 2024). (d) 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 and FinanceBench, assessing correctness of the predictions against the ground-truth using a GPT-40-mini judge. We use perplexity for Wildchat and FineWeb.

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). For simplicity, we choose GPT-style models, and postpone analysis of reasoning models for future work (which will require a breakdown of reasoning and answer tokens). See Appendix D.2 for further details.

Predictor model We evaluate larger frontier models GPT-40 (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 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). On LONGHEALTH, 7B and 8B models are up to $3.1\times$ more accurate than 1B models and surpass the GPT-40-only baseline by 4pp. On FINANCEBENCH, 7B and 8B models are up to $2.6\times$ more accurate than 1B models and are able to

recover 97% of the GPT-40-only baseline accuracy. The same scaling behavior holds for GEMMA-3 models.

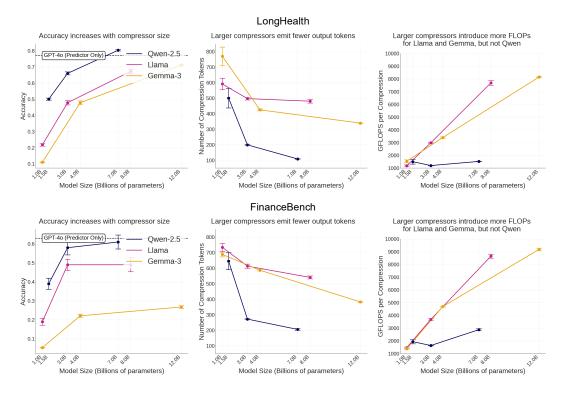


Figure 2: **Downstream accuracy, length of compression, and inference cost scale with compressor size.** The *y*-axis shows the (**Left**) accuracy, with the black dashed line showing the GPT-40 model baseline, (**Middle**) number of tokens in compression, (**Right**) GFLOPs per compression on: (**Top**) LONGHEALTH, (**Bottom**) FINANCEBENCH. Larger models are more token-efficient, trading off accuracy for compute. We see consistent scaling trends on WILDCHAT (Appendix E.1.1) and FINEWEB (Appendix E.1.2)

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 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.6.

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

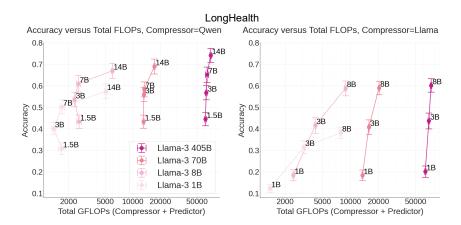


Figure 3: **QA Accuracy versus total compute cost on LONGHEALTH.** y-axis shows accuracy and x-axis plots total compute cost in FLOPs-per-generation on a log-scale for (**Left**) QWEN-2.5, (**Right**) LLAMA-3 compressor LMs. Markers indicate compressor sizes in the QWEN-2.5 (1.5B, 3B, 7B, 14B) and LLAMA-3 (1B, 3B, 8B) compressor model family; vertical and horizontal bars denote standard errors. Appendix E.1.3 shows consistent trends on FINANCEBENCH.

 $(\approx 8B\text{-}70B)$ is reached. The gains in accuracy by increasing predictor size from 70B to 405B are within 12% (LONGHEALTH) and 1% (FINANCEBENCH). In contrast, scaling compressor LM for both families leads to steeper increases in performance for fewer FLOPs-per-generation spent. For the QWEN-2.5, 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).

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 our information-theoretic framework: 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.

We find that smaller LMs at 1-3B were often miscalibrated and could assign high likelihoods to nonsensical token sequences. Thus, we evaluated the log probabilities using a proxy model at the 7B scale. To mitigate biases, the proxy model was selected from a different family.

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 size. By contrast, on FINANCEBENCH, mutual information saturates already at the 3B scale. We observe that this primarily occurs on datasets with a highly heterogeneous corpus of context documents.

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.

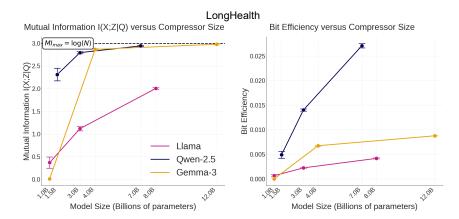


Figure 4: Larger compressors have outputs that carry more information about their inputs (conditioned on the query) on LONGHEALTH. Larger compressor model sizes compress documents with higher (Left) mutual information, (Right) bit efficiency (bits of mutual information per token; higher is better). The black dotted line represents the theoretical maximum of mutual information 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.1.3)

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, compute cost, MI, and bit efficiency remained consistent across all conciseness instructions on both LONGHEALTH and FINANCEBENCH (Figures 5, 13). The relative improvements from larger compressors persist regardless of prompted output length, confirming that model capacity drives these efficiency gains.

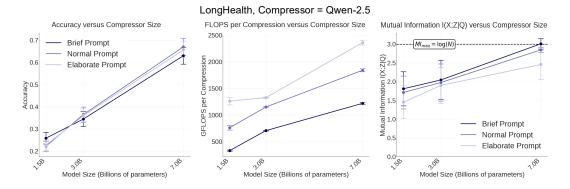


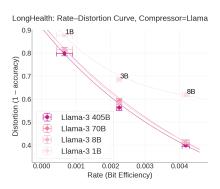
Figure 5: Scaling behaviors of compressor model size hold across instructed conciseness (COMPRESSOR = QWEN-2.5). (Left) Accuracy (Middle) GFLOPs per compression (Right) MI versus compressor model size. Accuracy and MI are largely unaffected by conciseness instructions. Compressors instructed to be more concise are more token-efficient, and thus compute-efficient. Trends in accuracy, compute cost, and MI as we scale compressor hold across conciseness constraints. Appendix E.1.4 shows analogous results on FINANCEBENCH.

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 (Appendix B.2), we fit decaying exponential functions to the rate-distortion data. This fit characterizes the correlation between

information rate and distortion and corroborates our previous finding that scaling predictors beyond 70B yield only marginal improvements in distortion (Figure 6).

Furthermore, we find in Figure 6 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.1.4).



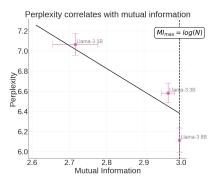


Figure 6: Mutual information and bit efficiency correlate strongly with downstream performance. (Left) Rate-distortion analysis across compressor sizes 1B, 8B, 70B, and 405B. Dotted lines show fitted exponential-decay functions. (**Right**) Perplexity versus mutual information. Black line shows fitted linear function (r = -0.84, $R^2 = 0.71$). Black dotted line represents theoretical maximum of MI. Vertical and horizontal bars denote standard errors.

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 15).

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

We find that compressor model family is the most important factor (Figure 14) 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 (Figure 8). 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 vary predictor sizes across the LLAMA family and compressor sizes across the QWEN-2.5 family. Full experimental details are in Appendix D.5.

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-40 predictor. A 14B QWEN-2.5 compressor paired with a GPT-40 predictor achieves 2% higher RACE scores at only 30% of the API cost compared to the uncompressed baseline. We detail further findings in our scaling experiments in Appendix E.3.

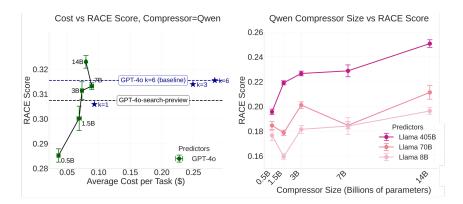


Figure 7: **Deep Research Scaling Results.** (**Left**) RACE score versus average task cost when using GPT-40 as a predictor with QWEN-2.5 compressors of varying sizes. Costs are based on GPT-40 API rates (Aug 2025: \$2.50/1M input tokens, \$10.00/1M output tokens). Larger compressors improve performance with minimal cost increases. For reference, we include GPT-40 results without compression and also for the GPT-40-SEARCH-PREVIEW model. (**Right**) RACE scores for different QWEN-2.5 compressor sizes (0.5B–14B) under three LLAMA predictors (8B, 70B, 405B).

4 DISCUSSION

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

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 for optimizing compressor-predictor communication. 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

- Scaling compressors is more effective than scaling predictors and can be done 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 compressor size yields
 greater accuracy gains than scaling predictor size. 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.
- 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.5, 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.2, while the generalized linear model specifications for meta-analysis are provided in Appendix D.2.

REFERENCES

- 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.
- 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.
- 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.
- 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/.
- 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.
- 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.
- 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.
- 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.
- 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.

- 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.
 - 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.
 - 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.
 - 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.
 - 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.
 - 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 Ilms and humans trade compression for meaning. *arXiv preprint arXiv:2505.17117*, 2025.
 - 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.
 - 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.

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). These benchmarks focus on measuring the output quality and downstream utility of generated research reports. Our own experiments on DeepResearch Bench similarly evaluate end-to-end scaling of predictors and compressors. We view these results as a complement to our bottleneck analysis, showing how design principles extend into realistic multi-agent workflows.

B EXTENDED DESCRIPTION OF METHODS

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

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:

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

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

B.2 RATE-DISTORTION-THEORY

Assume X to be an independent Gaussian random variable, we model the rate-distortion function as

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

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 (wil, 2005)

$$D_{Gaussian}(R) = \sigma^2 2^{-2R}$$

$$= \sigma^2 e^{-2\ln 2R}$$

$$= C e^{-\beta R}, \quad \text{with} \quad C = \sigma^2, \ \beta = 2\ln(2).$$

In practice, we treat C and β 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,

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

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

C PROMPTS

C.1 COMPRESSOR MODEL PROMPTS

We used the following prompt templates to compress the raw context documents on LONGHEALTH, FINANCEBENCH, FINEWEB, and each chat conversation on WILDCHAT:

Query-Specific Compression Prompt Template (LongHealth/FinanceBench/FineWeb) 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 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

We used the following prompt templates to answer extractive QA tasks on LONGHEALTH, FINANCEBENCH, WILDCHAT, and FINEWEB, and creative tasks on FINEWEB.

compress the raw context documents on LONGHEALTH, FINANCEBENCH, FINEWEB, and each chat conversation on WILDCHAT:

```
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: <bri>defly 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 chat memory.

Question:
{query}

Chat Memory:
{summary}

Please respond in the following JSON format: <bri>
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. We indicate in parentheses if the prompt was used for the Predictor LM or Compressor LM.

```
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.
```

```
864
              Return your response in this exact JSON format:
865
866
                  "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
867
                  analysis.",
868
                  "queries":
                       {{
    "search_query": "specific search terms optimized for Google",
    "search_query": "specific specific does this query address a
869
                          "sub_task": "What specific question does this query address and how does it serve the research
870
                          plan?"
                      }}
871
                        [{
    "search_query": "second strategic search query",
    "search_query": "second strategic search query",
872
                          "sub_task": "What does this query aim to discover and how does it fit the research
873
                           architecture?"
874
                        {{
    "search_query": "third targeted search query",
    "sub_task": "What aspect does this explore and why is it essential to the research plan?"
875
876
877
                          "search.query": "fourth strategic search query",
"sub_task": "What question does this answer and how does it complement other queries?"
878
                        [{
    "search.query": "fifth focused search query",
    "search.query": "fifth focused search query",
879
880
                           "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?"
882
883
884
                          "search.query": "seventh strategic search query",
"sub.task": "What specific gap does this fill in the research architecture?"
885
                        [{
    "search_query": "eighth concluding search query",
    "search_query": "eighth concluding search query",
886
887
                          "sub_task": "What final aspect does this cover and how does it complete the comprehensive
                          research?"
888
                      }}
889
                  "synthesis strategy": "Detailed strategy for combining findings from all 8 queries based on your
890
                  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."
891
892
              **Strategic Guidelines:**
              1. Each search query should be 3-8 well-chosen keywords targeted for your specific research objectives
893
                   Design queries to serve complementary roles in your research architecture (not just generic
894
              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
895
              5. Create a synthesis strategy that reflects your planned research structure
896
897
              **Research Focus Areas to Consider:**
             Foundational understanding and current stateKey challenges, problems, or limitations
898
              - Solutions, methodologies, and best practices
899
              - Evidence, data, and empirical findings - Future trends, developments, and implications
900
              - Multiple perspectives and stakeholder viewpoints
901
              CRITICAL: You must return ONLY the JSON object. Do NOT format it as a code block with '''json''' or
902
              any other markdown formatting. Return the raw JSON object directly.
903
904
```

```
DeepResearch Source Summarization Prompt Template (Compressor)
```

906

907

908

909 910

911

912

913

914

915

916

```
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
```

945 946

947 948

949

950

951 952

953

954 955

956

957

958 959

960

961

962

963 964

965

966

967

968

969

```
Specific research findings and study results
919
             - Direct quotes and expert opinions
- Policy details and regulatory information
             - Implementation details and methodologies
921
              **Comprehensive Coverage:**
              - Key facts directly relevant to both the main query AND the specific sub-task
922
             Important concepts, definitions, and explanationsCause-and-effect relationships and underlying mechanisms
923
              - Trends, patterns, and developments over time
924
              - Challenges, limitations, and problem areas identified
925
              **Analytical Insights:**
              - Implications and significance of the information
926
             - Relationships between different data points
927
             - Comparative information and benchmarks - Future projections and forecasted trends
928
             - Expert assessments and professional evaluations
929
              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
930
931
              comprehensive research report. Pay special attention to information that directly addresses the sub-task
932
933
              Return your extraction in JSON format with these fields:
              - "explanation": Your detailed extraction of specific information, facts, data, examples, and evidence with extensive detail
934
               "answer": "relevant" if this content contains information relevant to the query and sub-task, "not
935
              relevant" otherwise
936
              CRITICAL JSON FORMATTING RULES:
937
             - Replace all double quotes (") inside text with single quotes (') - Replace all newlines with spaces
938
             - Ensure the JSON is valid and parseable
             - Do NOT use line breaks within the JSON fields
939
940
              {{\text{"explanation": "Your detailed extraction with specific facts, numbers, examples, and evidence using }}
941
              single quotes for any nested quotes", "answer": "relevant"}}
942
              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.
943
```

```
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
     {\tt **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
     **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.
```

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 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-40-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}
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**.
question you can generate from it>
   "question": "<question>",
"answer": "<answer>",
}}
Your answer (YOU MUST ONLY RESPOND WITH THE JSON OBJECT):
```

D.1.4 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-40-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.
{context}
Use only information from {\tt 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 =
- 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: <br/>
Spriefly think about the information you have and guestions
you can generate from it>
     "questions": [
         "question": "<question 1>",
"answer": "<answer 1>",
"type": "qa"
              "topic": "<topic 2>",
"question": "<question 2>",
"answer": "<answer 2>",
"type": "qa"
               "topic": "<topic 3>",
              "question": "<question
"answer": "<answer 3>",
"type": "generation"
                                "<question 3>",
               "topic": "<topic 4>",
              "question": "<question
"answer": "<answer 4>",
"type": "generation"
                                 "<question 4>",
               "topic": "<topic 5>",
              "question": "<question
"answer": "<answer 5>",
"type": "generation"
                                  "<question 5>",
         }}
}}
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.

All three 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 and QWEN-2.5, and 1.0 for GEMMA-3.

D.3 PREDICTOR MODEL DETAILS

As predictor models we use GPT-40, 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 1{Compressor=Qwen} for the compressor model family,

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

D.5 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.5.1 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.

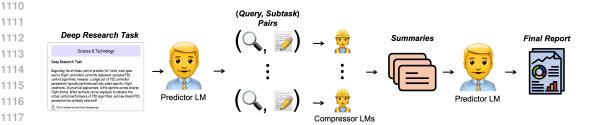


Figure 8: **Deep Research workflow.** A predictor LM decomposes a Deep Research task into (*Query, Subtask*) pairs, where each pair specifies a targeted web search and an associated analysis instruction. Compressor LMs work in parallel to retrieve evidence, process it according to the subtask, and compress 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-40 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.5.2 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,

1138 1139

1140

1141

1142 1143

1144 1145

1146

1147

1148 1149

1150 1151

1152 1153

1154

1155

1156

1157

1158

1159 1160

1161

1162

1163

1164

1165

1167

1168

1169

1170

1171

1172 1173

1174 1175

1176

1177 1178

1179

1180

1181 1182

1183

1184 1185

1187

1134 enabling free, high-throughput parallel inference. All compressor models use a temperature of 0.7 1135 and a maximum output token limit of 2,000 tokens per response. The specific compressor models 1136 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.5.3 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-40: 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.6 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 1186

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 compressionprediction setup.

E.1 EXTENDED RESULTS ON SCALING LAWS OF COMPRESSOR MODELS

We extend our analysis by constructing synthetic QA tasks on two further datasets and evaluate the 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 given the chat context/memories generated by GEMMA-3 and QWEN-2.5 compressor models, and the log probabilities of a QWEN-2.5-7B model for LLAMA compressor models.

E.1.1 CONSTRUCTING CHAT MEMORY ON WILDCHAT

In our experiments, a compressor model summarizes long contexts with regards 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 model. The predictor model 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.

1247

1251

1254

1255

1256

1257

1258

1259

1260 1261

1262

1263

1264

1265 1266

1267

1268

1269

1270

1271 1272

1273

1274

1275

1276

1277

1278

1279 1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290 1291

1293

1294

1295

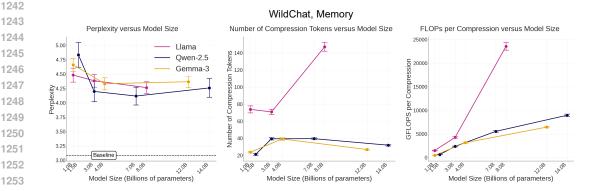


Figure 9: Perplexity and inference cost scale with compressor model size when generating memory for LMs (WILDCHAT). The y-axis shows the (Left) perplexity on, with the black dashed line showing the baseline perplexity given all 10 full uncompressed chat conversations, (Middle) Number of tokens in compression of each chat conversation, (**Right**) GFLOPs-per-generation of each chat conversation. Larger models display lower perplexity and similar behavior in compute scaling as on LONGHEALTH, FINANCEBENCH, and FINEWEB.

Larger compressor models yield lower perplexity As expected, we find in Figure 9 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.

Compute cost scaling holds on WILDCHAT The scaling of compression/memory size on WILD-CHAT (Figure 9) is less clear than for the question-specific compressions generated on LONGHEALTH and FINANCEBENCH (Figure 2). However, we observe consistent trends in scaling of compute cost between compressor model families: It is significantly cheaper to scale compressor model size for QWEN-2.5 and GEMMA-3 than for LLAMA.

E.1.2VARYING 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 compressor size scaling behavior of both query-specific (Figure 10) and query-agnostic (Figure 11) summaries.

Larger compressor models yield lower perplexity As expected, we find that increasing compressor model size consistently reduces perplexity for both extractive and creative tasks, query-specific and query-agnostic summaries. Larger compressor models approach the baseline performance of giving the predictor direct access to the full uncompressed context, rather than a lossy compression thereof (Figures 10, 11).

We observe that perplexity scores differ 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 values 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.

Compute cost 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).

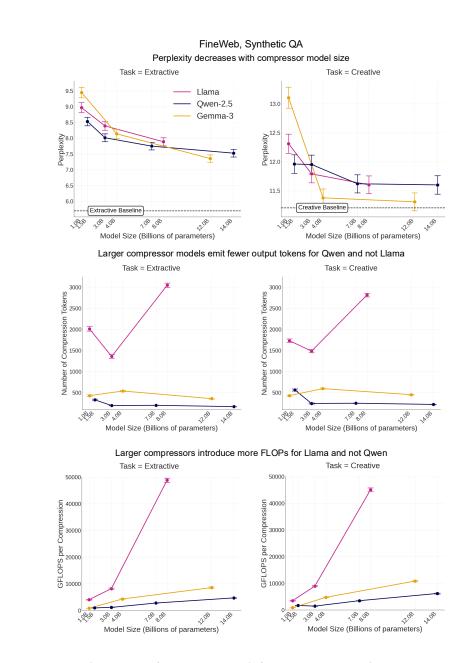


Figure 10: Perplexity, length of summary, and inference cost scale with compressor model size on FINEWEB for query-specific summaries. The y-axis shows the (Top) perplexity, with the black dashed line showing the baseline perplexity given the full uncompressed context, (Bottom) GFLOPs-per-compression on: (Left) extractive tasks, (Right) creative tasks. Larger models are more token-efficient, trading off perplexity for compute.

E.1.3 SCALING OF MUTUAL INFORMATION AND BIT EFFICIENCY ON FINANCEBENCH

To understand whether our info-theoretic findings generalize beyond LONGHEALTH, we repeat the mutual information and bit efficiency analysis (Section 3.2) on FINANCEBENCH. Figure 12 shows that the scaling behavior remains consistent: larger compressor models retain more information about the original document while compressing more efficiently.

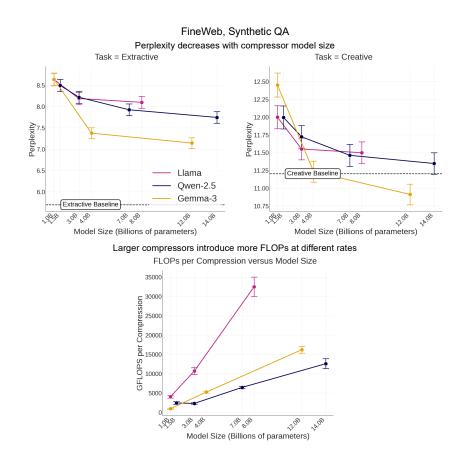


Figure 11: **Perplexity, length of summary, and inference cost scale with compressor model size on FINEWEB for "general" query-agnostic summaries.** The *y*-axis shows the **(Top)** perplexity on **(Top Left)** extractive and **(Top Right)** creative tasks, with the black dashed line showing the baseline perplexity given the full uncompressed context. **(Middle)** Number of tokens in compressed text, **(Bottom)** GFLOPs-per-compression. Larger models are more token-efficient, trading off perplexity for compute.

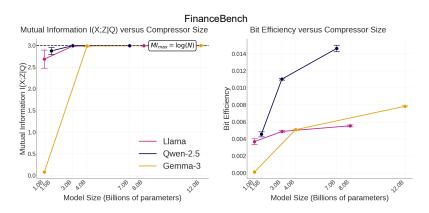


Figure 12: Larger compressor models have outputs that carry more information about their inputs (conditioned on the query) on FINANCEBENCH. Larger compressor model sizes compress documents with higher (Left) mutual information, (Right) bit efficiency (bits of mutual information per token). The black dotted line (Left) represents the theoretical maximum of MI at the natural logarithm $\log(N)$, where N is the number of documents MI is computed across.

E.1.4 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 to be *concise* (3 sentences), *normal* (6 sentences), and *elaborate* (9 sentences). We find in Figure 5 and 13 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.

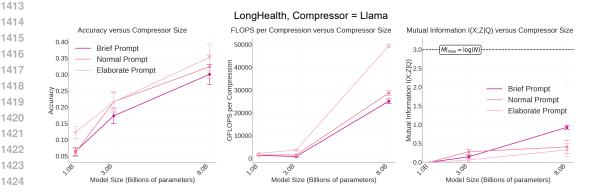


Figure 13: Scaling behaviors of compressor model size hold across instructed conciseness (COMPRESSOR = LLAMA). (Left) Accuracy (Middle) GFLOPs-per-compression (Right) MI versus compressor model size. Accuracy and MI are largely unaffected by conciseness instructions. Compressors instructed to be more concise are more token-efficient, and thus compute-efficient. Trends in accuracy, compute cost, and MI as we scale compressor hold across conciseness constraints.

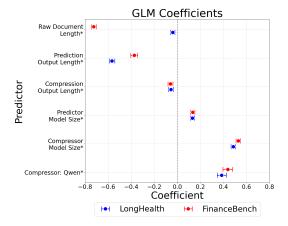
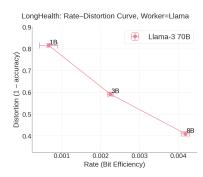


Figure 14: Generalized Linear Model (GLM) Coefficients. Regression analysis on a GLM predicting OA correctness (0/1) on (Blue) LONGHEALTH, (Red) FINANCEBENCH. The y-axis shows coefficient estimates for each variable, horizontal bars are 95% confidence intervals, asterisks mark variables that are significant at p < 0.05 on both datasets. For more details on our GLM setup, refer to Appendix D.4.

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.2. We further investigate the fidelity, compute cost, and communication efficiency of different compressorpredictor 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.



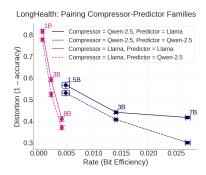


Figure 15: Exploring the trade-off between compression and fidelity loss: rate-distortion curve. y-axis depicts distortion and x-axis shows rate. Markers indicate compressor sizes (1B, 3B, 8B) in the LLAMA-3 compressor model family; vertical and horizontal bars denote standard errors. (Left) We examine 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"). See Appendix B.2 for results on further compressor-predictor LM pairings.

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 predictor's capacity to decode affect downstream QA performance? We fix the compressor model to be LLAMA, and vary the predictor to be LLAMA models of different sizes (1B, 8B, 70B, 405B parameters). We fit exponential decay functions to the rate-distortion data points in Figure 16:

$$D(R) = \hat{C} e^{-\hat{\beta}R} + \hat{D}_0.$$

We optimize the parameters \hat{C} for initial value, $\hat{\beta}$ for decay rate, and \hat{D}_0 for distortion floor. See Appendix B.2 for further information.

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-40 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 computational cost of the system, as shown in Figure 20. We observe an increase in the number of total FLOPs-per-generation used by the Deep Research system as predictor LM size increases. We generally observe that larger compressors extract and provide more tokens of information in their compressions.

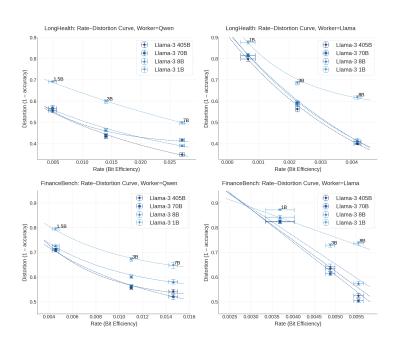


Figure 16: **Effects of predictor scaling.** Dotted lines show fitted exponential-decay functions for each predictor. Average root mean square error is 0.0468. Markers indicate compressor sizes (1B, 3B, 8B) in the LLAMA-3 compressor model family; vertical and horizontal bars denote standard errors.

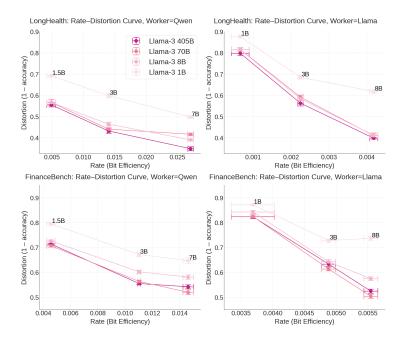


Figure 17: **Exploring the trade-off between compression and fidelity loss: rate-distortion curve.** The y-axis depicts distortion and the x-axis shows rate. *Distortion* measures how far our decompression is from the original data, which is quantified by 1 - accuracy. *Rate* measures the amount of information (as quantified by 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.

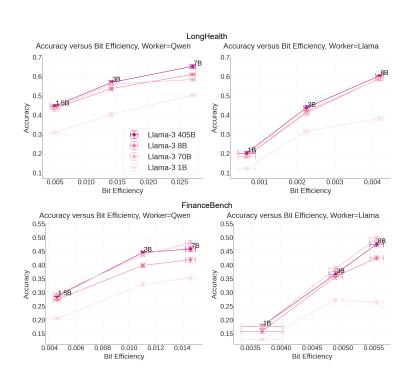


Figure 18: **Exploring the relationship between compression and accuracy.** The y-axis depicts accuracy and the x-axis shows bit efficiency. *Bit efficiency* is defined as the bits of mutual information encoded in each compression token. Markers indicate compressor sizes in the QWEN-2.5 (1.5B, 3B, 7B) and LLAMA (1B, 3B, 8B) compressor model family; vertical and horizontal bars denote standard errors.

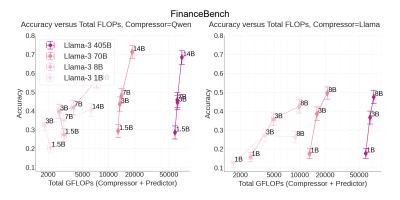


Figure 19: **QA** Accuracy versus total compute cost on FINANCEBENCH. In each panel, the y-axis shows the accuracy and the x-axis plots total compute cost in FLOPs-per-generation on a log-scale for (**Left**) QWEN-2.5, (**Right**) LLAMA-3 compressor LMs. Markers indicate compressor sizes in the QWEN-2.5 (1.5B, 3B, 7B) and LLAMA-3 (1B, 3B, 8B) compressor model family; vertical and horizontal bars denote standard errors.

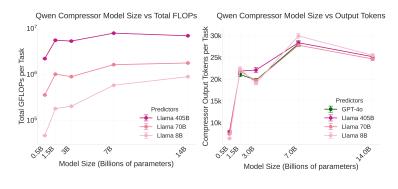


Figure 20: **Compressor model size versus compute and token usage.** In each panel, the x-axis shows the Qwen compressor size (in billions of parameters). (**Left**) Total GFLOPs per task grows with compressor model size, with larger predictors (Llama 405B, 70B, 8B) amplifying compute cost. (**Right**) Compressor output tokens per task, which remain relatively stable across predictors (GPT-40, Llama 405B, 70B, 8B), increase moderately with larger compressors.