

Fall 2024 Update of

Generative AI for Economic Research: LLMs Learn to Collaborate and Reason

Journal of Economic Literature*

by Anton Korinek[†]

November 2024

Abstract

Large language models (LLMs) have seen remarkable advances in speed, cost efficiency, accuracy, and the capacity to process larger amounts of text over the past year, enabling more advanced use cases compared to what I described in the initial version of 'Generative AI for Economic Research' (JEL, 2023). This article explores how these advancements support (i) new reasoning capabilities and (ii) new workspaces for interactive LLM collaboration such as Claude's Artifacts, ChatGPT's Canvas or Microsoft's Copilot. Moreover, it describes (iii) recent improvements in LLM-powered internet search. Incorporating these advances allows economist to achieve significant productivity gains in research. Additionally, I highlight new use cases in promoting research, such as automatically generated blog posts, presentation slides and interviews as well as podcasts via Google's NotebookLM.

*This paper covers the most recent advances in generative AI that are useful for our work as economists in Fall 2024. The original paper was published as "Generative AI for Economic Research: Use Cases and Implications for Economists," *Journal of Economic Literature* 61(4), pp. 1281-1317, December 2023. Check the journal website for future updates and for the original version of the paper, which provides an introduction to generative AI for economists. Sign up here for update notifications.

[†]I would like to thank Hemanth Asirvatham, Paul Bousquet, Kevin Bryan, Alan Chan, and Sam Manning for helpful comments on this update, and David Romer for wisely asking me to commit to producing semi-annual updates of this rapidly evolving material when publishing the original 2023 paper in the JEL. I have learned so much from working on these updates!

Reader’s guide:

For readers of the original December 2023 JEL article: Sections 1, 2, and 4 of the text below are new. Section 3 supersedes Section 3 of the original article and contains several new use cases indicated by "[All New in Fall 2024]" or "[Updated in Fall 2024]," including a new Section 3.7 on promoting research.

For readers of the June 2024 JEL update: Sections 1 and 4 are new, and Section 2 is largely new. Section 3 supersedes the June 2024 update and contains several new use cases indicated as such, including Section 3.7 on promoting research.

For new readers: The text below provides an overview of the state of generative AI in November 2024. As a complement, Sections 1 and 2 of the original December 2023 article offer an introduction to LLMs, and Section 4 of the original article discusses longer-term implications for economists.

1 Introduction [All New in Fall 2024]

Large language models (LLMs) have experienced remarkable progress since the original publication of “Generative AI for Economic Research” in December 2023 (Korinek, 2023a), characterized by significant gains in speed, cost efficiency, accuracy, and the capacity to process larger amounts of text. These advancements have not only enhanced existing capabilities but have also enabled entirely new ways of interacting with LLMs, demonstrating how substantial quantitative improvements can lead to paradigm shifts in functionality.

The paper starts with an updated overview of the landscape of LLMs, highlighting the rapid evolution and current state of leading models and their applications. All frontier AI labs have released new models in recent months. OpenAI still ranks first on a range of LLM benchmarks, with an updated version of its GPT-4o model. Google DeepMind has released a significantly updated version of Gemini 1.5 Pro 002 with a 2m token context window, making it able to process about 3000 pages of text simultaneously. Elon Musk’s xAI has shot to the #3 spot and is tightly integrated into the X (formerly Twitter) ecosystem. Claude 3.5 Sonnet excels at writing-related tasks. But there are now also excellent open-source LLMs from Meta and Alibaba that are close in capabilities to the models of the other four labs.

A key focus of this update are the new access modes enabled by the cumulative performance gains over the past year. Workspaces for interactive collaboration, such as Anthropic’s Claude Artifacts and OpenAI’s ChatGPT Canvas, Microsoft Copilot or Cursor, are changing how we interact with LLMs. They create an environment where users can iteratively develop and refine ideas, shifting away from static chat-style interactions towards a more dynamic, document-oriented collaboration. They allow users to work in tandem with LLMs, offering real-time feedback and allowing for iterative editing. Another example is a new generation of real-time voice assistants that can also assist with research tasks.

A second recent breakthrough are LLM-based reasoning capabilities, exemplified by OpenAI’s o1 series. I describe why traditional LLMs were not very good at reasoning and how work on a new generation of reasoning models is helping to overcome these barriers, enabling AI models to engage in multi-step problem-solving and logical deduction. This advancement opens new avenues for LLM use in economic research.

Finally, LLM-powered search, newly integrated into ChatGPT in November 2024 and also offered by startups like Perplexity and traditional search engines like Google, is starting to become a useful tool to provide up-to-date answers to questions that are grounded in facts found on the internet, together with the requisite citations—a crucial capability for researchers.

The paper also describes technical advances in the space of structured outputs and prompt caching as well as practical considerations like declaring LLM use and watermarking.

In addition to describing these new developments, I also cover several other novel examples and use cases for LLMs in the subsections below, many of which center around new ways of promoting research outputs—the current generation of LLMs is highly capable of processing the main insights of research papers and accurately translating and conveying them into a wide range of formats:

- Advanced mathematical derivations using o1-preview
- Sophisticated coding tasks using o1-preview
- Creating a ‘deep dive’ podcast of your research paper
- AI-powered search
- Creating presentation slides
- Drafting blog posts
- Conducting interviews

2 Advances at the Frontier of LLMs [Fully Updated in Fall 2024 Version]

Table 1 provides an overview of the top proprietary and open-source LLM providers as of November 4, 2024. The table is ranked by the score of each provider’s leading models in the LMSYS leaderboard (column 4), which pits randomly-selected pairs of LLMs against each other and employs user ratings to compile an Elo-like score for each

AI Lab	Best Model	Released	LMSYS	Tokens	Data Cutoff	URL
OpenAI	gpt-4o-latest	Sep 2024	1340	128k	Oct 2023	chat.com*
GoogleDM	Gemini 1.5 Pro 002	Sep 2024	1303	2m	Nov 2023	gemini.google*
xAI	Grok-2	Aug 2024	1290	128k	Mar 2024	x.ai / x.com*
Anthropic	Claude 3.5 Opus	Oct 2024	1286	200k	Apr 2024	claude.ai
Meta	Llama 3.1-405b	Jul 2024	1267	128k	Dec 2023	OS / meta.ai
Alibaba	Qwen 2.5-72b	Sep 2024	1263	128k	Sep 2024	OS (GitHub)

Table 1: Overview of top proprietary and open-source LLM providers according to their best model score in the LMSYS leaderboard.

Source: <https://lmarena.ai/?leaderboard>. See Chiang et al. (2024). Last accessed on Nov. 4th, 2024.

* denotes chatbots that can also access real-time information on the internet

model.¹² Columns 5 and 6 of the table list how many tokens (or syllables of text) the models can process simultaneously, and the date on which their training data cuts off. Models generally do not have knowledge of facts that occurred past this date, except if they have the capacity to access the internet. The last column lists the URLs under which the models can be accessed. The designation “OS” reflects that the model is available on an open-source basis, i.e., that it can be freely downloaded, run, and modified by researchers.

Several observations stand out from the table:

1. The field is moving fast—all six of the listed models have been released or updated in the past four months. In fact, older models quickly fall in the rankings. For example, if OpenAI had not released any model updates since April 2024, it would currently rank at the bottom of Table 1.
2. OpenAI continues to be the clear leader in the space with the latest update to its GPT-4o model.
3. The gap between the LMSYS scores of the top models is, however, relatively small. For example, using the Elo formula from footnote 1, OpenAI’s GPT-4o would win against the next-ranked Google DeepMind Gemini 1.5 Pro in 55.3%

¹The Elo-system was designed by the physicist Arpad Elo to rank chess players by their relative skills. It is designed so that a score difference of D points between two players (or LLMs) corresponds to the higher-ranked one having a probability of $1/(1+10^{D/400})$ of winning in a direct match-up.

²Like all ranking systems that condense the capabilities of candidates who differ across many dimensions into a single dimension, the LMSYS score offers only a partial and imperfect snapshot of LLM capabilities. I chose to use it for the overview table here because it has almost universal coverage of LLMs, it is updated in close to real-time, and it aggregates many different types of use cases when evaluating models. The LMSYS score is also highly correlated with other benchmarks of general LLM performance such as the MMLU.

GPT-4 Token Pricing Trend (\$/1M tokens) with LMSYS Scores

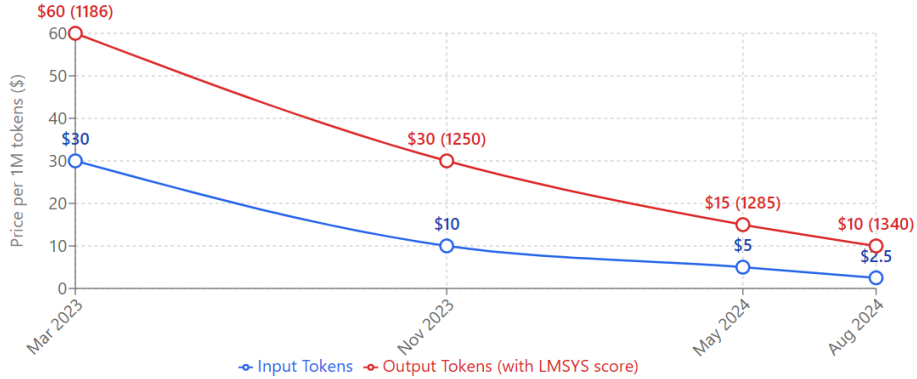


Figure 1: Decline in operating costs and quality improvement of GPT-4 models
Source: compiled by author.

of match-ups – hardly a decisive victory. In the words of Microsoft CEO Satya Nadella, LLMs are becoming “more of a commodity.”

4. The open-source models by Meta and Alibaba, listed in the bottom two rows of the table, have caught up and are now close to the frontier – a very different situation from a year ago, when open-source models were significantly behind proprietary models.
5. Chinese-made LLMs have ascended particularly rapidly, as reflected in the last row. Since LMSYS rankings are based on mostly Western user preferences, they may in fact understate the capabilities of Qwen 2.5. What is notable is that the model ranks so close to the best Llama 3.1 model, even though its parameter count (72bn) is just a fraction of the latter’s (405bn).

Speed of Progress To provide data on the speed of progress, I list a few quantitative indicators from OpenAI’s series of GPT-4 models as an example. Since the initial release of GPT-4 in March 2023—less than two years ago—the models’ context window size has increased 16-fold, allowing it to process far more content at once, the quality of the model’s responses has significantly improved (the current LMSYS score of the original GPT-4 is only 1186), and the speed of output generation has increased 3-fold. Figure 1 illustrates the steep decline in the cost of reading and generating text (input and output tokens) of GPT-4 level models since March 2023—by 92% and 83% respectively—even though their LMSYS score steadily improved. See Ho et al. (2024) for a detailed examination of algorithmic progress in LLMs.

In the following, I describe the leading LLM products of the frontier labs listed in Table 1. Readers who are most interested in the conceptual advances may want to skip to Section 2.1 describing advances in reasoning, followed by a description of novel collaborative access modes for LLMs.

Each of the labs listed in Table 1 offers families of models of different sizes that reflect different trade-offs between model performance, speed, and cost. Larger models are more “intelligent” and generally offer better performance and greater capabilities, but they also require more computational resources and take longer to process requests, making them more expensive. Smaller models, on the other hand, are faster and more cost-effective, but may not provide the same level of quality in their outputs. This allows users to consider their specific needs and budget when choosing the appropriate model size for their applications.

Leading Proprietary Models

The first four labs listed in the table offer proprietary models, which means that their models can only be accessed via the labs’ computer servers. They do not share the source code, architecture, and model weights of their LLMs but allow users to access them via chatbots, web-based experimentation platforms, or APIs, subject to the certain conditions and controls.

OpenAI’s GPT-4o model, last updated in September 2024, continues to lead the market for LLMs in terms of both general capabilities and popularity. (OpenAI’s o1 model, also released in September 2024, demonstrates new advances in LLM-based reasoning that are extremely valuable for research, as described in Section 2.1, but less valuable for general use, resulting in a lower LMSYS score than GPT-4o.) GPT-4o is an evolution of the original GPT-4 model of March 2023 that is considerably smaller, faster, cheaper, and more capable, as shown in Table 1. The suffix “o” stands for “omni” to reflect that the model can process text, images, and sound. GPT-4o also offers workspace extensions that make it easy to interact collaboratively with the model, including Canvas and Advanced Data Analysis (described in Sections 2.2 and 3.5 below), and the ability to search the web (described in Section 2.3 below). GPT-4o is subject to usage limits in the free version of ChatGPT. The model’s smaller sibling, GPT-4o-mini, is faster and 94% cheaper but would still rank in the number 5 spot in Table 1, making it an attractive choice for bulk data processing.

Google DeepMind’s Gemini series of LLMs carries the distinction of having a 2m token context window—the longest of all publicly available LLMs, which allows it to simultaneously process a few dozen books or several hundred papers. This offers new use cases—for example, it allows researchers to upload a significant body of their work all at once and process queries based on it, or to simultaneously process videos or large corpora of images. The most powerful version is currently *Gemini 1.5 Pro 002*, updated September 2024, and is only available to paying subscribers. It also comes with a smaller sibling, *Gemini 1.5 Flash*, which offers greater speeds at lower cost but slightly lower performance. Gemini is also accessible via an eponymous chatbot that can access the internet to include real-time information in its responses and allows users to cross-check results and follow links to its sources.

xAI’s Grok-2 is a relative newcomer in the LLM space. xAI was founded by Elon

Musk in March 2023, and its Grok-2 model has ascended into the top-3 a bit over a year after the lab’s founding, offering state-of-the-art performance in most tasks. xAI benefits from its close relationship with X, formerly Twitter, which Elon Musk took over in 2022 and uses for training data. This allows Grok-2 to be up-to-date on news. Moreover, it distinguishes itself by not imposing any limits on user queries, following instructions and generating controversial content that many may consider unethical, reflecting Elon Musk’s “free-speech absolutism.”

Anthropic’s Claude 3.5 Sonnet, by contrast, brands itself as being a helpful, honest, and harmless assistant, employing a process called constitutional AI to train the LLM to follow a set of high-level ethical principles (Bai et al., 2022). Claude is the model I use most for writing as I like its succinct, elegant and insightful writing style. The latest update, released in October 2024, ranks the model in the top spot of several technical benchmarks. Claude 3.5 has a context window of 200k tokens, which makes it able to process about 150,000 words in one go—for example, several academic papers. Anthropic pioneered many LLM applications and access modes, for example the chat-bot format before ChatGPT or, more recently, interactive collaboration in workspaces called “Artifacts” (see Section 2.2) and autonomous computer use (see Section 2.2.3). Another recent update, PDF support (beta), allows Claude to visually process PDF documents uploaded in its chat interface or via its API so that it can read figures and graphs in PDFs, which is highly valuable in processing academic papers or other documents that contain visual information such as charts or figures.

Table 2 compares the cost of the models listed above—it has become industry practice for leading labs to offer two main models: a more expensive frontier model and a cheaper model well-suited for bulk data processing. xAI is only offering beta access to its models as of November 2024. OpenAI and Anthropic offer a 50% discount for batch processing that may be executed at a delay when their servers face a lower load; all three labs offer discounts for cached content. For Google DeepMind, the first 50 requests per day for its Pro model and 1500 requests per day for its Flash model are free, and using more than 128k tokens incurs double the cost displayed in the table.³

Leading Open-Source Models

The top LLM providers that release their models open source are listed in the last two rows of Table 1. Their models are freely available to download, use, modify, and distribute.⁴ This offers several benefits for economic research. Firstly, the transparency of

³Up-to-date pricing information for the three labs is available at <https://openai.com/api/pricing/>, <https://ai.google.dev/pricing> and <https://www.anthropic.com/pricing>.

⁴More precisely, the models are “open weights,” which means that the weights and software to run inference on the LLM can be freely downloaded but not the training source code and data. Frequently, there are additional restrictions. For example Meta imposes limits on the large-scale commercial use of their models. The AI lab TogetherAI has a project named RedPajama to reproduce and distribute an open source version of the LLaMA dataset.

Model (cost per 1M tokens)	Input Cost	Output Cost
OpenAI GPT-4o	\$2.5	\$10
OpenAI GPT-4o-mini	\$0.15	\$0.60
Google DeepMind Gemini 1.5 Pro	\$1.25	\$5
Google DeepMind Gemini 1.5 Flash	\$0.075	\$0.30
Anthropic Claude 3.5 Sonnet	\$3	\$15
Anthropic Claude 3.5 Haiku	\$1	\$5

Table 2: Price comparison for input and output tokens across leading models.

Source: compiled by author.

open-source models allows researchers to examine the underlying architecture, enabling them to better understand the model’s structure. Secondly, open-source projects allow anybody to innovate upon the model. This can help accelerate the development of LLMs tailored to specific needs. Thirdly, if researchers have access to low-cost computing resources, they can leverage open-source models for their work without incurring financial costs. Fourthly, open-source models that are operated locally offer significant privacy benefits as sensitive data does not need to be channeled over the internet to be processed on the servers of proprietary model providers. Finally, open-source models allow for greater reproducibility, which is helpful for ensuring scientific integrity in research as it enables other researchers to verify and build upon the reported results. These benefits make open-source language models an attractive choice for researchers seeking to harness the power of natural language processing in their work.

From an economic perspective, open-source models are highly beneficial as they freely distribute the economic social surplus created by LLMs and stimulate innovation (Korinek and Vipra, 2025). On the downside, as open-source LLMs become more capable, they also pose growing safety risks (Anderljung et al., 2023).⁵

Meta’s LLaMA 3.1 is currently the most powerful series of open source models, which have been downloaded more than 350,000 times so far. The most powerful publicly available model is currently Llama 3.1-405B which features 405B parameters. However, as of November 2024, LLaMA is transitioning to version 3.2, offering multi-modal versions with 11B and 90B parameters as well as text-only versions with 3B and 1B parameters, which can be operated on many devices. All available Llama models are also accessible on leading cloud computing platforms, including Microsoft Azure, AWS, and Hugging Face. NVIDIA released a fine-tuned version of Meta’s 70B parameter model as Llama-3.1-Nemotron-70B-Instruct, which has obtained an LMSYS score of 1271 despite its smaller size.

⁵For example, LLaMA has already allowed researchers to construct adversarial attacks that circumvent the safety restrictions of all the LLMs listed above (Zou et al., 2023). Seger et al. (2023) discuss the pros and cons of open-sourcing LLMs as well as intermediate solutions between proprietary and fully open source models that may be desirable as LLMs become more capable and pose growing safety risks.

Alibaba’s Qwen 2.5 (short for Tongyi Qianwen, which translates to “Unified Thousand Questions”) has made rapid progress and reached a spot in Table 1 in Sept 2024, even though Alibaba, being located in China, is subject to export controls on cutting-edge GPU chips that are crucial for training LLMs. The Qwen 2.5 series consists of 100 open-source models with parameter sizes ranging from 0.5B to 72B, including multimodal models and excellent LLMs specialized in math and coding that reach state-of-the-art performance.

2.1 Advances in Reasoning [All New in Fall 2024 Version]

One of the most significant advances in recent months is that LLMs are becoming better at reasoning. Traditional LLMs generate output via token-by-token prediction, as described, e.g., in Section 2 of the originally published version of this paper (Korinek, 2023a). Although this basic architecture has proven surprisingly powerful, it makes it hard for basic LLMs to go back in the text that they have already generated to reason about it and iteratively improve it, as humans do when they write. A good analogy is that token generation by LLMs proceeds like a human’s stream-of-consciousness. This makes it easy for such systems to emulate what Kahneman (2011) called system-1 thinking but difficult to perform cognitive tasks that correspond to system-2 thinking and require reasoning. For example, a famous test that poses no problem for fourth-graders but has regularly tripped up even the most advanced LLMs before o1 was the so-called strawberry test: asking an LLM “How many R’s are there in strawberry?” typically delivers false responses.⁶

Aware of these limitations, researchers have worked hard on finding ways to enable LLMs to become better at reasoning (see, e.g., the surveys by Huang and Chang, 2023; Plaat et al., 2024). An influential mechanism to obtain better-reasoned results has been chain-of-thought prompting, which instructs LLMs to proceed step-by-step when generating responses to a prompt. This technique has delivered significant performance gains by guiding LLMs to break down complex questions into smaller logical steps that are easier to accomplish—akin to a student who performs better on an exam when asked to report his intermediate steps. Wei et al. (2022) show that chain-of-thought prompting considerably improves LLM performance on a range of arithmetic, commonsense, and symbolic reasoning task. For example, when given a question like “The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?” they show that OpenAI’s GPT-3 from 2020 failed (“The answer is 27.”) but succeeds when guided to reason through the calculation, producing the answer: “The cafeteria had 23 apples originally. They used 20 to make lunch. So

⁶One of the reasons for this rather basic failure is that LLMs encode text not in letters, as we do in the English language, but in tokens that correspond to syllables or words and that imply that the spelling is not directly observable for LLMs when they process text. This implies that the model needs to reason about the English spelling corresponding to the underlying tokens. Dell’Acqua et al. (2023) use the term “jagged frontier” to observe that LLMs easily perform some tasks but fail at other tasks that are of seemingly similar difficulty for humans.

they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9.”

When academics reason about a novel research problem, an additional strategy that they employ is to perform a sort of tree search: they generate different hypotheses, evaluate them, pursue the most promising ones and refine them further. Recent advances in LLM-based reasoning attempt to emulate this process. For example, Yao et al. (2024) propose what they call a tree-of-thoughts technique, which extends chain-of-thought prompting by generating multiple intermediate steps or pro-verbial “thoughts” at each stage of the reasoning process. This allows LLMs to explore different paths of reasoning, evaluate their potential, and select the most promising ones to continue—much like, for example, a chess engine evaluating different moves. This approach has shown particular promise in solving complex reasoning tasks that require planning and strategic thinking. In spring 2024, Anthropic introduced a feature that allows Claude to engage in short episodes of reasoning that are hidden from the user behind a message like “Thinking deeply...” or “Ruminating...” before generating output. This led to clear performance gains, but with little fanfare.

OpenAI’s o1 series of models, released on September 12, 2024, is the first that is explicitly designed for LLM-based reasoning (OpenAI, 2024).⁷ Although an official description of the model’s architecture is not publicly available, OpenAI seems to have employed reinforcement learning to hone the model’s automated use of the two techniques described in the preceding two paragraphs: it employs a chain-of-thought technique to break down complex steps into simpler ones while also employing a form of tree search to attempt different approaches to solving a problem and to recognize and correct mistakes. Unlike earlier LLMs, o1 models react to prompts by first generating reasoning tokens that are hidden from the user—akin to a simulated inner monologue—as the model “thinks” through the problem at hand. Depending on the question, this may take from a few seconds to several minutes. Once the hidden thinking process is finished, the model generates a response for the user that summarizes the outcome of the reasoning process. This mechanism has enabled o1 to achieve significant gains in reasoning compared to GPT-4o, which itself was a leader in this category before o1’s release. However, Mirzadeh et al. (2024) document that even the reasoning capabilities of o1 still have significant shortcomings, for example, that small changes in numbers or the addition of irrelevant information reduce model performance.

The o1 series consists of several models. In Sept. 2024, OpenAI released a preview version of the full model, **o1-preview**, as well as a smaller and more cost-efficient model, **o1-mini**, which is particularly well-suited for math and coding applications and better than **o1-preview** in some of these applications. The full version o1 is the most powerful of the three and expected later in fall 2024. Figure 2 visualizes the advances compared to GPT-4o on three benchmarks: in the American Invitational Mathematics Examina-

⁷Originally nick-named “Q*” and later “strawberry,” its creators argued that the system’s architecture is sufficiently different from its earlier GPT series to merit a new name that is simply an abbreviation of “OpenAI 1.”

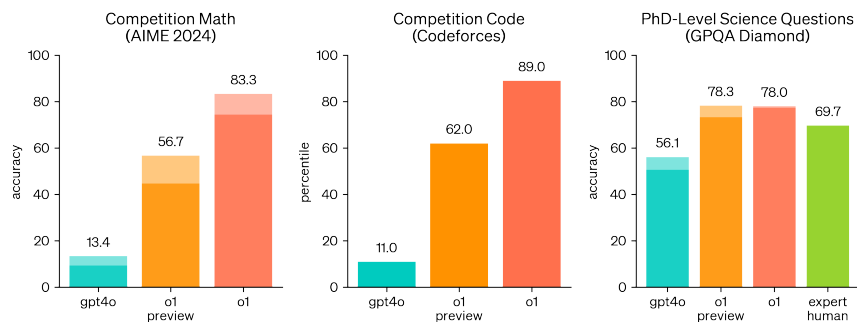


Figure 2: o1’s performance on reasoning benchmarks. Source: OpenAI (2024).

Note: Solid bars show one-shot accuracy; shaded regions show performance of a majority vote of 64 responses.

tion (AIME), the qualifying exam for the US Mathematical Olympiad, o1 performed at a level corresponding to the top-500 contestants; in the Codeforces programming competition, it performed at the 89th percentile of participants; in the hardest (diamond) category of the Graduate-Level Google-Proof Q&A benchmark, which tests for complex reasoning abilities in biology, chemistry, and physics, it outperformed PhD-level human experts.

Interestingly o1 also fundamentally changes the economics of operating frontier AI models: Under the traditional LLM paradigm, the computational cost of training models exceeded the cost of generating output (inference) by orders of magnitude. By contrast, o1 relies heavily on generating reasoning tokens during inference, and—in analogy to the scaling laws for training LLMs discussed in Section 2 of the original article—the model’s responses improve predictably in the amount of reasoning tokens employed (OpenAI, 2024). This implies far greater variable costs compared to traditional LLMs. At present, o1 models are only available to OpenAI’s paying customers, with limits on the number of queries that can be performed per week.

The following reproduces an experience of mine with o1-preview for a task that was far beyond the capabilities of earlier LLMs:

Chat 1 (o1-preview): The following is a no-arbitrage relationship between an interest rate on new long-term debt in period t , $R_{N,t}$, and expected future short-term rates, R_{t+s} :

$$\frac{1}{R_{N,t} + \delta} = E \sum_{i=0}^{\infty} \left[\frac{(1 - \delta)^i}{\prod_{s=0}^i (1 + R_{t+s})} \right] \quad (1)$$

Can you log-linearize this expression around the steady state value of the interest rates, which we denote by R^* , and which is the same for both interest rates? Denote deviations from steady state with lower-case variables.

Thought for 53 seconds [drop-down menu summarizing reasoning steps]

To log-linearize the given no-arbitrage relationship around the steady state R^* , we'll start by rewriting the original equation and then expand it using first-order approximations.

Original Equation:

$$\frac{1}{R_{N,t} + \delta} = E_t \left[\sum_{i=0}^{\infty} \frac{(1 - \delta)^i}{\prod_{s=0}^i (1 + R_{t+s})} \right]$$

Steady State Analysis:

At steady state, $R_{N,t} = R^*$ and $R_{t+s} = R^*$. The equation simplifies to:

$$\frac{1}{R^* + \delta} = \frac{1}{1 + R^*} \cdot \frac{1}{1 - k}$$

where $k = \frac{1-\delta}{1+R^*}$.

Log-Linearization Steps:

1. **Left-Hand Side (LHS):** Expand $\frac{1}{R_{N,t} + \delta}$ around R^* :

$$\frac{1}{R_{N,t} + \delta} \approx \frac{1}{R^* + \delta} \left(1 - \frac{r_{N,t}}{R^* + \delta} \right)$$

2. **Right-Hand Side (RHS):**

Approximate the denominator and numerator for small deviations:

- **Denominator:**

$$\prod_{s=0}^i (1 + R_{t+s}) \approx (1 + R^*)^{i+1} \exp \left(\frac{1}{1 + R^*} \sum_{s=0}^i r_{t+s} \right)$$

- **Numerator:**

$$(1 - \delta)^i \approx e^{-i\delta}$$

- **Combining Terms:**

$$\frac{(1-\delta)^i}{\prod_{s=0}^i (1+R_{t+s})} \approx \frac{k^i}{(1+R^*)} \left(1 - \frac{1}{1+R^*} \sum_{s=0}^i r_{t+s} \right)$$

3. **Simplify RHS Sum:**

$$E_t \left[\sum_{i=0}^{\infty} k^i \left(1 - \frac{1}{1+R^*} \sum_{s=0}^i r_{t+s} \right) \right] = \frac{1}{1-k} - \frac{1}{(1+R^*)(1-k)} \sum_{s=0}^{\infty} k^s r_{t+s}$$

4. **Equate LHS and RHS:**

$$\frac{1}{R^* + \delta} \left(1 - \frac{r_{N,t}}{R^* + \delta} \right) = \frac{1}{R^* + \delta} \left(1 - \frac{1}{1+R^*} \sum_{s=0}^{\infty} k^s r_{t+s} \right)$$

5. **Simplify and Solve for $r_{N,t}$:**

$$r_{N,t} = \frac{R^* + \delta}{1+R^*} \sum_{s=0}^{\infty} k^s r_{t+s}$$

6. **Express $\frac{R^* + \delta}{1+R^*}$ in Terms of k :** Since $1-k = \frac{R^* + \delta}{1+R^*}$, we have:

$$r_{N,t} = (1-k) \sum_{s=0}^{\infty} k^s r_{t+s}$$

Final Log-Linearized Expression:

$$r_{N,t} = (1-k) \sum_{s=0}^{\infty} k^s r_{t+s}$$

This equation shows that the deviation of the long-term interest rate from its steady state is a weighted sum of expected future deviations of the short-term interest rate, where the weights decrease geometrically with factor k .

Answer:

Yes; the log-linearized expression is $r_{N,t} = (1-k) \sum_{s=0}^{\infty} k^s r_{t+s}$ where $k = \frac{1-\delta}{1+R^*}$.

Chat 1 (o1-preview) ctd.: Great. Can you express this as a forward difference equation in the style of $r_{N,t} = coeff1 * r_{N,t+1} + coeff2 * r_t$?

Thought for 22 seconds [drop-down menu summarizing reasoning steps]

Yes, we can express the log-linearized relationship as a forward difference equation of the form:

$$r_{N,t} = k r_{N,t+1} + (1 - k) r_t$$

where $k = \frac{1-\delta}{1+R^*}$.

Derivation

[full derivation omitted for space reasons]

Final Answer

Yes; the log-linearized expression can be written as:

$$r_{N,t} = k r_{N,t+1} + (1 - k) r_t$$

where $k = \frac{1-\delta}{1+R^*}$.

This example illustrates that even o1-preview, which is less capable than the ultimate o1, can perform complex mathematical derivations that can save researchers significant amounts of time. I should note that the prompt in Chat 1 contains an important hint for how to efficiently approach the problem, that the steady state of the two interest rate variables will be the same. Without this hint, the system struggled. Even with the hint, the model occasionally produced mistakes in derivations such as the one requested in Chat 1 but was frequently able to correct mistakes when I pointed them out. Moreover, the derivation steps reported in the example appear at times a bit clumsy—it looks like this LLM is still “in its infancy.”

According to Jones (2024), o1 has impressed many scientists with its complex problem-solving performance, which is particularly beneficial in disciplines that rely on mathematical modeling. While o1 can reason through problems step-by-step, they also noted that these chains of thoughts can increase the risk of "hallucinations."

For complex coding tasks, o1 excels at following, generating, and debugging intricate code structures. Its reasoning capabilities enable it to tackle problems that require deeper contextual understanding and strategic thinking than earlier LLMs, making it particularly valuable for academics working on sophisticated coding projects. The following example documents o1-preview’s response to a request to code a solution of the Ramsey growth model, which earlier LLMs could not do (as documented in the December 2023 version of the paper):

Chat 2 (o1-preview): Can you write python code to solve the Ramsey growth model for standard parameter values?

Thought for 34 seconds [drop-down menu summarizing reasoning steps]

Certainly! Below is Python code that solves the Ramsey growth model using standard parameter values. The code sets up the differential equations representing the Ramsey model, computes the steady-state values, and then uses a shooting method to find the optimal initial consumption that leads to convergence to the steady state. It then simulates the model over time and plots the capital and consumption per capita.

```
1 import numpy as np
2 from scipy.integrate import solve_ivp
3 from scipy.optimize import root_scalar
4 import matplotlib.pyplot as plt
5
6 # Parameters
7 alpha = 0.3      # Output elasticity of capital
8 rho = 0.04       # Time preference rate
9 delta = 0.05     # Depreciation rate
10 theta = 2.0     # Coefficient of relative risk aversion (CRRA)
11 n = 0.01        # Population growth rate
12
13 # Steady-state capital per capita
14 k_star = (alpha / (rho + delta)) ** (1 / (1 - alpha))
15 # Steady-state consumption per capita
16 c_star = k_star ** alpha - (n + delta) * k_star
17
18 print(f"Steady-state capital per capita (k*): {k_star:.4f}")
19 print(f"Steady-state consumption per capita (c*): {c_star:.4f}")
20
21 # Initial capital per capita (start below steady state)
22 k0 = 0.5 * k_star
23
24 # Time span for the simulation
25 t_span = [0, 100] # From t=0 to t=100
26 t_eval = np.linspace(t_span[0], t_span[1], 1000) # Time
27 # points for evaluation
28
29 # Differential equations of the Ramsey model
30 def ramsey(t, y):
31     k, c = y
32     dkdt = k ** alpha - c - (n + delta) * k
```

```

32     dcdt = (1 / theta) * (alpha * k ** (alpha - 1) - rho -
33         delta) * c
34     return [dkdt, dcdt]
35
36 # Shooting method to find the optimal initial consumption c0
37 def shooting_function(c0):
38     y0 = [k0, c0]
39     sol = solve_ivp(ramsey, t_span, y0, t_eval=t_eval, method
40         ='RK45')
41     k_T = sol.y[0, -1]
42     return k_T - k_star # We want k(T) to be close to k_star
43
44 # Initial guesses for c0
45 c0_lower = 0.1 * c_star
46 c0_upper = 2.0 * c_star
47
48 # Use root_scalar to find the root of the shooting function
49 res = root_scalar(shooting_function, bracket=[c0_lower,
50     c0_upper], method='brentq')
51 if res.converged:
52     c0 = res.root
53     print(f"Optimal initial consumption per capita (c0): {c0
54         :.4f}")
55 else:
56     raise ValueError("Root-finding did not converge to a
57         solution.")
58
59 # Simulate the model with the optimal initial consumption
60 y0 = [k0, c0]
61 sol = solve_ivp(ramsey, t_span, y0, t_eval=t_eval, method='
62     RK45')
63 k = sol.y[0]
64 c = sol.y[1]
65 t = sol.t
66
67 # Plotting the results
68 plt.figure(figsize=(12, 5))
69 # Plot for capital per capita over time
70 plt.subplot(1, 2, 1)
71 plt.plot(t, k, label='Capital per capita (k)')
72 plt.axhline(y=k_star, color='r', linestyle='--', label='
73     Steady-state k*')
74 plt.xlabel('Time')
75 plt.ylabel('Capital per capita (k)')
76 plt.title('Capital per Capita Over Time')

```



```

69 plt.legend()
70 # Plot for consumption per capita over time
71 plt.subplot(1, 2, 2)
72 plt.plot(t, c, label='Consumption per capita (c)')
73 plt.axhline(y=c_star, color='r', linestyle='--', label='
    Steady-state c*')
74 plt.xlabel('Time')
75 plt.ylabel('Consumption per capita (c)')
76 plt.title('Consumption per Capita Over Time')
77 plt.legend()
78 plt.tight_layout()
79 plt.show()

```

Explanation:

- Parameters: The code uses standard parameter values commonly found in the literature.
- Steady-State Calculation: It computes the steady-state values of capital and consumption per capita.
- Differential Equations: The ramsey function defines the system of differential equations for capital accumulation and consumption growth.
- Shooting Method: The shooting_function uses a shooting method to find the optimal initial consumption c_0 that ensures convergence to the steady state.
- Simulation: It simulates the model over the specified time span using solve_ivp.
- Plotting: The results are plotted to show the paths of capital and consumption per capita over time. Output:

The code will display the steady-state values and the optimal initial consumption. It will then produce two plots showing how capital per capita and consumption per capita evolve over time, converging to their steady-state values.

Notes:

The shooting method is essential because the Ramsey model is saddle-path stable, meaning there is a unique consumption path that leads to the steady state from a given initial capital stock. If the root-finding does not converge, you may need to adjust the initial guesses for c_0 or check the parameter values. This code provides a complete solution to the Ramsey growth model using standard parameter values and should serve as a solid foundation for further analysis or extensions of the model.

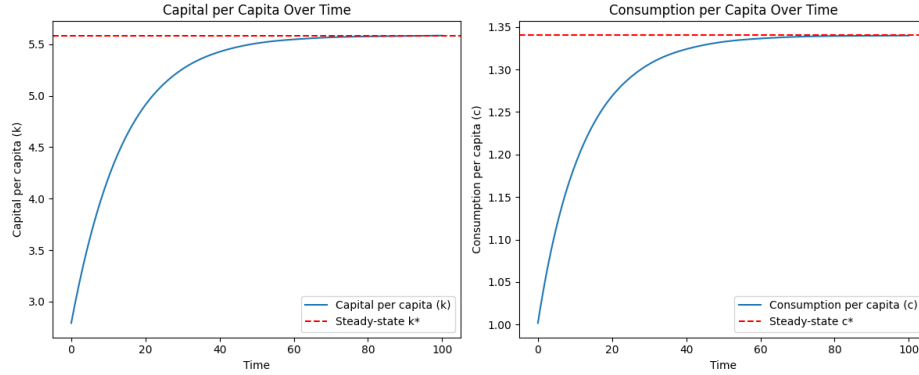


Figure 3: Solution to Ramsey model produced by o1

The code is written elegantly and explained well. The output produced by the generated code is shown in Figure 3.

Meta recently published a paper on an approach called "Thought Preference Optimization" that demonstrates that the benefits of LLMs thinking can extend even further (Wu et al., 2024). Their method shows that optimizing thought generation before responding improves performance across a broad range of applications, from marketing and content creation to general knowledge tasks. It is likely that this approach will be implemented by future versions of Meta's open-source models.

The Road to LLM Agents A key strategic goal of frontier AI labs is to evolve LLMs into autonomous AI agents—systems that can maintain objectives across multiple exchanges, plan sequences of actions, and independently pursue specified goals (see e.g. Heikkilä, 2024). While current LLMs primarily react to immediate prompts, agents would proactively gather information, formulate plans, and take actions to accomplish tasks. The advances in reasoning capabilities described above represent a crucial step toward this goal, as autonomous agents require the ability to plan multi-step actions and revise strategies based on outcomes.

Two other key requirements for agency are already emerging: first, the ability to maintain coherent objectives over time through long-term memory and expanded context windows, as discussed earlier in this section; and second, the capability to interact with external tools and APIs to gather information and take actions in the world. The latter developments will be covered in the next subsection on access modes, particularly in the context of LLMs' autonomous "computer use" capabilities, and in the ensuing subsection on LLM-powered search, which gives LLMs real-time internet access. As these components come together, we may see LLMs evolve from passive tools into more active research collaborators.

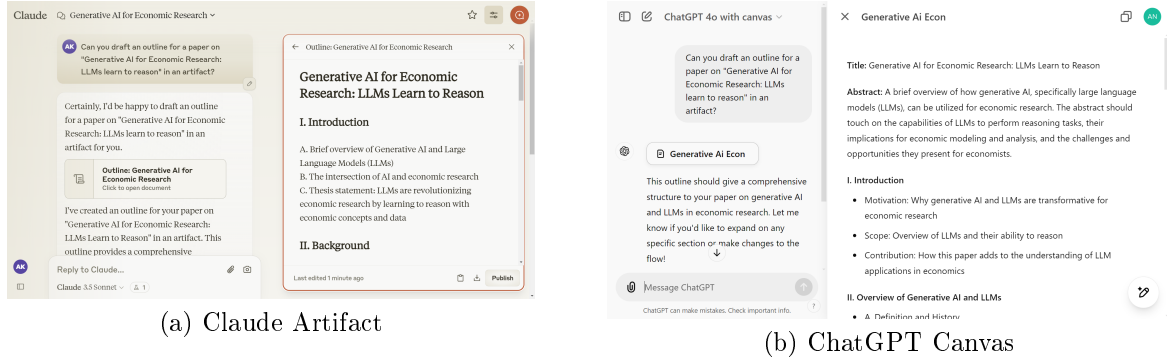


Figure 4: Anthropic’s and OpenAI’s workspaces for interactive LLM collaboration

2.2 Access Modes to Frontier LLMs [Fully Updated in Fall 2024 Version]

Several new access modes to frontier LLMs have emerged over the course of 2024—in addition to the traditional text-based interaction modes. The most notable are workspaces for interactive LLM collaboration, real-time voice assistants, and the emerging autonomous “computer use” capabilities of LLMs.

2.2.1 Workspaces for interactive LLM collaboration [All New in Fall 2024 Version]

Enabled by longer context windows and greater speeds, LLM providers have introduced innovative workspace environments in recent months that allow users to interactively collaborate with LLMs on content, as summarized in Table 3. Anthropic led the way with the introduction of Artifacts for Claude in June 2024 (left panel in Figure 4), followed by OpenAI’s launch of Canvas for ChatGPT in October 2024 (right panel). Concurrently, developers of office package such as Microsoft and Google have increasingly integrated LLM capabilities into their existing workspaces. The resulting products allow users to move beyond the limitations of traditional chat interfaces and to interact with AI assistants in a more dynamic and collaborative environment for content creation, editing, and interaction. These features represent a significant leap forward in human-AI collaboration, providing dedicated spaces for users to credit, edit, and build upon AI-generated content in real-time.

Artifacts in Claude can be activated by clicking at the user button at the bottom left and choosing “Settings,” which opens a menu that lets the user check the option “Enable artifacts.” When Claude finds it useful, or when the user explicitly asks for an artifact, the chatbot opens a dedicated output window to the right of the chat interface where the user can instantly view and interact with the generated content. This feature

Table 3: Interactive Workspaces for LLM Collaboration

Workspace	Key Features
Anthropic Claude Artifacts	<ul style="list-style-type: none"> • Dedicated output window • Supports text, code, flowcharts, SVG graphics, websites, dashboards • Real-time refinement and modification • Sharing and remixing capabilities
ChatGPT Canvas	<ul style="list-style-type: none"> • Separate collaboration window • Text editing and coding capabilities • Options for edits, length adjustment, reading level changes • Code review and porting features
OpenAI Advanced Data Analysis	<ul style="list-style-type: none"> • Data upload and analysis • Visualization capabilities • Python code execution in backend • Error correction and refinement
Claude Analysis Tool	<ul style="list-style-type: none"> • Fast exploratory data analysis • Interactive visualizations with real-time adjustments
Google NotebookLM	<ul style="list-style-type: none"> • Document upload for research grounding • Quick summarization and questioning • Citation and quote provision • "Deep dive conversation" podcast generation
Microsoft Copilot	<ul style="list-style-type: none"> • Integration with Microsoft 365 products • Assistance in Word, Excel, PowerPoint, etc. • Data analysis, formula construction
Google Gemini for Workspace	<ul style="list-style-type: none"> • Integration with Google's office suite • Assistance in Docs, Sheets, Slides, Gmail
Cursor AI Code Editor	<ul style="list-style-type: none"> • AI-assisted coding • Code suggestions and queries • Optimization recommendations • Debugging assistance • Real-time collaboration

supported a range of different outputs, including text (as in Figure 2a), code snippets, flowcharts, SVG graphics, websites, and interactive dashboards, which are all worth trying out. The user can interact with the artifact by asking the chatbot to refine or modify the output in line with her instructions, allowing for rapid prototyping and iteration. The feature also allows users to publish and share artifacts with other users who can subsequently remix them, enabling easy collaboration.

Canvas in ChatGPT can be activated for paying users by selecting “GPT-4o with canvas” from the model selection menu at the top left of the screen. It is based on a similar concept as Claude Artifacts, opening in a separate window that allows users and ChatGPT to collaborate on writing and coding projects, but also offers some unique features that make it more powerful. The interface allows users to edit the content in the canvas like in a text editor or to select specific paragraphs and provide instructions on how to change the text or ask questions about it. Moreover, for text, the button at the bottom right of the canvas (Figure 2b) includes options to ask the LLM for suggested edits, to adjust the length of the content (shorter or longer) and the reading level (from kindergarten to graduate level), or to add a “final polish.” For code, the button includes options to review and comment on the code, to port it to a different language, to fix bugs, and to add comments or logs—all while keeping the entire content in mind.

Advanced Data Analysis in ChatGPT has been available in a rudimentary form since mid-2023 (originally named “Code Interpreter,” explored in Examples 27 and 32 below) but has been significantly improved after the release of GPT-4o in 2024, turning it into a collaborative workspace where the main focus is on interacting with the data. It allows users to upload data in a variety of formats, for example, spreadsheets, and analyze, visualize and process the data in a multitude of ways. On the backend, ChatGPT writes and executes python code to perform the necessary operations, implying that a wide range of analyses can be performed, limited only by the “intelligence” of the LLM that is writing the code. Simple routine tasks are usually no problem. For more complicated forms of analysis, Advanced Data Analysis may make errors but is frequently able to correct them automatically in response to python error messages or the user’s requests.

Claude Analysis Tool is Anthropic’s response to ChatGPT’s Advanced Data Analysis, using JavaScript rather than python as its underlying engine. Released in October 2024, it allows Claude to visualize and explore data in an elegant, intuitive, and interactive manner. This makes the tool most valuable for quick data explorations and visualizations. For example, I created Figure 1 using the Claude Analysis tool. However, it is less suitable for specialized econometric analysis than ChatGPT’s data analysis as it lacks the wide range of python libraries offering econometric tools.

NotebookLM by Google represents a more specialized approach to AI-assisted research, facilitating background research and literature reviews, leveraging the long context window of Google's Gemini models. Its key feature is to allow users to upload a wide range of relevant research papers or documents that ground the system AI in a specific corner of the literature and enables them to quickly summarize uploaded documents, ask targeted questions about the content, and brainstorm on the materials provided. Importantly for academic work, NotebookLM provides citations and relevant quotes from original sources during exchanges, supporting the rigorous documentation required in research.

In September 2024, NotebookLM introduced a groundbreaking feature called "Deep dive conversation," which allows users to generate podcasts based on the provided materials. These podcasts feature an insightful conversation between two virtual agents who explain the essence of the material in simple and clear terms for non-experts, narrated in a highly engaging manner. For research use, this is one of the most impressive capabilities that have emerged in recent months. It offers a novel way to disseminate complex economic research to a broader audience, potentially increasing the impact and accessibility of economic studies. The tool is available for free at <https://notebook.google>.

Below is the beginning of a sample podcast generated based on uploading the original version of this research paper (Korinek, 2023a):

Chat 3 (NotebookLM): Generate deep dive conversation.
[Attachment: w30957.pdf]

Male host: Okay, so you're interested in the economics of AI...

Female host: Yeah.

Male host: ...specifically, all the buzz about these large language models, LLMs.

Female host: *mmhmm*

Male host: You're in the right place, because we're doing a deep dive into how these things actually work, what they could do for economists like you right now, and what the future might hold.

Female host: And it's a future that's changing rapidly. It seems like every day there's some new breakthrough in what these models can do.

Male host: Well, let's unpack that a bit.

Female host: *mhm*

Male host: Imagine having a research assistant who can not only brew you a perfect cup of coffee, but also write code in multiple languages, ...

Female host: Yeah.

Male host: ...summarize dense academic papers in seconds, and even brainstorm potential downsides to your economic model. That's the promise of LLMs, right?

Female host: Yep.

Male host: But how do they actually work? Is it really some kind of artificial

intelligence magic?

Female host: It's not magic, but it's pretty close. Think of LLMs as incredibly complex neural networks, but instead of crunching numbers, they're trained on massive data sets of text. We're talking about everything from academic journals and books to code repositories and even social media posts—they process all of that information, learning patterns, relationships between concepts, and even different writing styles... [Remainder omitted for space reasons.]

[Transcription using otter.ai. Full audio version available at <https://t.ly/GrC0d>.]

The generated podcast lays out the material in the paper in an engaging conversational format while covering the content in a clear and insightful manner targeted at laypeople. One interesting observation is that the male host mistakenly attributes physical capabilities to AI research assistant (“...who can not only brew you a perfect cup of coffee...”).

Since October 2024, NotebookLM allows users to customize the generated podcasts with specific instructions. For researchers, a useful prompt may look like this: “Your audience are PhD economists who are eager to learn how to effectively use LLMs in their research work.”

Whereas the workspace tools described so far are based on the strategy of creating a new interface from scratch, like Artifacts or Canvas, in order to facilitate interactions with LLMs, the following tools follow the opposite strategy—they incorporate LLMs or similar systems to automatically perform functions in existing workspaces. Given the wide user base of the associated products, this integration will likely lead to widespread distribution of the benefits of these tools:

Copilot in Microsoft 365 integrates LLM capabilities into Microsoft Office products for an add-on subscription fee of \$20/month. Copilot is based on OpenAI’s latest GPT-4o model (and o1 in a pilot) as well as Microsoft-internal LLMs and can serve as an assistant and tutor for a wide range of office tasks. Use cases in Microsoft Word include creating drafts, including by brainstorming or referencing existing files; transforming text according to criteria like length, tone, formality, or intended reader; and summarizing and asking questions about content. In Microsoft Excel, use cases including extracting, converting, or reformatting data (use cases in data analysis); constructing, editing, and explaining formulas, even complicated ones; creating tables and charts to analyze and visualize data. Use cases in Microsoft PowerPoint include brainstorming, outlining and creating slides; enhancing content with images, design elements, and interactivity; summarizing and organizing presentations to highlight key points and action items; and even anticipating the most likely audience questions. Copilot is also available for all other programs that are part of the Microsoft 365 package, including Outlook, Teams, OneNote, as well as for Microsoft Edge.

Gemini for Google Workspace uses Google DeepMind’s Gemini series to offer a set of similar capabilities for Google’s office suite, including in Google Docs for writing documents, Google Spreadsheets, and Google Slides for presentations, as well as in the Gmail service.

Cursor – The AI Code Editor is a tool that is specialized in AI-assisted coding, making it particularly relevant for researchers engaged in computational work, data analysis, and econometric modeling. Cursor takes advantage of the long context windows and greater speeds of the latest LLMs to take the code suggestions pioneered by Github Copilot to new heights. It integrates AI assistance into all aspects of the coding process, including code suggestions, queries about code, edits to selected code according to instructions, recommendations for code optimization, and help for debugging—all while keeping the entire code base of a project in its context window to recognize interdependencies. It also offers support real-time collaboration, facilitating teamwork on large-scale projects. Cursor is built on (or, more specifically, forked from) the popular VS Code environment and can employ multiple different LLMs, including GPT-4o, o1, and Claude 3.5 Sonnet. Peng et al. (2023) report that GitHub Copilot delivered productivity gains of 126% for coding back in 2023. Based on user reports, the gains from Cursor may be even larger.

In a similar vein, the LaTeX editor Overleaf offers a tool called Writefull that is specialized in producing and editing LaTeX code, including tables and equations.

Additional Tools for Providing LLMs with Context One of the challenges in using the current generation of frontier LLMs is that they are excellent at processing content, but it is difficult or time-intensive for the user to supply the most relevant context for a work task. Anthropic and OpenAI have developed two slightly different solutions to this problem:

Anthropic introduced *Projects for Claude* in June 2024, which allow users to upload background documents that are relevant for multiple chat interactions and to organize and bring together related chats and artifacts in one place. For example, I have created a project on “Generative AI for Economic Research,” to which I added the earlier versions of this paper (Korinek, 2023a) and the project-specific custom instructions “Help me draft content for my research project on ‘Generative AI for Economic Research’ in a similar style to the earlier versions and in a format that is helpful for economic researchers.” When I use Claude to work on content related to this paper, I start a new chat that is part of this project, automatically providing Claude with all the relevant content. Projects help ground the LLM’s outputs in relevant context and background knowledge to effectively mitigate the “cold start” problem when opening a new chat. They can also be shared across teams, enabling more collaborative workflows.

OpenAI allows users to create *Custom GPTs* that enhance GPT-4o’s functionality by adding specialized background knowledge, interactive tools, and customized instructions tailored to specific instructions, ranging from writing to economic analysis tools.

Users can also create their own custom GPTs by clicking “Explore GPTs” at the top left and the “Create” button, and going through the ensuing process step by step. The resulting custom GPTs can be used privately or shared publicly in a GPT Store. Custom GPTs created by others can also be located in the “Explore GPTs” menu at the top left, which offers users the ability to browse, install, and use a wide range of extensions and applications that are developed by third-party creators. Two custom GPTs that economists may find useful are (1) Wolfram, which provides access to computation, math, curated knowledge and real-time data from Wolfram Alpha, the maker of Mathematica, and (2) Consensus, which offers an AI-based research assistant that searches 200m academic papers to provide science-based answers with citations to the underlying articles.

2.2.2 Real-time Voice Assistants [Fully Updated in Fall 2024 Version]

A significant innovation in access modes is a new generation of real-time voice assistants. Earlier voice assistants transcribed a user’s spoken language into text that was fed into an LLM; after processing the request, the LLM’s response was translated into audio again. By contrast, the new generation natively processes spoken text with all its nuances in tone and emotional expression and responds accordingly. Moreover, it also allows users to jump in and interrupt the flow mid-sentence in a way that allows for a more natural and fluid conversation. Some users report that they keep the ChatGPT app open on their phone in voice mode throughout certain work tasks, or even throughout the day, so that they can easily draw on the their digital assistant at any point without raising a finger.

The following are the leading interactive voice assistants of this new generation:

- OpenAI’s Advanced Voice Mode is a feature of its ChatGPT mobile app that offers perhaps the most natural interaction, using the GPT-4o model.⁸ In a version that is not yet publicly released, the model can also use the mobile phone’s camera to include a video view of the user or their surroundings in its interactions. A desktop app to use Advanced Voice Mode that can see and respond to the information on a user’s desktop is also in the works.
- Google’s Gemini Live also allows for fluid voice conversations with users based on the Gemini series of models. Google is working on integrating Gemini Live with apps across the Google eco-system, including Gmail, Calendar, Docs, YouTube, and Maps, to turn it into a powerful productivity assistant. Moreover, it is also working on a “Project Astra” (**A**dvanced **s**eeing and **t**alking **r**esponsive **a**gent) that will incorporate vision features in Gemini Live.

⁸At the time of writing, Advanced Voice Mode is not yet available to ChatGPT Plus or Free users in the European Union.

- Apple Intelligence voice assistant is part of the latest round of operating system updates across all Apple devices. It introduces an assistant that integrates seamlessly with Apple’s ecosystem, handling general queries, managing tasks, and interacting fluidly across apps like Mail, Calendar, and Notes. For more complex or nuanced inquiries, Apple Intelligence selectively leverages ChatGPT, adding depth to its responses when necessary. Although Apple’s proprietary AI system provides a smooth, integrated experience, some users report that it lacks the advanced capabilities of standalone ChatGPT, particularly in handling complex, multi-layered questions.
- For researchers interested in open-source solutions, the AI lab Standard Intelligence has publicly released Hertz-dev, accessible at <https://si.inc/hertz-dev/>. Their models provide a versatile, real-time voice assistant solution that facilitates natural spoken interactions with LLMs. Its efficient compression and ultra-low latency make it ideal for real-time applications, while its audio generation capabilities enable nuanced, responsive conversations. As an open-source platform, Hertz-dev offers high customizability, allowing researchers to tailor it to their specific needs, such as automated interviews or integration with other research tools for a seamless, interactive experience.

To combine voice interactions and traditional text-based interactions, both OpenAI’s and Google’s models provide users with transcripts of their voice interactions which can be copied and pasted for further processing and for written research products.

2.2.3 Autonomous Computer Use [All New in Fall 2024 Version]

The perhaps most breathtaking recent advance has been an autonomous desktop assistant, simply labeled “Computer use,” which was released by Anthropic in beta mode in October 2024. The system enables Anthropic’s most cutting-edge model, Claude 3.5 Sonnet, to directly interact with your computer’s interface, allowing it to see your computer screen and giving it access to virtually any software application that can be installed on a computer. This implies that the model can control your cursor, click buttons, type into text fields, and even navigate through software interfaces—as if another intelligent being were sitting at your computer. Although still preliminary, computer use gives LLMs the ability to automate a wide range of tasks that require multiple applications or complex workflows on a computer, ranging from organizing files and updating software to conducting online research. In effect, this development gives LLMs nearly unlimited access to external tools, enabling them to seamlessly interact across platforms and applications without manual intervention. Anthropic’s Claude with "Computer use" can currently be accessed through Anthropic’s API, which enables users to programmatically direct Claude to perform any desired operations on a computer. An instructive demo video is available at <https://www.youtube.com/watch?v=0DaHJz0yVCQ>.

Google’s Project JARVIS (acronym for “Just A Rather Very Intelligent System,” which is inspired by the AI assistant in the *Iron Man* franchise) is an experimental AI assistant that operates within the Google’s Chrome browser environment, where it can perform web-based tasks such as filling out forms, navigating websites, and making online purchases. Currently, Jarvis is in the experimental phase and expected to be available more broadly in December 2024. Its functions are limited to browser-based tasks, unlike Anthropic’s Claude, which can interact with any software installed on a computer. This makes Jarvis highly useful for automating tasks online but less versatile for complex workflows that require access to local applications.

For researchers, autonomous desktop assistants offer significant potential. They can automate standard research workflows, such as organizing datasets, managing references, and conducting data analyses in econometric packages. Additionally, for bulk operations like systematically collecting information from multiple sources, running batch simulations, or automating data entry, an LLM with desktop control can handle repetitive tasks with ease and efficiency. However, these capabilities also introduce risks, including grave security risks and privacy concerns, as such systems obtain full control over the user’s device or browser.⁹ Researchers must weigh these risks carefully, ensuring that sensitive data and systems are protected when taking advantage of the automation benefits these tools can offer.

2.2.4 LLM-based research tools [All New in Fall 2024 Version]

There are also a growing number of dedicated research tools that are based on LLMs and facilitate or automate research tasks. I will highlight two:

Expected Parrot Horton et al. (2024) develop an open-source python package to facilitate research on LLM-based simulations and surveys. In a dig at the term “stochastic parrot” that was used to critique LLMs, they have developed *Expected Parrot Domain-Specific Language* (EDSL), which takes advantage of LLMs’ ability to generate a wide range of context-specific data that closely mirror human behavior and social dynamics. EDSL allows researchers to define a set of as *Questions* that are answered by *AI Agents* simulated by defined *Models* to produce a set of *Results*, which can be grouped into *Surveys* and contextualized with *Scenarios* (capitalization used to refer to specific objects in EDSL).

This approach enables economists to efficiently manage large-scale tasks with intricate dependencies, agent behaviors, and model parameters without getting bogged down in

⁹For example, Anthropic notes that Claude with computer use sometimes erroneously follows instructions that it happens to read on open webpages or in images, thereby overriding the instructions that it has been given by its user. For this reason, they recommend that computer use is run on a dedicated virtual machine or container with minimal access privileges to prevent system attacks or accidents.

programming details. As a result, EDSL offers a powerful toolkit to conduct LLM-based simulations of detailed surveys and experiments, label large datasets, augment existing data, and generate synthetic data. Researchers can design AI agents with specific traits, utilize multiple language models simultaneously, and incorporate complex logic and agent memory into their surveys. EDSL’s built-in analysis and visualization tools, integrated into the python ecosystem, allow for both seamless execution and interpretation of research outcomes.

Sakana.ai AI Scientist Lu et al. (2024) at the Japanese startup sakana.ai introduce an automated framework for end-to-end scientific paper generation in computer science based on LLMs. The AI Scientist, as they call it, is designed to autonomously generate research ideas, implement experiments by running code, analyze results, and produce complete academic papers. While currently limited to a specific area within computer science in which progress can be made simply by writing code (machine learning algorithms and architectures), this approach demonstrates the potential for LLMs to assist across the research process.

The AI Scientist operates by generating novel research ideas, writing code to implement experiments, executing those experiments, and then drafting a full scientific paper based on the results. The system incorporates an automated reviewing process to evaluate the generated papers, mimicking the peer review system in academic publishing. Sample papers are available at <https://sakana.ai/ai-scientist/>.

Although the current quality of the generated papers is mediocre, lacking the full originality, depth and rigor of research authored by human experts, the framework points towards the potential future capabilities of LLMs in scientific research. It serves as a proof of concept for how LLMs could be leveraged to augment and accelerate the scientific process in the future. As LLM capabilities continue to advance, especially as they make breakthroughs in reasoning (see Section 2.1), systems like the AI Scientist may evolve into powerful tools for idea generation and the execution of research even in fields like economics.

2.2.5 Traditional text-based access modes [Updated in Fall 2024 Version]

The following summarizes the more traditional access modes for LLMs that have been available for the past two years:

- **Web-based Chatbots:** The models in Table 1 are all accessible as chatbots under the URLs listed in the last column. The chatbot interface, pioneered by Anthropic but first publicly released by OpenAI in the form of ChatGPT in Nov. 2022, allows users to prompt LLMs as assistants or tutors. Most of the examples documented in the use cases below illustrate this mode of interaction, which has been the most popular way of accessing LLMs over the past two years. However, I anticipate that LLM use will gradually shift towards the interactive workspaces

described in Section 2.2.1.

The free versions of the listed chatbots typically comes with usage restrictions or provides access to less powerful model versions. In my experience, this makes it worthwhile to pay the \$20 monthly subscription fee that is typically required for full access to the frontier models listed in the table.

All of these chatbots are also available via apps on Apple and Android mobile phones. Moreover, OpenAI and Anthropic have also developed desktop apps for their chatbots that are available for download at <https://openai.com/chatgpt/download/> and <https://claude.ai/download> respectively and, once installed, can be conveniently accessed via the keyboard shortcuts Ctrl+Space and Ctrl+Alt+Space.

- **Web-based Experimentation Platforms:** All major LLM providers also offer web-based interfaces that offer greater functionality and flexibility than chatbots but do not require programming knowledge. These platforms, such as OpenAI Playground (<https://playground.openai.com>), Google AI Studio (<https://aistudio.google.com>), and Anthropic Console (<https://console.anthropic.com>), allow users to experiment with different model settings, like temperature and top-p sampling, and provide more control over the input and output formats compared to chatbots. Such experimentation platforms are particularly useful for exploring the capabilities of LLMs, testing prompts, and fine-tuning models for specific tasks.
- **Application Programming Interfaces (APIs):** For the maximum level of customization and integration, the listed models are also accessible through APIs, which allow programmers to integrate LLMs directly into their own software applications. This enables a wide range of more advanced and customized use cases, such as automating repetitive tasks or analyzing large datasets using natural language processing techniques. APIs provide more flexibility and control compared to above two options and can be employed on a pay-per-use basis, but they also require a higher level of technical expertise to use effectively. Accessing LLMs through APIs typically involves signing up for an API key from the model provider (which can be thought of as a credit card for LLM tokens), installing a client library in the programming language of choice, and writing code to interact with the API endpoints. While this process may be more complex than using a chatbot, it unlocks the full potential of LLMs for those with the necessary programming skills. The replication package for this paper demonstrates how to use APIs to automatically query LLMs.
- **Locally Operating LLMs:** Open-source models allow researchers to run LLMs on their own computers, offering advantages such as data privacy, cost-effectiveness, customization, and offline accessibility. The computational resource requirements imply that only small models can be executed at a reasonable speed on desktop

computers. However, advances in computational capacity as well as rapid efficiency gains of LLMs that allow greater capabilities of smaller models are rapidly making the local use of LLMs more attractive. Two solutions that make it particularly easy to deploy LLMs locally are:

- LM Studio allows users to download and run a range of open-source LLM, including VLMs, on their personal computer or server.
- llamafile makes it possible to download LLMs in a single file and run it on a wide range of computer systems.

Centralized Hubs for LLM Interaction and Experimentation A useful website with a user-friendly chat interface that offers access to all leading LLMs is <https://poe.com>. Similarly, a website that offers users a web-based experimentation platform with access to a wide range of different models is <https://nat.dev/>.

Prompting Guides for Text-based Access Although it is useful for everyone to gain experience interacting with LLMs through their own experimentation and learning-by-doing, most leading model providers now also offer public documents that describes strategies for how to prompt LLMs. The following is a list of such guides:

- <https://platform.openai.com/docs/guides/prompt-engineering>
- <https://ai.google.dev/gemini-api/docs/prompting-intro> and an example of a research assistant at <https://aistudio.google.com/app/prompts/research-assistant>
- <https://docs.anthropic.com/claude/docs/intro-to-prompting> and an associated prompt library at <https://docs.anthropic.com/en/prompt-library/library>
- <https://llama.meta.com/get-started/#prompting>

Anthropic also offers a “prompt generator” that helps users to optimize the prompts that they employ for querying LLMs through the Anthropic Console Dashboard (click “Generate a Prompt”). This tool converts basic instructions in written language describing a user’s intent into a more complex and specific set of instructions that induce LLMs to generate higher-quality output. Naturally, the user can edit and adjust the generated instructions before use to ensure that they capture their intent as well as possible. I illustrate this using the following example. Throughout the remainder of this section, I will present all examples generated by LLMs in boxes, with the prompt in the header and the LLM’s generated response in the body of the box.

Chat 4 (Anthropic Prompt Generator): I have written an economics paper and would like to create a draft of a blog for Brookings.

Your task is to help draft a blog post for the Brookings Institution based on an economics research paper. Here is the full text of the paper:

```
<paper>
{{PAPER}}
</paper>
```

The target audience for this blog post is:

```
<audience>
{{AUDIENCE}}
</audience>
```

First, carefully read through the entire paper, taking notes on the key points, main findings, and implications in a <scratchpad> section.

Next, write a short outline for the blog post in a <outline> section. Keep the target audience in mind as you decide what to focus on and how to present the information.

Then, using your outline as a guide, write out the full draft of the blog post in a <blog_post_draft> section. Remember to:

- Use clear, engaging language that will be easily understood by the target audience
- Incorporate relevant examples, anecdotes or analogies where appropriate to make the post more relatable and interesting
- Avoid copying word-for-word from the original paper. Put the key concepts into your own words.
- Focus on the implications and "so what" of the research rather than getting too in-the-weeds with technical details

After you finish the draft, review it to make sure it covers the most important and interesting points from the paper in a way that will resonate with the target Brookings audience. Make any final tweaks before submitting your draft.

The generated prompt provides a remarkable level of detail that employs several cutting-edge prompting techniques to obtain the best possible results: it asks the model to start by taking notes in a “scratchpad” section, which induces the model to synthesize the main lessons of the text. Then it asks the model to prepare an outline, which induces the model to generate a plan. Only after these steps does the prompt ask the model to write a draft of the text. The prompt also leaves two locations for the users to insert the paper in question and to define the audience. I simply substituted these by writing “The full text of the paper is attached” and inserting “policymakers” as the target audience. The prompt generated useful results on all three of the leading LLMs that would save considerable time in such a project, although the generated blog

post was not quite up to the depth and quality of a think tank like Brookings.¹⁰

Bsharat et al. (2024) systematically study 26 different guiding principles for prompting leading LLMs and offer prompting advice based on a systematic evaluation of their effectiveness. Curiously, they find that strategies such as (fictitiously) offering LLMs a tip or threatening a penalty improve performance.

2.3 Advances in LLM-Powered Search [All New in Fall 2024]

In recent months, Perplexity AI and OpenAI's ChatGPT Search have become increasingly capable at combining the strengths of traditional search engines with those of generative AI. Like search engines, these systems have access to real-time information via the web, which overcomes the shortcoming of earlier generations of LLMs that their world knowledge was confined to their increasingly outdated training data. However, like LLMs, they can process the results using the sophisticated synthesis capabilities of generative AI. While traditional search engines provide lists of links that users must manually browse and evaluate, LLM-powered search tools can perform this for the user and synthesize information from multiple sources into coherent, direct responses that can be formatted in any desired way, for example, in table format. They maintain all the powerful features of LLMs, including the ability to understand conversation context and support follow-up questions, but now enhanced with real-time knowledge such as the latest economic research, current market data, and emerging policy debates.

The following two chats provide examples, for which I uploaded an earlier version of this article:

¹⁰I asked Anthropic Prompt Generator to create prompts for blog posts for several different outlets. For some reason, asking for a blog targeted at Brookings gave rise to a higher-quality prompt than asking for any other outlet I attempted. My best explanation is that the model internalizes that Brookings is known for its high-quality analysis. (Disclaimer: The author is a non-resident fellow at Brookings.)

Chat 5 (ChatGPT-4o with Search): Can you draft a new subsection on LLM-powered search for my paper on "Generative AI for Economic Research," focusing on ChatGPT Search and Perplexity. Search the web to find out what people find most useful about these search tools. Then draft four concise paragraphs in an engaging academic style that fits well with the paper:

1. Introduce LLM-powered search as one of the recent innovations - what's the benefit of combining LLMs with search?
2. Overview and comparison of ChatGPT Search and Perplexity AI
3. Applications for academic research
4. Risks (but emphasize that search greatly reduces hallucinations)

[Attachment: LLMs_2024h2.pdf]

LLM-Powered Search: Enhancing Information Retrieval in Economic Research

The integration of Large Language Models (LLMs) with search functionalities represents a significant advancement in information retrieval, particularly beneficial for economic research. By combining the extensive language understanding capabilities of LLMs with real-time data access, researchers can obtain synthesized, contextually relevant information efficiently. This fusion facilitates the extraction of nuanced insights from vast datasets, streamlining the research process and enabling more informed analyses.

Two prominent LLM-powered search tools are ChatGPT Search and Perplexity AI. ChatGPT Search, developed by OpenAI, extends the conversational abilities of ChatGPT by incorporating real-time web search capabilities, providing users with up-to-date information and direct citations. Perplexity AI, on the other hand, functions as an AI-driven search engine that delivers concise answers with source attributions, emphasizing accuracy and transparency. While both tools aim to enhance information accessibility, ChatGPT Search offers a more interactive experience, whereas Perplexity AI focuses on precise, citation-backed responses. In academic research, particularly within economics, these tools offer substantial advantages. They enable scholars to swiftly access current data, literature, and analyses, thereby expediting the research process. The ability to retrieve information with source citations ensures the credibility of the data, which is crucial for rigorous academic work. Moreover, the conversational interfaces of these tools allow for iterative querying, facilitating deeper exploration of complex economic topics.

Despite their benefits, it is important to acknowledge potential risks associated with LLM-powered search tools. One concern is the possibility of generating inaccurate or misleading information, known as "hallucinations." However, the integration of search functionalities significantly mitigates this issue by grounding responses in real-time data and providing source citations. This enhancement not only improves the reliability of the information retrieved but also fosters

transparency, allowing researchers to verify and cross-reference data effectively.

Chat 6 (Perplexity AI Pro): [same user prompt as in Chat 5]

Here's a draft subsection on LLM-powered search for your paper:

LLM-Powered Search: Enhancing Information Retrieval

The integration of Large Language Models (LLMs) with traditional search engines represents a significant leap forward in information retrieval. By combining the vast knowledge base and natural language understanding capabilities of LLMs with real-time web search, these tools offer a more intuitive and comprehensive search experience. LLM-powered search can interpret complex queries, synthesize information from multiple sources, and present coherent, contextually relevant responses, effectively bridging the gap between raw data retrieval and insightful analysis.

Two prominent examples of LLM-powered search are ChatGPT Search and Perplexity AI. ChatGPT Search, integrated into OpenAI's ChatGPT Plus, leverages the GPT-4 model to provide up-to-date information and web browsing capabilities. It excels in offering detailed, conversational responses and can follow up on complex queries. Perplexity AI, on the other hand, is designed as a standalone search engine that combines web search with AI-generated summaries. It distinguishes itself by providing source citations for its responses, allowing users to verify information easily. While both tools offer similar core functionalities, Perplexity AI's emphasis on transparency and its dedicated search interface may appeal more to researchers seeking clear attribution of sources.

For academic research, these tools offer several advantages. They can rapidly summarize the state of knowledge on a given topic, potentially accelerating literature reviews. Researchers can use them to explore interdisciplinary connections, generate research questions, or find relevant datasets and publications. The ability to ask follow-up questions and receive clarifications in natural language can be particularly useful when delving into unfamiliar fields or complex topics. Additionally, these tools can assist in identifying research gaps by synthesizing information across multiple studies and highlighting areas where consensus is lacking or evidence is sparse.

While LLM-powered search tools significantly reduce the risk of hallucinations compared to standalone LLMs by grounding responses in current web data, some risks remain. The potential for bias in search results and AI interpretations persists, and the recency and reliability of sources can vary. Researchers must maintain critical thinking and verify important information through traditional academic channels. Despite these considerations, the integration of LLMs with search functionality represents a powerful tool for enhancing research efficiency and exploration, offering a valuable complement to traditional research methodologies.

Both of these summaries are actually quite good and free of hallucinations although perhaps a bit sterile for my taste. However, they illustrate that LLM-powered search represents a useful step in the direction of future AI models that will be capable of autonomously writing future updates of this article. However, for now, let me also point out that the prompt that I employed provided the LLM with a lot of structure that was based on my understanding of how LLM-powered search fits into the overall landscape of generative AI for economic research.

Perplexity AI is accessible at <https://www.perplexity.ai> and offers a chat interface much like ChatGPT but is specialized in searching the web to compile responses to user prompts based on the sources it identifies. What is most useful is that it provides links supporting the statements that it makes. After submitting a query, the site first shows the keywords it searches for and then a preliminary list of the materials that it evaluates as preliminary readings. Based on these, it generates a response that starts with a clickable list of final sources, continues with a synthesis of the materials found, and ends with a conclusion that summarizes everything. The user can follow the sources provided to judge their reliability and verify the model's output. Perplexity has several different "Focus" options, including "Academic," which focuses its search on published academic papers, or "Math." It also offers a "Pro Search" option that dives deeper into topics and allows users to fine-tune the results but requires a subscription for frequent use. Another example of Perplexity AI in background research is provided in Chat 22 on page 58.

OpenAI released ChatGPT Search only recently, in October 2024, but the model represents the new state of the art in LLM-powered web search. It can be activated by explicitly instructing ChatGPT to search the web or by clicking a globe button at the bottom of the prompt window. It largely replicates the functionality of Perplexity AI but integrates this with the other capabilities of ChatGPT, making it better at multi-turn conversations and integration with applications that require data analysis. Google's Gemini chatbot also draws on an internet search for questions for which this is relevant.

2.4 Technical Improvements [All New in Fall 2024]

The following two technical advances have been particularly useful for programming when using LLMs via API:

Structured Outputs The ability of leading LLMs to reliably produce structured outputs has advanced enormously over the past year. Traditionally, LLMs used to produce unstructured, free-text responses that could not directly be fed into other computer applications. By contrast, structured outputs enable LLMs to take in raw, unstructured, qualitative information and convert it to machine-readable data that seamlessly integrates with other more traditional computer programs, such as econo-

metric packages. Structured outputs therefore bridge the gap between LLMs and traditional computer applications.

JSON (JavaScript Object Notation) is one of the most common formats for structured data. It organizes information into key-value pairs, which makes it both flexible and easy for computers to interpret. JSON can store everything from simple numerical values and text to complex hierarchical data, such as nested lists or objects. For example, a JSON output might look like this: `{"country": "United States", "population": 332915073}`. XML (Extensible Markup Language) serves similar purposes, with data represented in tagged structures, such as `<country><name>United States</name><population>332915073</population></country>`, but it is a bit more verbose. Both formats are compatible with many programming environments, making them widely applicable across research and data-driven applications. Here is an example:

Chat 7 (Claude 3.5 Sonnet API): List the the world's three largest economies as `[{"country", "population", "GDP_per_capita"}]` in JSON.

```
[
{
  "country": "United States",
  "population": 332915073,
  "GDP_per_capita": 69288
},
{
  "country": "China",
  "population": 1411750000,
  "GDP_per_capita": 12556
},
{
  "country": "Japan",
  "population": 125507472,
  "GDP_per_capita": 39285
}
]
```

To illustrate the benefits of this, assume this JSON was obtained from an LLM by a python program via API and is stored in the variable `country_data`. Then individual entries, say for the first country (the index starts at 0) can simply be accessed by referring to `country_data[0]['country']` or `country_data[0]['population']` in the python code. This type of structured output was difficult to obtain reliably from LLMs just two years ago but is now readily available so LLMs can be integrated into programmatic workflows.

Shorten et al. (2024) introduced the benchmark StructuredRAG to evaluate LLMs

on their ability to produce structured outputs consistently and accurately, assessing tasks such as generating structured responses in JSON, based on criteria like precision, adherence to format, and reliability. On their benchmark, Anthropic’s Claude 3.5 ranked first with an almost perfect score, followed by Google Gemini 1.5 Pro and OpenAI GPT-4o, which delivered less consistent results in August 2024. However, in a September 2024 update, OpenAI included new functionality to allow users to specify any JSON format for the LLM to adhere to, greatly improving GPT-4o’s capabilities to produce structured outputs.¹¹

For economists, structured outputs are useful in a variety of applications, from organizing country-level economic indicators as in the example above, to managing survey data, financial data, sentiment data, or a wide range of other data sources.

Prompt Caching is a technique to enhance LLM performance by storing and reusing previously processed text. This can reduce computational redundancy, resulting in higher speed and lower costs. For example, if a user repeatedly requests similar analyses based on a long introductory prompt, caching of the prompt instead of processing it again and again multiple times can save costs and speed up processing.

Both OpenAI and Anthropic have implemented prompt caching systems in their APIs. OpenAI’s implementation automatically activates when prompts exceed 1,024 tokens, caching the longest previously computed segment for reuse, which incurs only half of the usual cost of API use. Anthropic has developed a system that charges users 25% extra to write prompts into a cache but then allowing the reuse of cached information at a cost of only 10% of the regular price of text processing. For researchers, prompt caching is particularly appealing when performing text analysis in bulk as prompts can be cached and redundant computations be avoided, saving both time and money.

2.5 Practical Considerations for LLM Usage [Significantly Updated in Fall 2024]

Data Confidentiality An important issue for researchers is how to ensure the confidentiality of the data that they enter into LLMs. OpenAI offers a “Temporary Chat” option in its ChatGPT app as well as a privacy option in the user settings (turn off “Improve the model for everyone”) to let users opt out from their inputs being used for training future LLMs. OpenAI does not use user data that are entered via APIs. Anthropic does not use user data for future training except with an explicit opt-in or, in rare circumstances, if it is flagged for safety review. Google advises users against entering confidential information into its Gemini apps since input data may be used for future training purposes. For highly confidential data, the safest way of using LLMs is to run a cutting-edge open-source model on a local computer.

¹¹For further details, see <https://platform.openai.com/docs/guides/structured-outputs/>.

Declaring LLM Use In economics, most AEA journals will soon require authors to declare whether and how they have employed LLMs in their research. Although I usually welcome transparency, my own perspective is that such a requirement is unnecessary and may be potentially counterproductive. LLMs are rapidly becoming essential tools in the research process, akin to word processors, calculators, or econometric software. If used responsibly, they do not inherently compromise the integrity or originality of research any more than these other widely accepted tools. It is crucial to remember that authors remain solely accountable for the content they submit, regardless of the tools used in its creation. While it may be beneficial to remind authors of this responsibility when submitting, a formal declaration requirement could inadvertently create unwarranted skepticism among readers and discourage the use of these powerful productivity-enhancing tools. Moreover, such declarations are difficult to verify conclusively, rendering them at odds with the spirit of the revelation principle, which emphasizes designing mechanisms that naturally encourage truthful disclosure. My own perspective is to be cautious about introducing additional bureaucratic steps that may impede the research process without clear benefits. I advocate for the judicious and responsible use of LLMs in research, always coupled with careful verification of results—a practice no different from how we treat output from human research assistants or other analytical tools. The focus should remain on the quality and integrity of the final research product, rather than the specific tools used in its creation.

Watermarking [All New in Fall 2024] Relatedly, watermarking of LLM outputs has become an important new consideration when using LLMs. Watermarking embeds markers in AI-generated text by introducing a specific fingerprint key in the pseudo-random token selection during the text generation process (Dathathri et al., 2024). This makes it possible to trace back the origin of the text to the LLM for those who know the associated key while remaining undetectable to regular readers. Google has implemented this watermarking method, called SynthID, in the output of its Gemini models, representing the first known large-scale deployment of text watermarking.

While watermarking could help establish provenance and potentially address concerns about academic integrity and unauthorized AI use, its implications deserve careful consideration. First, the markers are not reliable since they can be defeated through simple paraphrasing using other LLMs. Secondly, the practice raises privacy concerns since watermarks could theoretically enable tracking of AI-generated content back to individual users. For economic researchers, watermarking has important implications both as a subject of study (for example, regarding information asymmetries and verification mechanisms) and as a practical consideration when using LLMs for research tasks. Users of Google DeepMind’s Gemini models should be aware of the watermarks contained in the generated output. It is unknown whether other labs employ similar mechanism.

Reproducibility is a challenge when working with LLMs, for several reasons. First, chatbots are programmed to be random—users typically rate responses more highly when the so-called “temperature” parameter of the LLM that controls the degree of randomness introduced into the text generation is greater than zero. Secondly, even at zero temperature, the output of LLMs is not always perfectly reproducible for internal technical reasons.¹² Third, to the extent that models draw on web search (as described in Section 2.3 on page 32), the continually evolving nature of information available on the internet changes the search results that feed into LLMs’ responses. Finally, the models offered by companies with proprietary models change over time and older, less efficient, models are regularly deprecated.¹³

The examples and use cases in the remainder of this article use the leading publicly available LLMs at the time that each use case was incorporated into this living document. The latest examples employ OpenAI’s o1 and GPT-4o as well as Claude 3.5 Sonnet (New). The examples that originate from the Dec. 2023 JEL version of the article used primarily OpenAI’s GPT-4, version `gpt4-0613`. In the online materials associated with this article (see footnote on the frontpage of the article), I provide python code to reproduce those results by calling OpenAI’s API. Non-programmers can replicate the results (subject to the limitations discussed above) on the web-based experimentation platform <https://platform.openai.com/playground>, setting the temperature parameter to zero. Both the OpenAI API and the Playground platform require a paid subscription for access.¹⁴

Additional Resources Let me also point to two additional resources for readers interested in the topic of this paper. First, Ash et al. (2024) provides a survey of how LLMs have transformed text analysis in economic research. Dell (2024) offers

¹²For example, OpenAI states that “setting temperature to 0 will make the outputs mostly deterministic, but a small amount of variability will remain.” See <https://platform.openai.com/docs/guides/gpt/why-are-model-outputs-inconsistent> for further information on the inconsistency of model output at temperature zero, and <https://community.openai.com/t/a-question-on-determinism/8185> for a discussion of the inherent indeterminacy of efficiently performing LLM inference.

In a nutshell, the efficient execution of LLMs with hundreds of billions of parameters requires that calculations are parallelized. However, given the discrete nature of computers, calculations such as $(a \cdot b) \cdot c$ sometimes deliver a slightly different result than $a \cdot (b \cdot c)$. When an LLM calculate which word has the top probability to be next, minor differences in the parallelization of the exact same calculations sometimes come to matter, resulting in different word choices. And once one word changes, everything that follows becomes different.

¹³See, for example, <https://platform.openai.com/docs/deprecations> on OpenAI’s policy of model deprecations as well as the current timelines for how long existing models are guaranteed to remain available.

¹⁴Executing all of the examples based on OpenAI models in Oct. 2023 cost slightly below 50 cents. Using the updated and more powerful GPT-4o for the same queries in June 2024, the cost had fallen to 10 cents. At the time of writing, the latest version of GPT-4o has halved costs yet again. Up-to-date pricing information for OpenAI’s models is available at <https://openai.com/pricing>.

a JEL survey of deep learning for economists, covering classifiers, regression models, generative AI, and embedding models, together with a companion website, EconDL.

I now turn to tangible uses of generative AI in economic research.

3 Use Cases of Generative AI [Updated in Fall 2024]

Table 4 summarizes all the sample tasks illustrated in this paper, categorized by the seven different domains of application of LLMs. In the Fall 2024 version of this paper, I created a new category, “Promotion,” in which there are several new impressive use cases, described in Section 3.7. Readers who are familiar with earlier versions of this paper may want to focus on that section to check out these new use cases.

In the third column of Table 4, I report my subjective rating of how useful I found the described LLM capabilities as of November 2024. My ratings come in three categories, where an empty circle (\circ) describes capabilities that I currently consider more experimental and that deliver inconsistent results, requiring significant human oversight; a half-full circle (\bullet) signifies capabilities that are useful and likely to save time but are somewhat inconsistent so that they still require careful oversight; and a full circle (\bullet) reflects capabilities that are already highly useful and work in the expected manner most of the time. Incorporating these latter capabilities into your workflow will definitely save you time and make you more productive. The online resources associated with this paper (see title footnote) include an up-to-date and more interactive version of these use cases.

The new inclusions in the table since the original publication in December 2023 are marked with superscripts for 2024/06 or 2024/11 in the second column. The superscripts in the third column mark capabilities for which I have increased my ratings since December 2023. These cover the following: In the category “Writing,” the ability of LLMs like GPT-4o to transcribe hand-written equations has improved significantly due to the greater vision capabilities of frontier LLMs. In the category “Data Analysis,” LLMs can now classify and score text and extract sentiment due to greater ability to understand context and reason. In “Coding,” LLMs have become highly useful in writing clean code, explaining code, and even debugging code, with OpenAI’s new reasoning model o1 having led to significant breakthroughs. The same model is also highly useful for deriving equations in the category “Math.”

3.1 Ideation and Feedback

Research starts with the process of ideation, i.e., generating, developing, and selecting ideas. I start my exploration of LLMs with use cases that involve ideation and feedback for two reasons. First, starting with ideas follows the natural sequence of research. Second, ideation and feedback showcase a new set of capabilities that starkly distinguish LLMs from earlier applications of deep learning in research – they display

Category	Task	Usefulness
Ideation & Feedback	Brainstorming	●
	Feedback	◐
	Providing counterarguments	◐
Writing	Synthesizing text	●
	Editing text	●
	Evaluating text	●
	Converting hand-written equations ^{24/6}	● ⁺
	Generating titles & headlines	●
Background Research	Summarization	●
	Condensing YouTube videos ^{24/6}	●
	Literature Research	◐ [*]
	LLM-Powered Search ^{24/6}	◐
	Formatting References	●
	Translating Text	●
	Explaining Concepts	◐
Coding	Writing code	● ⁺
	Explaining code	● ⁺
	Translating code	●
	Debugging code	● ⁺
Data Analysis	Locating data sources ^{24/6}	◐
	Creating figures	◐
	Extracting data from text	●
	Reformatting data	●
	Classifying and scoring text	● ⁺
	Extracting sentiment	● ⁺
	Simulating human subjects	◐
Math	Setting up models	◐
	Deriving equations	◐ ⁺
	Explaining models	◐
Research Promotion	Social media posts	●
	Presentation slides ^{24/11}	●
	Blog posts ^{24/11}	●
	Conducting interviews ^{24/11}	●
	Podcasts ^{24/11}	●

The third column reports my subjective rating of LLM capabilities as of November 2024:

◐: experimental; results are inconsistent and require significant human oversight

◐: useful; requires oversight but will likely save you time

●: highly useful; incorporating this into your workflow will save you time

Superscripts 24/6 or 24/11 in 2nd row represents year and month of new inclusions

Superscripts in last row denote upgraded ratings in 2024/06(*) and 2024/11(+)

Table 4: Summary of LLM capabilities and rating of usefulness

a form of creativity that had long been reserved for humans. Ideation and feedback are areas where it pays off to use the most advanced LLMs available. A model of idea generation by Girotra et al. (2010) observes that creative performance depends on (i) the quantity of ideas, (ii) the average quality of ideas and (iii) the variance which determines how many exceptional ideas are generated. Girotra et al. (2023) find that GPT-4 outperforms MBA students at a top US business school on all three dimensions in a contest to develop innovative new product ideas. As a result, they argue that the bottleneck in ideation is increasingly shifting from generating to evaluating ideas. Si et al. (2024) find that Claude 3.5 Sonnet is better at identifying interesting questions in NLP research, although its questions are ranked as slightly less feasible.

As we will see in the following, although the current capabilities of cutting-edge LLMs in the areas of ideation and feedback are impressive, they also have limitations. There are also broader potential pitfalls. Any researcher who uses LLMs for ideation and feedback must of course be careful about which points they use and which points they reject in a given use case – just as we do when we discuss ideas with colleagues. There may also be subtle downsides that materialize over time. The reliance on LLM-generated ideas may make individual researchers rely more on automation and practice less critical thinking of their own. Moreover, if more and more economists rely on the same one or two cutting-edge LLMs to generate ideas and obtain feedback, there is a risk that the ideas that economists work on will become more and more homogeneous and include fewer truly novel ideas. This risk of homogenization is also discussed in Bommasani et al. (2021). Moreover, when using LLMs for brainstorming or feedback, it is important to keep in mind when their training data cuts off (see Table 1).

Fall 2024 Status Update for “Ideation and Feedback” This category has significantly benefitted from the broad-based advances in the capabilities of LLMs and their ability to process ever larger context windows. However, no novel breakthroughs have occurred.

3.1.1 Brainstorming

Cutting-edge LLMs are quite useful for brainstorming (or, perhaps more aptly, *neural-net*-storming) ideas and examples related to a defined theme. Having been trained on a vast amount of data that represents a cross-section of all human knowledge, the breadth of the representation of the world that cutting-edge LLMs have developed from their training data includes a fair bit of knowledge of economics. However, at present, human experts still have an edge when it comes to depth, and so LLMs are best suited for brainstorming in areas in which one is not an expert.

The following prompt illustrates a simple example using GPT-4. Notice that I added an instruction to limit the response to 10 words for each point – otherwise the LLM produced a whole paragraph on each point, which may be useful in general but would be too lengthy for our illustrative purposes here:

Chat 8 (GPT-4): Can you brainstorm 20 channels through which AI may increase inequality? Limit your response to 10 words for each point.

1. Job loss due to automation in lower-skilled industries.
2. AI-driven wealth concentration in tech-savvy organizations.
3. Digital literacy gap leading to economic disparity.
4. Lack of universal access to AI technology.
5. AI-driven bias in hiring and selection processes.
6. Imbalance in job market due to AI specializations.
7. Data privacy issues affecting vulnerable populations.
8. AI-driven services predominantly targeting wealthy consumers.
9. Algorithms exacerbating social inequality through bias.
10. Inclusive AI product development lacking.
11. Higher prices due to AI-enhanced products/services.
12. AI-fueled gentrification in tech-centered cities.
13. Anticompetitive practices bolstered by AI applications.
14. Lack of labor rights for jobs displaced by AI.
15. Educational imbalance due to AI-learning tools.
16. AI in healthcare excluding lower socioeconomic groups.
17. Disproportionate influence of AI in political campaigns.
18. Undervaluing of human skills in favor of AI.
19. Biased AI systems perpetuating discrimination.
20. AI reinforcing societal hierarchies via data analysis.

The channels listed by the model are on-topic, largely reasonable, and span a wide range of what I could think of plus some more. However, some of the bullet points are a bit repetitive.

A noteworthy aspect to underscore is the remarkable speed and volume of responses generated by LLMs during activities like brainstorming, which generates its own distinct form of usefulness. Even if only a single suggestion out of 20 in examples like this proves beneficial, it may make our research significantly more productive.

Other brainstorming prompts that I found useful include the following:

- I am an economist working on AI and inequality. Can you brainstorm an outline on [insert topic]?
- I am an economist working on AI and inequality. Can you brainstorm 5 potential paper topics and describe each in one sentence?
- I am an economist working on an academic paper on [insert topic]. Can you brainstorm a research plan for me?

3.1.2 Feedback

LLMs can also evaluate ideas, highlighting, for example, the pros and cons of different hypotheses or research plans. The following example asks the LLM to list the pros and cons of working on a specific area of research. This shows that LLMs can provide useful input on different research directions.

Chat 9 (GPT-4): What are the top-3 pros and cons of working on the economics of AGI?

Pros of Working on the Economics of AGI:

1. **High Impact:** The development of AGI (Artificial General Intelligence) has the potential to revolutionize many sectors of the economy, from healthcare to finance to transportation. Working on the economics of AGI allows you to be at the forefront of this transformation and potentially shape its direction.
2. **Interdisciplinary Approach:** The economics of AGI involves a combination of economics, computer science, and philosophy, among other disciplines. This interdisciplinary approach can be intellectually stimulating and rewarding.
3. **Potential for Innovation:** Given that AGI is a relatively new and rapidly evolving field, there is a lot of room for innovative thinking and novel approaches. This can be exciting and fulfilling for those who enjoy problem-solving and creative thinking.

Cons of Working on the Economics of AGI:

1. **Uncertainty:** The future of AGI is highly uncertain, making it difficult to make accurate predictions or develop robust economic models. This uncertainty can make the work challenging and sometimes frustrating.
2. **Ethical and Societal Implications:** The development and deployment of AGI raise a host of ethical and societal issues, from job displacement to privacy concerns to potential misuse. Working on the economics of AGI means grappling with these complex and often contentious issues.
3. **Limited Data:** Given that AGI is still in its early stages, there is limited data available for economic analysis. This lack of data can make it difficult to test hypotheses or draw definitive conclusions.

Another example of a useful prompt for eliciting feedback is:

- I am an economist working on an academic paper on [insert topic]. What are the main challenges in researching this topic? How can I best address them?

Iteration What is particularly useful is to iterate between brainstorming and evaluation. Similar to how a researcher comes up with ideas, selects the most promising ones, and refines them, LLMs can be prompted to brainstorm, select which ideas it rates as the most promising, and brainstorm further on them.

Feedback on entire paper drafts The long context window of Claude 2 makes it possible to upload entire research papers into the LLM and ask for feedback. I fed the Feb. 2023 NBER working paper version of this paper (Korinek, 2023b) into Claude 2 and asked it the following:

Chat 10 (Claude 2): Can you suggest ways to improve this paper?
[Attachment: w30957.pdf]

Here are some suggestions for improving the paper on large language models and cognitive automation:

- Provide more specific details and examples when describing the different use cases. For some of them, the explanations are a bit vague. Adding more concrete illustrations would strengthen those sections.
- Expand the discussion of the limitations and potential downsides of over-relying on LLMs. The paper focuses mostly on the benefits, but more balance discussing the risks would be helpful.
- Strengthen the analysis of the long-term implications. The outlook section is thought-provoking but remains speculative. Adding more rigorous analysis to back up the predictions could make it more convincing.
- Consider testing the use cases on multiple LLMs and comparing their capabilities. The examples all seem to use a single LLM, but testing different ones could reveal variation in strengths/weaknesses.
- Provide guidance on how to properly evaluate LLM outputs before using them. Since LLMs can make mistakes, some tips on double-checking the results would be useful.
- (additional suggestions omitted for space reasons)

In general, providing more specifics, analysis, examples, and structure could take the paper from a promising start to a more polished and convincing final product. Let me know if you would like me to elaborate on any of these suggestions!

All of the listed bullet points are good suggestions, and I have incorporated several of them in the paper during the revision process.

Since Claude 2 can hold the content of the entire paper in its memory, it can offer comments on any parts of it if requested. The following are additional examples of useful prompts:

- What are the main strengths and weaknesses of this paper?
- What are the main novel ideas in the paper that are not sufficiently emphasized?

- Can you identify any instances of bias in this paper?
- How could I improve section [insert number]?
- Can you draft a referee report for this paper for the Journal of Economic Literature?

The capability unlocked in the last example is likely to revolutionize editing and refereeing, for better or worse. To provide an example, I asked Claude 2 to draft a referee report of the same working paper (Korinek, 2023b):

Chat 11 (Claude 2): Can you draft a referee report for this paper for the Journal of Economic Literature?
[Attachment: w30957.pdf]

Here is a draft referee report for the paper "Large Language Models and Cognitive Automation for Economic Research":

To the editors of the Journal of Economic Literature:

I have completed my review of the paper "Large Language Models and Cognitive Automation for Economic Research" by Anton Korinek.

This paper provides a timely and useful overview of how large language models (LLMs) like ChatGPT can be leveraged to automate and enhance various research tasks for economists. The paper is well-written and clearly presented. I recommend it for publication after minor revisions.

Summary:

The paper categorizes potential applications of LLMs into six domains - ideation, writing, background research, coding, data analysis, and mathematical derivations. For each domain, the author provides general guidance on prompt engineering as well as specific examples to showcase the capabilities and limitations of LLMs. Based on experiments using the GPT-3 model, the author assigns a usefulness rating to each application ranging from experimental (inconsistent results requiring oversight) to highly useful (will likely save researcher time).

The paper concludes by predicting how LLMs may affect economic research productivity in the short and long run. In the short run, the author argues LLMs will act as useful assistants to automate micro-tasks, increasing researcher productivity. In the long run, the author speculates LLMs may produce superior economic models, potentially displacing human economists. The paper ends by proposing new research questions economists should explore regarding the impacts of cognitive automation.

Comments:

The paper succeeds in its aim of providing an accessible introduction and framework for economists to begin leveraging LLMs in their work. The catalog of potential applications is comprehensive and the examples are illuminating. Prompt engineering is crucially important to effectively using LLMs, and the author helpfully includes prompts for replicability.

I have only minor recommendations to further strengthen the paper:

- Given the speed of developments in this area, mentioning the timing of experiments and capabilities of the specific LLM used would help frame the generalizability of the results.
- The usefulness ratings are subjective. Surveying researchers who have used LLMs may produce more robust assessments.
- Additional guidance on the factors that contribute to inconsistent vs. highly useful capabilities could better set expectations.
- Exploring collaborative applications where LLMs complement humans would enrich the long-run outlook.

Overall, I believe this paper makes a valuable contribution in a rapidly evolving area. It will serve as a helpful guide to researchers looking to exploit LLMs to improve their productivity. I recommend publication after minor revisions.

Please let me know if I can provide any clarification or additional feedback.

Sincerely,

[Your name here]

The LLM offered several comments that resemble comments made by the actual (presumably human) referees of the paper and wrote that a revised version of this paper would be a good fit for the *Journal of Economic Literature*. Interestingly, however, the comments in the draft referee report are shallower than the comments in previous chat, which directly prompted Claude 2 for ways to improve the paper.

It is well known that Claude 2 is programmed to be friendly and upbeat. To check whether the positive assessment in the previous chat simply reflected a positivity bias, I also asked Claude 2 whether the paper would be suitable for the *American Economic Review*:

Chat 11 (Claude 2) ctd.: Would this paper be suitable for the American Economic Review?

Based on my review, I do not think this paper would be suitable for publication in the American Economic Review in its current form. Here is my assessment:

The American Economic Review is one of the most prestigious and selective economics journals, publishing theoretical and empirical research that makes significant advancements in economic analysis and knowledge.

While this paper provides a useful and timely overview of how economists can apply large language models in their research, it is more expository and demonstrative rather than presenting novel research or analysis. As such, it may not rise to the level of contributing major new economic insights demanded by AER.

Some factors to consider regarding AER suitability:

- The paper catalogs potential LLM applications but does not rigorously evaluate their impact on economic research or productivity. An empirical analysis demonstrating and quantifying the benefits would strengthen it.
- There is limited theoretical or conceptual economic analysis. Expanding on implications for economic theory and modeling could improve fit for AER.
- The mathematical derivations section is quite limited in coverage. More technical treatment would be expected for AER.
- Long-run predictions about LLMs surpassing human abilities are interesting but speculative. Linking these to formal economic growth models would make them more rigorous.

With additional empirical analysis, technical depth, and formal economic modeling, this line of research could potentially yield papers suitable for AER. However, in its current form as an applications overview, the paper may not satisfy AER's standards for novelty and rigor. Journals focused on applications like the Journal of Economic Perspectives may be better fits.

I hope these comments are useful. Please let me know if you would like me to expand on my assessment of the paper's suitability for AER. I'm happy to provide additional feedback.

Claude 2 argued that the paper would not be suitable since "it is more expository and demonstrative rather than presenting novel research or analysis." It continued: "As such, it may not rise to the level of contributing major new economic insights demanded by AER." This (correct) assessment suggests that the model already possesses the rudimentary capacity to judge what makes for suitable research papers in different journals.

Whereas Claude 2 is able to provide reasonable feedback on a qualitative paper like this one, the current generation of LLMs struggle with evaluating more analytic or quantitative papers in an insightful manner.

Feedback via Voice Chats [All New in 2024Q2 Version] One of my colleagues reports that it is highly useful to voice chat with leading LLM chatbots such as ChatGPT or Claude to talk through research problems, obtain feedback, or become unstuck from a problem. The ability of LLMs to synthesize the conventional wisdom captured in their training data, to be a tutor, or to offer a broader framing to questions makes them well-suited for this purpose. Based on my own experiments, I concur that interacting with LLMs via voice chat makes for a different experience than interacting via text – an experience that can be useful just like it is sometimes useful to talk through a problem with a colleague.

3.1.3 Providing counterarguments

No matter what point we are arguing, there are always counterarguments. LLMs do not care about which side of an argument they are on – they are just as good at providing arguments in favor of a given point as they are for the counterarguments. They are also unconcerned about hurting our ego when we ask them for a critique. This may be helpful to counteract the confirmation bias common to our human brains. The following is an example (for space reasons, asking for short responses):

Chat 12 (GPT-4): I am working on an economic model that predicts that cognitive automation will increase inequality. What are the top 5 counterarguments? One line each.

1. Cognitive automation could lead to job creation in new sectors, offsetting job losses in others.
2. It may increase productivity and economic growth, leading to overall wealth increase.
3. Automation could potentially lower the cost of goods and services, benefiting all consumers.
4. It could provide opportunities for upskilling and reskilling, leading to wage growth.
5. Cognitive automation might reduce human error, increasing efficiency and potentially leading to more equitable outcomes.

The counter-arguments make sense, although I would point out that – in contrast to most laypeople – economists would view points 2. and 3. as the same since productivity is a real concept that adjusts for cheaper goods and services.

3.2 Writing

The core competency of LLMs is to generate text. This implies that they are quite capable and useful for many tasks related to writing, from synthesizing sentences based

on bullet points, changing the style of text, editing text, evaluating styles to generating titles, headlines, and tweets. I cover each in turn in the following.

Basic writing tasks do not generally require the most cutting-edge LLM available. For example, to synthesize basic text, any of the models listed in Table 1 will do a good job. However, when the LLM is asked to go beyond the immediate instructions, differences start to emerge based on the capabilities of each model. At the time of writing, tools for synthesizing text are also starting to be included in office packages such as MS Office and Google Workspace as well as the associated email programs Outlook and Gmail. Alternatively, a dedicated writing assistant that offers plugins for many common applications (including web browsers and office packages) and that suggests corrections and stylistic improvements as the user is typing is [grammarly.com](https://www.grammarly.com).

Fall 2024 Status Update for “Writing” This category of use cases has benefitted enormously from the new workspaces for interactive LLM collaboration, described in Section 2.2.1. The use cases below capture the essence of the capabilities of LLMs but are significantly easier to use in the described collaborative workspaces.

3.2.1 Synthesizing text

One of the most useful capabilities of LLMs for writing assistance may be to translate rough bullet points into well-structured sentences that are clear and easy to read. The following is an example. Note that the bullets also include requests for the LLM to provide examples and offer solutions.

Chat 13 (GPT-4): Write a paragraph in the style of an academic economist that integrates the following points. Include a topic sentence.

- * large language models (LLMs) automate many cognitive tasks
- * the work of academic economists is largely cognitive [add examples]
- * we economists should therefore worry about recent progress in language models
- * in the short run, these models can help us be more productive
- * in the long run, there is a good likelihood that they will displace human economists
- * we should think how to prepare [offer solutions]

The advent of large language models (LLMs) that automate a plethora of cognitive tasks presents a significant paradigm shift that academic economists should be cognizant of, given the cognitive nature of our work, which includes tasks such as data analysis, economic forecasting, and policy recommendation. The rapid advancements in language models could potentially disrupt our profession, necessitating a proactive approach to this emerging technology. In the short term, these models can augment our productivity by automating routine tasks, thereby allowing us to focus on more complex and nuanced aspects of economic research.

However, in the long run, there is a substantial likelihood that these models could displace human economists, as they become increasingly adept at performing complex cognitive tasks. Consequently, it is incumbent upon us to prepare for this eventuality. This could involve developing a deeper understanding of these models to leverage their capabilities effectively, diversifying our skill sets to include areas not easily automated, and advocating for policies that ensure a just transition for those potentially displaced by these technological advancements.

Utilizing this allows researchers to concentrate their energy on the ideas in their text as opposed to the mundane writing process. The text is generally high-quality, although some claim that it is a bit sterile and may lack the idiosyncrasies and elements of surprise that characterize human writing.¹⁵

LLMs are very responsive to requests to write in different styles. I obtained both impressive and hilarious results when I replaced “in the style of an academic economist” with the request to write:

- in a colloquial style
- in a style that is comprehensible to non-economists
- in the style of [enter your favorite economist, politician, etc.]¹⁶

LLMs can also write text in LaTeX format, which automatically format mathematical expressions in the required format, and it can draft beamer slides in LaTeX code based on bullet points.

Writing is of course a skill that is not only useful for research - cognitive workers in many areas report using, e.g., ChatGPT to draft emails, memos, performance reviews, or summaries based on a few bullet points. Likewise, professors report generating the first draft of letters of recommendation or syllabi using LLMs.

3.2.2 Editing text (for mistakes, style, clarity, simplicity, ...)

Another useful capability is editing. LLMs can edit text for grammatical or spelling mistakes, style, clarity, or simplicity. This set of capabilities is useful for all writers, but perhaps most for non-native speakers who want to improve their writing. In the following example, I have bolded each correction made by the LLM for emphasis:

¹⁵This is a feature that detectors of LLM-written text zero in on, for example the tool GPTZero, which is available at <https://gptzero.me/>. However, it should be noted that such tools are not always reliable and can easily be circumvented.

¹⁶When I requested “in the style of a gangster,” the LLM started with “Listen here, see, these big shot large language models (LLMs) are muscling in on our turf...”

Chat 14 (GPT-4): Can you correct the following sentence?

These paper describe six domain in which large language models (LLMs) displays emergant capabilities that are useful to the economic research.

This paper describes six domains in which large language models (LLMs) display emergent capabilities that are useful for economic research.

Chat 14 (GPT-4) ctd.: Can you explain the main corrections you made?

Sure, here are the main corrections:

1. "These paper" was corrected to "This paper". The subject-verb agreement was incorrect. If the subject is singular (like "paper"), the verb should also be singular (like "describes").
2. "describe" was corrected to "describes". The verb should match the singular subject "paper".
3. "domain" was corrected to "domains". The context implies that there are multiple domains, so the plural form should be used.
4. "displays" was corrected to "display". The subject "LLMs" is plural, so the verb should also be plural.
5. "emergant" was corrected to "emergent". This was a spelling mistake.
6. "to the economic research" was corrected to "for economic research". The preposition "for" is more appropriate in this context.

As illustrated in the follow-up question, the LLM can also largely explain its edits so that writers can learn from any corrections it has made. This latter capability has improved significantly going from GPT-3.5 to GPT-4. Notice, however, that it left out a seventh correction from the list – it also corrected a typo in the word “capabilities.” This highlights the need to carefully check the results of LLMs.

Another useful LLM application is to convert text to be comprehensible to readers of different levels, for example, using a prompt like “Rewrite the following text so that it is comprehensible to an eight-year-old”, “comprehensible to an eighth-grader,” or “comprehensible to a policymaker without economics education.”

3.2.3 Evaluating text

LLMs can also evaluate text for its style, its clarity, or similar. The following question about a draft version of the abstract of this paper turned out to be a bit of a deflating experience:

Chat 15 (GPT-4): Give a concise list of the top 5 stylistic shortcomings of the following abstract:

Generative AI, in particular large language models (LLMs) such as ChatGPT, has the potential to revolutionize research. I describe dozens of use cases along six domains in which LLMs are starting to become useful as both research assistants and tutors: ideation and feedback, writing, background research, data analysis, coding, and mathematical derivations. I provide general instructions and demonstrate specific examples of how to take advantage of each of these, classifying the LLM capabilities from experimental to highly useful. I argue that economists can reap significant productivity gains by taking advantage of generative AI to automate micro tasks. Moreover, these gains will grow as the performance of AI systems across all of these domains will continue to improve. I also speculate on the longer-term implications of AI-powered cognitive automation for economic research. The online resources associated with this paper offer instructions for how to get started and will provide regular updates on the latest capabilities of generative AI that are useful for economists.

1. Lack of Clarity: The abstract uses jargon and technical terms without providing clear definitions or explanations, making it difficult for readers unfamiliar with the topic to understand.
2. Overuse of Passive Voice: The abstract frequently uses passive voice, which can make the writing seem less direct and engaging.
3. Lack of Specificity: The abstract mentions "dozens of use cases" and "six domains" but does not provide any specific examples or details.
4. Redundancy: The abstract repeats the idea of AI's potential benefits for economists multiple times without adding new information.
5. Lack of Structure: The abstract does not have a clear structure or flow, making it difficult for readers to follow the author's argument.

Chat 15 (GPT-4) ctd.: Can you rewrite it to correct these shortcomings?

Generative AI, specifically large language models (LLMs) like ChatGPT, holds the potential to transform research. This paper explores the application of LLMs in six key areas: ideation and feedback, writing, background research, data analysis, coding, and mathematical derivations. For each area, I provide practical instructions and real-world examples, categorizing the usefulness of LLMs from experimental to highly beneficial.

The focus of this paper is on the potential benefits for economists. By leveraging generative AI, economists can automate micro tasks, leading to significant productivity gains. As AI systems continue to evolve, these benefits are expected to increase.

The paper also delves into the long-term implications of AI-powered cognitive automation for economic research. Accompanying online resources provide a step-by-step guide on how to get started with generative AI, along with regular updates on its latest capabilities beneficial for economists.

The shortcomings identified by the LLM are well-taken, and the rewritten version does a decent job of alleviating some but not all of them.

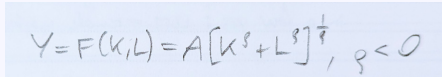
Other useful LLM prompts for style evaluation include:

- Which arguments are the hardest to follow in the following text?
- Evaluate the clarity and coherence of the following text.
- Are there any ambiguities in the following text?

3.2.4 Converting equations to LaTeX [All New in 2024Q2 Version]

The vision capabilities of most advanced LLMs allow them to interpret images and follow instructions on how to process them. This can be useful, for example, when typing up mathematical equations. The following example illustrates the concept.

Chat 16 (GPT-4): Convert the attached image to LaTeX code.



The LaTeX code for the equation you provided would look like this:

$$Y = F(K, L) = AK^\alpha L^\beta, \quad \alpha, \beta \leq 0$$

This LaTeX code will render the production function Y as a function of capital K and labor L , with A representing total factor productivity, and α and β as the output elasticities of capital and labor, respectively, which are both non-positive in this case.

The result is clearly still lacking: it omits the square brackets, turns the Cobb-Douglas function into a CES function, replacing the exponent ρ with α and β , and omits the exponent outside the square brackets. Gemini performed about equally well as ChatGPT; Claude 3 fared slightly worse on this task. A superior tool that is dedicated to converting math in pictures to LaTeX format is [MathPix.com](https://mathpix.com).

3.2.5 Generating titles & headlines

Another useful application is to come up with catchy headlines or paper titles, as illustrated by the following example based on the version of the abstract used in the

previous example:

Chat 17 (GPT-4): Brainstorm two or three catchy titles for a paper with the following abstract:

[Abstract from Chat 15 inserted]

1. "Revolutionizing Economic Research: The Power of Generative AI and Large Language Models"
2. "Unleashing AI in Economics: A Comprehensive Guide to Utilizing Large Language Models"
3. "The Future of Economic Research: Harnessing the Potential of Generative AI and Large Language Models"

Chat 17 (GPT-4) ctd.: Make them a bit more sober and keep them short.

1. "Generative AI in Economic Research"
2. "Large Language Models: A New Tool for Economists"
3. "The Role of AI in Modern Economic Studies"

The suggested titles fit the described work quite well and adhere closely to the requested styles.

3.3 Background Research

There are a variety of tasks that researchers perform in order to conduct background research and learn about both existing and new work in their research area. These include summarizing text, searching relevant literature, seeking to understand new concepts, and translating text from other languages.

Fall 2024 Status Update for “Background Research” Advances in background research have derived primarily from the capacity of frontier LLMs to process larger and larger amounts of text in their context window. This enables cutting-edge systems to process multiple academic papers or books and even YouTube videos with growing accuracy. However, when performing literature reviews, LLMs still struggle with synthesizing the state of the art on a given research topic.

3.3.1 Summarization [Updated in 2024Q2]

First, LLMs are very capable at summarizing. They are adept at extracting the most relevant content from a passage of text, as illustrated here:

Chat 18 (GPT-4): Summarize the following text in one sentence:
[Abstract from Chat 15 inserted]

The author discusses the potential of Generative AI, specifically large language models like ChatGPT, to revolutionize research in various domains such as ideation, writing, data analysis, coding, and mathematics, and suggests that economists can significantly increase productivity by automating micro tasks with AI, with the performance of these systems expected to improve over time.

The summary sentence it provided covers all the important bases.

Paper Summaries Both Claude and ChatGPT have now a sufficiently large context window to process entire academic papers as well as many books. To provide an example, I used the “Upload” button to feed the original Feb. 2023 NBER working paper version of this paper (Korinek, 2023b) into Claude 2 and asked for a short summary – “short” in order to not take up too much space here:

Chat 19 (Claude 2): Provide a short summary of this paper.
[Attachment: w30957.pdf]

This paper describes how large language models (LLMs) like ChatGPT are starting to display useful capabilities across a range of research tasks in economics. The author categorizes 25 potential use cases into six domains: ideation, writing, background research, data analysis, coding, and mathematical derivations. For each domain, the paper provides instructions and examples to demonstrate current LLM abilities, rating them on a scale from experimental to highly useful. The author argues LLMs can help automate micro tasks and increase researcher productivity in the short run. In the long run, the paper speculates cognitive automation via LLMs may have profound implications for economic research and the value of human cognitive labor.

The long context window of Claude implies that it can keep the entire text of a typical paper in its working memory.

The model can respond to follow-up questions on the content, going into significant detail on the text provided. Additional examples of useful prompts include:

- What is the evidence supporting these points [or a specific point from the summary]?
- What are the author’s main conclusions?
- What are the most interesting (or most novel) ideas in this paper?
- Can you explain the main findings of section [enter number]?

Claude allows the user to upload up to five documents and to ask the LLM to perform tasks such as comparing them or identifying connections between them.

Summaries of Videos [All New in 2024Q2 Version] Google DeepMind’s Gemini has the capacity to digest YouTube videos if the corresponding extension is activated in the settings of Gemini.

Chat 20 (Gemini): Provide a one-paragraph summary of this: <https://www.youtube.com/watch?v=sjL6G16ZIQs>

This video is about a new large language model, Claude 3 Opus, created by Anthropic. In the video, the creators test Claude 3 Opus’s ability to analyze the world economy. The video shows that Claude 3 Opus can access and process information from the web using a tool called web view. It can then use this information to answer questions and complete tasks. For example, the video shows that Claude 3 Opus can generate a graph of US GDP trends and predict future GDP growth. The video also shows that Claude 3 Opus can break down complex problems into subproblems and distribute the work among other versions of itself, which is called dispatch sub agents. This allows Claude 3 Opus to analyze the economies of multiple countries simultaneously. Overall, the video demonstrates that Claude 3 Opus is a powerful tool that can be used to analyze complex economic data.

The summary provided is excellent. This new capability makes it possible to streamline the lengthy process of retrieving information from videos in a much simpler form, providing new sources of information for economic research.

3.3.2 Literature Research [Significantly Updated in Fall 2024]

When writing the original version of this article (Korinek, 2023a), LLMs were of limited usefulness for searching and referencing the academic literature. When asked for citations or references, they were often able to provide standard references that are frequently cited in the literature (say, Rothschild-Stiglitz), but basic LLMs frequently hallucinated authoritative-sounding papers that do not actually exist or made other simple factual mistakes. The following is an example for GPT-4 in Fall 2023:

Chat 21 (GPT-4): Provide citation information for Anton Korinek’s paper with Zach Bethune.

Bethune, Z., & Korinek, A. (2021). Covid-19 Infection Externalities: Trading Off Lives vs. Livelihoods. *Journal of Monetary Economics*, 117, 1-13.

The paper title is correct but the publication information is hallucinated – and this is for a paper with over 200 citations. For small variations of my prompt (e.g., putting the

authors in alphabetic order), the LLM responded that it could not find such a paper. Requests for academic literature frequently lead to invented paper titles and journals that, however, sound superficially plausible. Buchanan et al. (2023) systematically examine the capability of GPT-3.5 to write about every JEL code and find that more than 30% of the citations generated do not exist.

Traditional LLMs struggle with academic references because they cannot retain their training data verbatim, but instead retain an abstract representation of it that involves some loss of information. While this approach works for many general concepts, it falls short for academic references where precision is crucial.

Literature research using LLM-powered search Recent advances in LLM-powered search have significantly improved the ability of LLMs to help with literature research. there are alternative tools available that employ LLMs to help researchers find academic references. For example, LLMs with web connections such as Bing or Bard can be prompted to look up specific papers on the internet and generate reference information from their findings. Moreover, the website <https://elicit.org> allows searches of the academic literature and displays a list of papers (that really exist) based on simple user prompts. It also includes a brief summary generated by an LLM.

3.3.3 Background Internet Research [All New in Fall 2024 Version]

As already observed in Section 5, LLM-powered search tools have also become powerful instruments for factual research. Like all LLMs, search tools based on LLMs sometimes hallucinates, but they do so much less frequently than LLMs that do not have the ability to access the web. Moreover, they offer links supporting the statements that they make, which the user can follow to judge the reliability of the sources and verify the model's output.

The following is an example:

Chat 22 (Perplexity.AI Pro): Can you explain the EU AI Act?

The EU AI Act is a groundbreaking piece of legislation that aims to regulate artificial intelligence systems within the European Union. Here's a comprehensive overview of its key aspects:

Risk-Based Approach

The AI Act classifies AI systems into four risk categories [1] [2]:

1. **Unacceptable risk:** These AI systems are prohibited outright.
2. **High risk:** Strictly regulated with specific requirements.
3. **Limited risk:** Subject to transparency obligations.
4. **Minimal risk:** Allowed with minimal restrictions.

Key Provisions

[2 pages of descriptions]

Conclusion

The EU AI Act represents the world’s first comprehensive legal framework for AI, aiming to foster innovation while protecting fundamental rights and ensuring ethical AI development. It sets a global standard for AI regulation and is likely to have far-reaching impacts on AI development and deployment worldwide.

After submitting a query, the site first shows the keywords it searches for (in my example, “EU AI Act explanation,” “European Union Artificial Intelligence Act overview,” and “What is the EU AI Act?”), then a preliminary list of the materials it evaluates as “Reading” (in my example, 11 websites). Then it generates a response that starts with a clickable list of sources, continues with a synthesis of the materials found, and ends with a conclusion that summarizes everything. The summary it provided is satisfactory, although I had to ask a follow-up question for it to report the date it was passed. Note that the leading LLM-based chatbots without search (e.g., Anthropic) were not able to correctly answer the question since their training data cut off before the passage of the act.

In my explorations, I found that LLM-powered search works well for general factual questions but regularly runs into difficulties in identifying the most relevant sources for more specialized queries. For example, when asking for academic literature on a given topic, a regular occurrence is that they base significant parts of their responses on obscure papers that happen to have a title that closely matches the query. Further progress will require that these tools better learn to prioritize the most impactful academic references, perhaps as measured by citation counts and journal and researcher impact factors.

3.3.4 Formatting References

Although they are really bad at generating references, even simple LLMs (such as GPT3.5) are quite capable at formatting references in the desired manner once the correct citation information is provided:

Chat 23 (GPT-4): Convert the following reference to bibtex format:

Anton Korinek, Large Language Models and Cognitive Automation for Economic Research, NBER Working Paper 30957, February 2023.

```
@techreport{korinek2023large,  
title={Large Language Models and Cognitive Automation for Economic Re-  
search},  
author={Korinek, Anton},  
year={2023},  
institution={NBER},  
type={Working Paper},
```

```
number={30957},  
month={February}  
}
```

Chat 23 (GPT-4) ctd.: Format it in Chicago style.

Korinek, Anton. 2023. "Large Language Models and Cognitive Automation for Economic Research." NBER Working Paper No. 30957, February.

LLMs can convert a whole bibliography in batch format, for example from APA to Chicago format or into bibtex format. For very highly-cited works that the LLM has frequently encountered in its training data, a prompt like “bibtex reference for stiglitz weiss” also works well and saves the user from having to copy or type the detailed citation information of the work in question. However, it is advisable to double-check the information.

3.3.5 Translating Text

Jiao et al. (2023) demonstrate that LLMs starting with GPT-3.5 perform competitively with commercial translation products on high-resource languages. The following is an example:

Chat 24 (GPT-4): Translate the following sentence into Spanish, German, and Finnish:

Generative AI, in particular large language models (LLMs) such as ChatGPT, has the potential to revolutionize research in many disciplines.

Spanish: La IA generativa, en particular los modelos de lenguaje grandes (LLMs) como ChatGPT, tiene el potencial de revolucionar la investigación en muchas disciplinas.

German: Generative KI, insbesondere grosse Sprachmodelle (LLMs) wie ChatGPT, hat das Potenzial, die Forschung in vielen Disziplinen zu revolutionieren.

Finnish: Generatiivinen tekoäly, erityisesti suuret kielen mallit (LLM) kuten ChatGPT, on potentiaalia mullistaa tutkimusta monilla aloilla.

During pre-training, LLMs develop representations of concepts that abstract beyond any single language, and this makes them able to easily map text from one language to the other. Performance on lower-resource languages for which less digitized text and fewer digitized translations are available is worse.

3.3.6 Explaining Concepts

LLMs can act as a tutor and explain many common economic concepts – at a level that is useful for students who are trying to learn new materials, and even for more advanced researchers who are branching out beyond their main area of expertise. For good explanations, it is critical to employ the most cutting-edge LLMs.

Chat 25 (GPT-4): What are instrumental variables useful for?

Instrumental variables are useful for addressing issues of endogeneity in statistical analysis, particularly in regression analysis. Endogeneity can occur due to unobserved variables, measurement errors, or simultaneity, which can lead to biased and inconsistent parameter estimates.

Instrumental variables are used to isolate the effect of the variable of interest from the effects of other confounding variables. They are particularly useful in situations where it is not possible to directly observe all relevant variables, or where the variable of interest is correlated with the error term.

In addition, instrumental variables can be used to estimate causal relationships in observational data, where random assignment of treatments is not possible. They provide a way to mimic the conditions of a randomized controlled trial, which is considered the gold standard for causal inference.

When employing LLMs in this way, follow-up questions that go into further detail can be very useful and allow the user to obtain personalized tutoring.

Other concept questions that the LLM answered well included the following:

- What is the invisible hand theorem and why does it hold?
- What is Monte Carlo simulation?
- How does Cournot equilibrium differ from Bertrand equilibrium?
- What are the leading theories of exchange rate determination?

But sometimes LLMs, especially earlier versions, hallucinate when providing answers. The following is an example using the older GPT-3.5 model text-davinci-003, which powered the original free version of ChatGPT in late 2022:

Chat 26 (GPT-3.5): What is the second theorem of welfare economics?

The Second Theorem of Welfare Economics states that any competitive equilibrium is Pareto efficient. This means that any competitive market outcome is the most efficient outcome possible, as it cannot be improved upon without making at least one individual worse off.

(Author's note: this is clearly the first not the second theorem!)

The two fundamental theorems of welfare economics are cornerstones of modern economics, but the text generated by GPT-3.5 confuses the two. Although LLMs give clear and impressive responses in many instances, this illustrates that their output cannot always be trusted, even on what seem like basic questions about fundamental concepts. Since LLMs can be quite helpful and powerful as tutors, one of the dilemmas they introduce is that students are likely to use them even though they occasionally produce false responses with great confidence.

3.4 Coding

LLMs are quite capable at coding tasks, especially in the languages python and R, for which the most online resources are available. LLMs can write, edit, modify, translate, or debug snippets of code based on instructions in plain English (or other natural languages). Moreover, they can act as tutors when using new libraries, functions, or even programming languages that the user is not very familiar with by quickly generating output that shows what libraries and functions are needed for specific types of operations or what syntactic structures to use in a given programming language. This allows the human programmer to consult the LLM and weave together code from many different snippets generated by it.

The reasons why LLMs are so proficient at coding include the following: There are vast repositories of code available online that are included in their training data, for example from GitHub. The syntax of computer code seems to be relatively easy to learn for these systems. Moreover, the AI labs producing cutting-edge LLMs themselves benefit from the code generation abilities of these systems, which provides them with strong incentives for improving these capabilities.

On ChatGPT Plus, Advanced Data Analysis has made coding tasks in python very convenient, as it can write code, execute it, learn from it, write follow-up code, and display the requested results. It also allows users to upload code, ask for specific modifications, and download it again to continue work on it. GitHub Copilot and tabnine offer seamless integration into Visual Studio and several other integrated development environments for programmers, and they work for many different programming and markup languages.¹⁷

Fall 2024 Status Update for “Coding” Coding is perhaps the category of research that has derived the greatest productivity gains from the advances of the past year. Tools like Copilot or Cursor (described in Section 2.2.1 can increasingly leverage LLMs to collaborate with researchers in quickly producing high-quality code. One of the

¹⁷While writing this paper, I used ChatGPT Plus to write LaTeX code. I asked it how to define the boxes around the sample prompts throughout the paper in a way that they are visually appealing and break across pages, and I asked it to produce Table 1 in LaTeX based on the table of contents for section 3. I used GitHub Copilot when working on the replication code for this section.

breakthroughs in recent months was OpenAI's o1 series of models, described in Section 2.1, which can reason through a coding problem before writing up the code.

3.4.1 Writing code

LLMs are very useful for writing a wide variety of code, including for standard programming tasks, data manipulation and repetitive tasks. For the purposes of this paper, I prompted ChatGPT Advanced Data Analysis to simulate a classic model in economics. The model occasionally asks the user for confirmation and feedback. The following reports my exchange:

Chat 27 (ChatGPT Advanced Data Analysis): Simulate the Solow growth model and plot the results in a four chart.

Alright, let's simulate the Solow growth model. To simplify things, let's consider the Solow-Swan model without technological progress. The model describes the evolution of capital (K), labor (L), and output (Y) in an economy. The basic equation for the change in capital stock over time is given by:

$$\frac{dK}{dt} = sY - \delta K$$

where:

- s is the savings rate
- Y is the total output, which is given by a production function, usually a Cobb-Douglas type: $Y = K^\alpha L^{1-\alpha}$,
- δ is the depreciation rate.

Assumptions for the model:

- Labor grows at a constant rate n , i.e., $L(t) = L(0)e^{nt}$,
- The savings rate s , the depreciation rate δ , and the labor growth rate n are all constants.

To get started, we need to specify the values of these parameters. For example, we can use the following:

- $s = 0.4$,
- $\delta = 0.1$,
- $n = 0.02$,
- $\alpha = 0.33$,

- initial capital stock $K(0) = 1$,
- initial labor $L(0) = 1$.

To simulate the model, we'll use the Euler method for integrating the differential equations. This method approximates the value of the function at a future time as the value at the current time plus the derivative of the function at the current time multiplied by the time step.

Then, we will plot the following four charts:

1. Capital over time,
2. Labor over time,
3. Output over time,
4. Capital per worker over time.

Does that sound good?

I just provided a short “Go ahead” to continue the generation of the program:

Chat 27 (ChatGPT Advanced Data Analysis) ctd.: Go ahead.

```
import numpy as np
import matplotlib.pyplot as plt

# Model parameters
s = 0.4
delta = 0.1
n = 0.02
alpha = 0.33
K0 = 1
L0 = 1
T = 100 # Total simulation time

# Time grid
dt = 0.01
times = np.arange(0, T, dt)

# Initialize arrays to store the values of K, L, Y over time
K = np.zeros(len(times))
L = np.zeros(len(times))
Y = np.zeros(len(times))
```



```

# Initial values
K[0] = K0
L[0] = L0
Y[0] = K0**alpha * L0**(1-alpha)

# Euler method
for t in range(len(times)-1):
    dK = s*Y[t] - delta*K[t]
    dL = n*L[t]
    K[t+1] = K[t] + dK*dt
    L[t+1] = L[t] + dL*dt
    Y[t+1] = K[t+1]**alpha * L[t+1]**(1-alpha)

# Capital per worker
K_L = K / L

# Create plots
fig, axs = plt.subplots(2, 2, figsize=(14, 10))

# Plot capital over time
axs[0, 0].plot(times, K)
axs[0, 0].set_title('Capital over time')
axs[0, 0].set_xlabel('Time')
axs[0, 0].set_ylabel('Capital')

# Plot labor over time
axs[0, 1].plot(times, L)
axs[0, 1].set_title('Labor over time')
axs[0, 1].set_xlabel('Time')
axs[0, 1].set_ylabel('Labor')

# Plot output over time
axs[1, 0].plot(times, Y)
axs[1, 0].set_title('Output over time')
axs[1, 0].set_xlabel('Time')
axs[1, 0].set_ylabel('Output')

# Plot capital per worker over time
axs[1, 1].plot(times, K_L)
axs[1, 1].set_title('Capital per worker over time')
axs[1, 1].set_xlabel('Time')

```

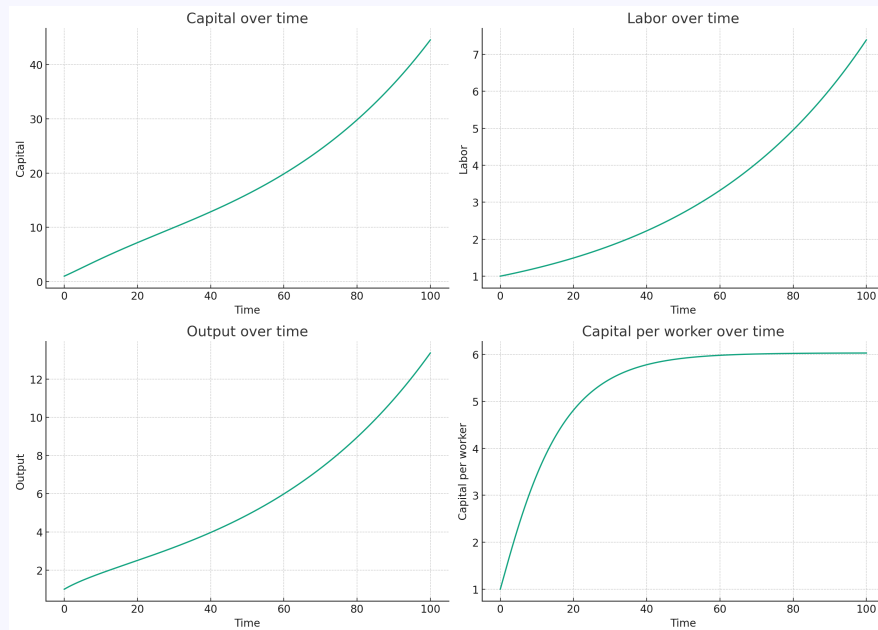
```

axs[1, 1].set_ylabel('Capital per worker')

# Make the layout tight
plt.tight_layout()

plt.show()

```



Here are the four plots for the Solow growth model:

1. **Capital over time:** The capital stock increases over time and approaches a steady state. This steady state is the point at which investment (saving) equals depreciation, so the capital stock no longer changes.
(Author's note: although the numerical simulation is correct, this point of the description is clearly wrong)
2. **Labor over time:** Labor grows exponentially at a constant rate, reflecting the assumption of constant population growth in the Solow model.
3. **Output over time:** Output also increases over time and approaches a steady state. This is because output is a function of capital and labor, which are both increasing. However, due to diminishing returns to capital, the growth of output slows down over time.
4. **Capital per worker over time:** Capital per worker also increases over time and approaches a steady state. This is the point at which the economy

achieves balanced growth in the Solow model. At the steady state, output per worker and capital per worker grow at the same rate, which is the rate of technological progress. In this simplified version of the model without technological progress, this growth rate is zero, so output per worker and capital per worker are constant at the steady state.

These results illustrate the key predictions of the Solow growth model: in the long run, an economy approaches a steady state level of capital (both total and per worker), and this steady state level is determined by the savings rate, the depreciation rate, and the rate of population growth. Changes in these parameters can lead to a higher or lower steady state, but they cannot sustain long-run economic growth. According to the model, long-run economic growth can only be sustained by technological progress.

The model performed the simulation task as requested. It started with a detailed description of the Solow model, introduced parameter assumptions, asked for user approval, and wrote and executed the correct python code. However, it also volunteered to interpret the results and incorrectly described the path of capital over time as “approaching a steady state.” Since it assumed positive population growth, this is clearly wrong and also contradicts the simulation displayed in the top left panel of the figure. (Advanced Data Analysis can generate figures but cannot actually “see” them so it simply generated what *seemed* like a plausible description of the figure, thereby introducing an error.) Moreover, in a concluding paragraph, the LLM wrote: “These results illustrate the key predictions of the Solow growth model: in the long run, an economy approaches a steady state level of capital (both total and per worker),” which – under positive population growth – is clearly impossible.

I also attempted to simulate the Ramsey model. The LLM correctly generated code to simulate the path of consumption and capital for given initial conditions, but it did not manage to solve for the correct starting point for consumption, even after I added a reminder to my prompt that initial consumption needs to lie on the stable arm of a saddle-path stable system.

This is representative of the state of LLM coding capabilities at the time of writing – LLMs are useful assistants to automate many of the mundane parts of programming, but they require human oversight for the more difficult parts. Moreover, LLMs can also act as powerful tutors that help users to find the packages and functions that they need for specific use cases.

Outlining a plan for a coding project [All New in 2024Q2] Cutting-edge LLMs, in particular Claude 3.5 Sonnet, also have the ability to outline a plan for how to approach coding projects. In the spirit of chain-of-thought prompting, they frequently deliver better results if first asked to outline a plan on how to code a programming task and then executing the plan.

3.4.2 Explaining code

Whereas the previous example wrote code based on English language instructions, LLMs can also do the reverse and explain what a given piece of code does in plain English, akin to a tutor. The following examples demonstrates this use case:

Chat 28 (GPT-4): What does the following function do?

```
def regression(X, y):
    X = np.hstack((np.ones((X.shape[0], 1)), X))
    XtXinv = np.linalg.inv(np.dot(X.T, X))
    beta1 = np.dot(np.dot(XtXinv, X.T), y)
    residuals = y - np.dot(X, beta1)
    V = np.diag(residuals**2)
    Vinv = np.linalg.inv(V)
    XtVinvXinv = np.linalg.inv(np.dot(X.T, np.dot(Vinv, X)))
    beta2 = np.dot(np.dot(XtVinvXinv, X.T), np.dot(Vinv, y))
    return beta1, beta2
```

This function performs a two-step generalized least squares regression. In the first step, it performs an ordinary least squares regression to estimate the parameters (beta1) of the model. In the second step, it uses the residuals from the first step to estimate the variance of the errors, and then performs a weighted least squares regression to estimate the parameters (beta2) of the model, taking into account the heteroscedasticity (non-constant variance) of the errors. The function returns the estimated parameters from both steps.

This functionality is particularly useful when using sources that have not documented their code well.

3.4.3 Translating code

LLMs can also convert code in one programming language into another language, for example to port a project to another platform or to integrate code snippets found in an online coding forum such as StackExchange that is in the wrong language.

The following prompt, building on the preceding conversation, provides an example:

Chat 28 (GPT-4) ctd.: Can you translate the function into Matlab?

Sure, here is the equivalent MATLAB code for the function:

```
function [beta1, beta2] = regression(X, y)
    X = [ones(size(X, 1), 1), X];
    XtXinv = inv(X' * X);
    beta1 = XtXinv * X' * y;
```

```

residuals = y - X * beta1;
V = diag(residuals.^2);
Vinv = inv(V);
XtVinvXinv = inv(X' * (Vinv * X));
beta2 = XtVinvXinv * X' * (Vinv * y);
end

```

Please note that in MATLAB, the operator for matrix multiplication is `'*`, not `'np.dot()'`. Also, the operator for element-wise power is `'.^'`, not `'**'`.

Current LLMs are quite reliable in translating short pieces of code across common programming languages. For longer sequences, human assistance is still required.

3.4.4 Debugging code

One of the most frustrating parts of programming is to debug code. LLMs also exhibit useful capabilities in this domain. They can not only help to interpret error messages but also verify code.

In my first example, I simply pasted a typical error message from python that is obtained when the user forgets to import a library, without any further instructions:

Chat 29 (GPT-4): Traceback (most recent call last):

```

File "C:/.../test.py", line 20, in <module>
X = np.array([[1, 2], [3, 4], [5, 6], [7,8]])
NameError: name 'np' is not defined

```

The error message is telling you that the name `'np'` is not defined. This usually happens when you try to use a library that you haven't imported yet.

In your case, `'np'` is a common alias for the numpy library. You can fix the error by importing numpy at the beginning of your script. Here's how you do it:

```

“python
import numpy as np
“

```

After adding this line, your script will know what `'np'` is and it should work as expected.

The LLM correctly interpreted my prompt as a request for help to debug the error. It recognized the source of the error, explained it, and provided the code needed to rectify it.

The next example, building on the python function two examples earlier but with an error introduced, illustrates how LLMs can help to catch mistakes such as typos in code:

Chat 30 (GPT-4): Can you debug the following function?

[Code from preceding example, with a spelling mistake introduced by changing the definition of the variable `Vinv` to `V_inv`.]

The function seems to be implementing a form of weighted least squares regression, but there are a few issues with it: 1. The function uses numpy functions and methods but numpy is not imported. You need to import numpy at the beginning of your script. 2. The variable ‘`Vinv`’ is not defined. It seems like it should be ‘`V_inv`’. Here is the corrected function:

[Generated correct code]

The first point observes that the code does not run unless the numpy package is imported (as in the previous example); the second point correctly caught the spelling mistake that I introduced.

More generally, LLMs are also sometimes able to detect when indices are mixed up, and to provide advice when one describes the ways in which a program malfunctions. They are less capable at detecting high-level mistakes in the algorithms underlying code – for this, human debugging is usually still required.

3.5 Data Analysis

LLMs can help locate data sources, format data, extract data from text, classify and score text, create figures, extract sentiment, and even simulate human test subjects. Most of these capabilities can be accessed not only through a web interface as shown in the demonstrations below, but also via an API (Application Programming Interface) that allows large amounts of data to be formatted, extracted, classified etc. The operations can also be performed in batches to remain within the token limit for each request. Moreover, building on the section on coding, it goes without saying that LLMs can write the computer code necessary to access their own APIs – for example, try out “Write python code to ask GPT-4 to do [any data extraction or manipulation task]”.

When performing data analysis tasks in bulk, cost is an important consideration. Although a single prompt to a cutting-edge LLM costs just fractions of a cent, the cost of performing thousands or millions of queries quickly adds up. For many of the tasks described below, smaller and cheaper models are available. In those cases, it is not advisable to use the most cutting-edge LLM.

Fall 2024 Status Update for “Data analysis” This category has benefited from the growing accuracy of LLMs, which now rival humans in tasks like classifying text or extracting sentiment. For individual researcher use, collaborative tools for data analysis in ChatGPT and Claude are becoming increasingly powerful and user-friendly. For bulk

data analysis, cost declines and better models are making it possible to process larger and larger amounts of data.

3.5.1 Locating Data sources [All New in 2024Q2 version]

LLMs can be useful starting from the first step of data analysis, i.e., in identifying and locating appropriate data sources. The following example illustrates the role that Claude 3 can play in advising users on what data sources to consider:

Chat 31 (Claude 3): What data would be useful to analyze the effects of AI on the labor market and where can I find it?

To analyze the effects of AI on the labor market, you would need data from various sources. Here are some key data points and potential sources:

1. Employment data by industry and occupation
 - Bureau of Labor Statistics (BLS) in the US
 - Eurostat for European countries
 - National statistical agencies of other countries
2. Adoption rates of AI technologies by industry
 - Market research reports from companies like Gartner, IDC, or McKinsey
 - Academic studies and surveys
 - Company reports and press releases

[Points 3. to 9. omitted for space reasons.]

Combining data from these various sources can provide a comprehensive picture of how AI is affecting the labor market across industries, occupations, and demographic groups. It's important to note that data availability and quality may vary across countries and regions.

Although the list provided by Claude is somewhat of a laundry list and generic, it serves as a good brainstorming basis that may give the researcher ideas about novel data sources. Moreover, Claude is also happy to offer follow-up advice on how to best download data sources and how to best clean them.

3.5.2 Creating figures

As described in Section 2.2.1 on workspaces for interactive LLM collaboration, one of the most useful functions of ChatGPT for economists is Advanced Data Analysis,

which employs the coding capabilities of GPT-4 to create versatile figures and graphs. In mid-2024, OpenAI made the resulting figures and graphs interactive.

In the following example, I uploaded a file that contained stock market prices for three large technology companies and instructed ChatGPT Advanced Data Analysis to create one graph displaying stock performance labeled with the corresponding betas and another graph displaying portfolio weights.¹⁸

Chat 32 (ChatGPT Advanced Data Analysis): Can you display a graph that shows the performance of these securities normalized to start at 100?

[Uploaded stock_data.csv]

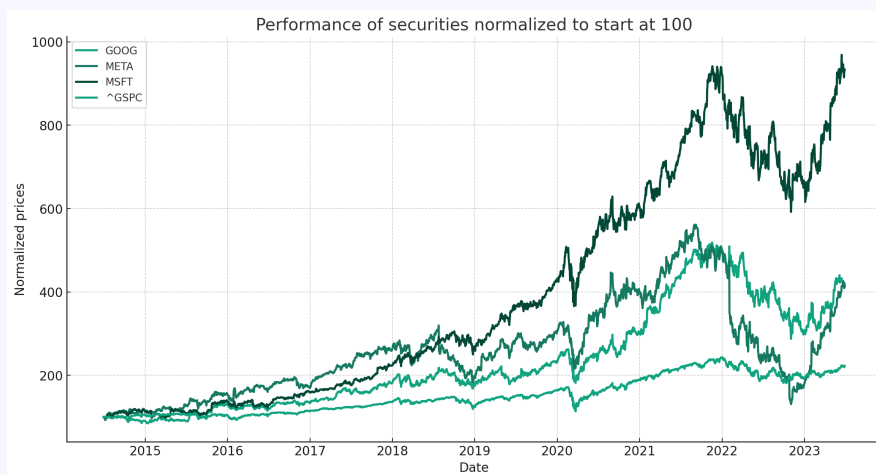
[Code generated and executed]

The dataset contains the closing prices for the securities Google (GOOG), Meta (META), Microsoft (MSFT), and the S&P 500 index (^GSPC) over a certain period of time.

Let's normalize the prices to start at 100 and plot the resulting performance of these securities. This normalization will allow us to compare the performance of the securities relative to their initial prices. To normalize the data, we'll divide all prices by their initial price and then multiply by 100.

[Code generated and executed]

[Code generated and executed]



Here is the graph showing the performance of the securities Google (GOOG), Meta (META), Microsoft (MSFT), and the S&P 500 index (^GSPC), all normalized to start at 100. The x-axis represents time (the dates), and the y-axis represents the normalized prices.

From the graph, we can see how each of these securities has performed relative

¹⁸To compile the underlying stock market data, I asked ChatGPT to write a script to download the data, as described in the Online Appendix of the paper.

to its starting point over time. This type of comparison can be helpful when assessing the relative performance of different securities or indices.

Chat 32 (ChatGPT Advanced Data Analysis) ctd.: Calculate the beta for each of the stocks and add it to the legend. Change the colors to make them easier to distinguish. Use the title "One-Decade Performance of Three Big Tech Companies".

The beta of a stock is a measure of the stock's volatility in relation to the overall market. The overall market has a beta of 1.0, and individual stocks are ranked according to how much they deviate from the market.

A stock that swings more than the market over time has a beta above 1.0. If a stock moves less than the market, the stock's beta is less than 1.0. High-beta stocks are supposed to be riskier but provide higher return potential; low-beta stocks pose less risk but also lower returns.

Let's calculate the beta for each of these stocks over the given period. We can calculate beta using the formula:

$$\beta = \frac{Cov(r_E, r_M)}{Var(r_M)}$$

where:

- r_E is the return of the stock
- r_M is the return of the market
- $Cov(r_E, r_M)$ is the covariance between the stock and market returns
- $Var(r_M)$ is the variance of the market returns

For the market return, we'll use the S&P 500 index (^GSPC). We'll calculate daily returns for each stock and the market, then calculate the beta for each stock.

After that, I'll recreate the graph with the new colors and the calculated betas included in the legend. The colors will be chosen to provide good contrast and clear visibility.

[Code generated and executed]

Here are the beta values for the stocks:

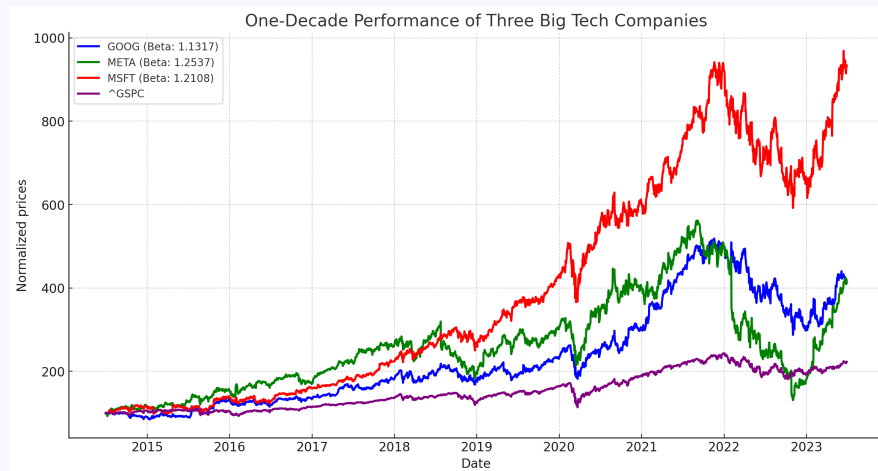
- Google (GOOG): 1.1317
- Meta (META): 1.2537

- Microsoft (MSFT): 1.2108

These values indicate that all three stocks are generally more volatile than the market (S&P 500), since their beta values are greater than 1.

Now let's update the graph with the new title, colors, and the beta values in the legend.

[Code generated and executed]

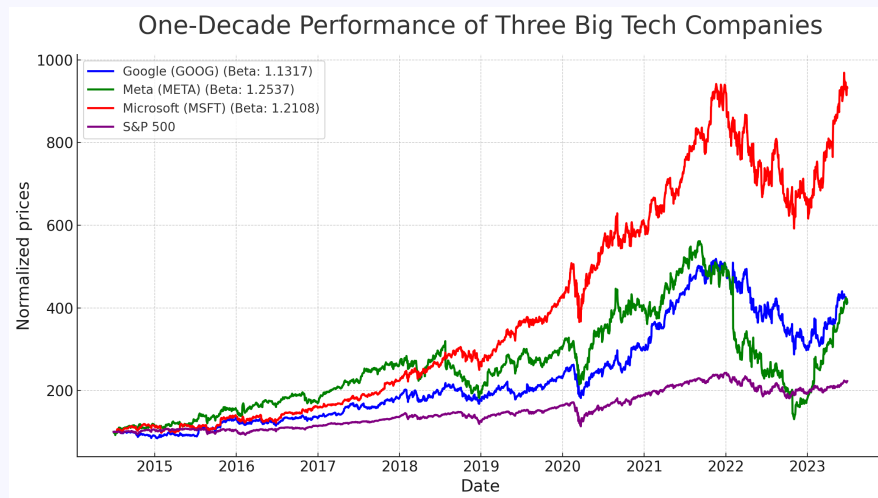


Here is the updated graph with the new title, colors, and beta values included in the legend.

The graph represents the one-decade performance of the three big tech companies: Google (GOOG), Meta (META), and Microsoft (MSFT), as well as the S&P 500 index (^GSPC), all normalized to start at 100. The beta values for the stocks, which measure their volatility in relation to the overall market, are included in the legend.

Chat 32 (ChatGPT Advanced Data Analysis) ctd.: Double the size of the title, legend, axes labels and ticks. Label "^GSPC" in the legend as "S&P 500".

[Code generated and executed]



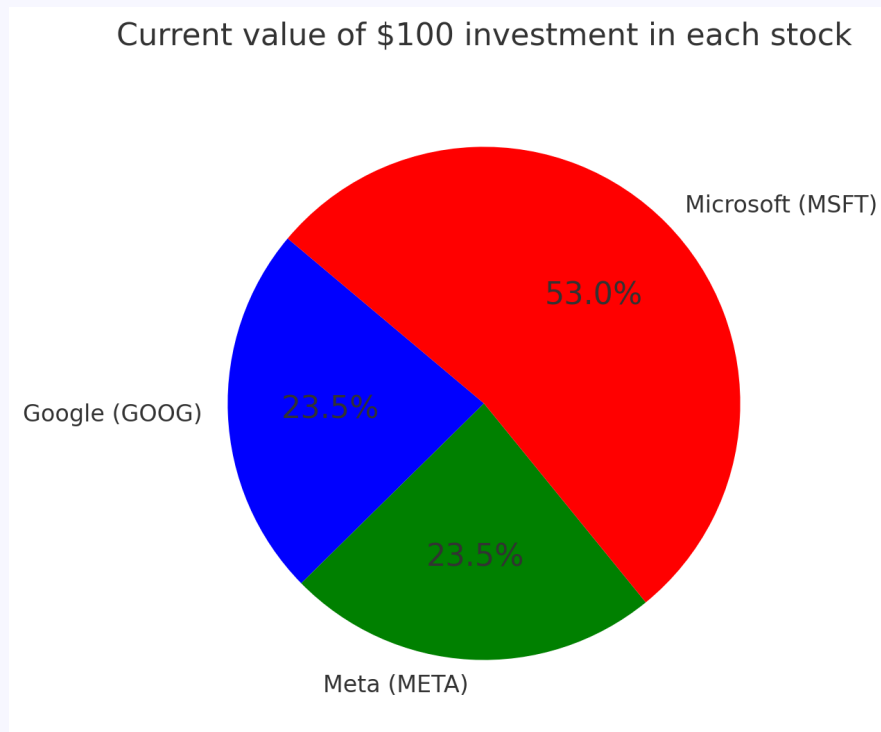
Here is the updated graph with the requested changes:

- The title, legend, axes labels, and ticks have all been doubled in size.
- The " \textasciitilde GSPC" label in the legend has been changed to "S&P 500".

This graph shows the one-decade performance of the three big tech companies: Google (GOOG), Meta (META), and Microsoft (MSFT), as well as the S&P 500 index, all normalized to start at 100. The beta values for the stocks are included in the legend.

Chat 32 (ChatGPT Advanced Data Analysis) ctd.: Assume I put \$100 in each of the three stocks at the start. Can you show a pie chart that shows how much I would have in each stock now?

[Code generated and executed]

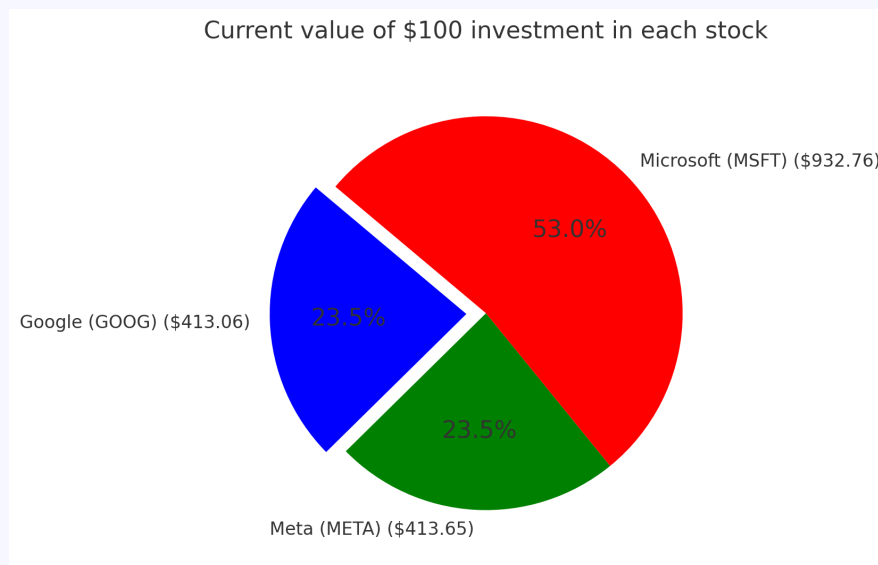


Here is a pie chart that shows the current value of a \$100 investment made in each of the three big tech stocks (Google, Meta, and Microsoft) at the start of the period represented in the data.

The pie chart is divided into three sections, each representing one of the stocks. The size of each section corresponds to the current value of the initial \$100 investment in that stock, expressed as a percentage of the total value of all three investments. The percentages are also labeled on the chart for clarity.

Chat 32 (ChatGPT Advanced Data Analysis) ctd.: Can you label each position with its current dollar value in addition to the percentage? Can you make one of the pieces pop out of the pie?

[Code generated and executed]



Here is the updated pie chart. Each slice now includes both the current dollar value and the percentage of the total portfolio that it represents.

The first slice, representing the Google stock, is "exploded" or offset from the pie to highlight it. This visual effect can be useful for emphasizing a particular piece of a pie chart. In this case, it emphasizes that Google has the highest value of the three stocks based on an initial \$100 investment.

(Author's note: the figure is correct, but ChatGPT's statement that Google has the highest value is false - this serves as a reminder that Advanced Data Analysis cannot "see" the figures it creates.)

Relying on the almost endless capabilities of python, ChatGPT Advanced Data Analysis can display many other types of figures.

3.5.3 Extracting data from text

A powerful capability of LLMs for data analysis is to extract data from written text. The following example is a simple illustration:

Chat 33 (GPT-4): Mark got an A in econ and a B+ in math. Sally got an A- in both econ and math. Frank got a B in econ and a C in math. Reformat as follows:

Name & econ grade & math grade

Mark & A & B+
Sally & A- & A-

Frank & B & C

The LLM correctly processes the meaning of the expression “both in econ and math” for Sally and extract the grades accordingly.

Other applications of extracting data from text include numerical data, e.g., stock prices from news articles or dosage information from drug databases. When I prompted the LLM with “Can you provide examples of what kinds of numbers you can extract from text?” it answered the following 10 types: phone numbers, zip codes, social security numbers, credit card numbers, bank account numbers, dates, times, prices, percentages, measurements (length, weight etc.) The process can be automated for large quantities of data using API access, and can typically performed with smaller and cheaper models than GPT-4. Dagdelen et al. (2024) show how to use LLMs for structured information extraction tasks from scientific texts. This can also be used in economics, for example, for entity recognition in economic history research.

3.5.4 Reformatting data

LLMs are equally versatile translating data into the desired format. Building on the previous example, I asked the LLM to format the data first into comma-separated values (CSV) format and then as a LaTeX table:

Chat 33 (GPT-4) ctd.: Reformat in CSV format.

```
Name,econ grade,math grade
Mark,A,B+
Sally,A-,A-
Frank,B,C
```

Chat 33 (GPT-4) ctd.: Write as a LaTeX table.

Name	econ grade	math grade
Mark	A	B+
Sally	A-	A-
Frank	B	C

GPT-4 excelled at both these and other similar tasks.

3.5.5 Classifying and scoring text

Social science research frequently employs statistical techniques to represent text as data (Gentzkow et al., 2019). Modern LLMs can go beyond traditional techniques for this because they are increasingly capable of processing the meaning of the sentences that they are fed.

The following example asks GPT-4 to classify whether a given task listed in the US Department of Labor’s Occupational Information Network (O*NET) database is easy or hard to automate and to justify its classification.¹⁹ Following the principle of chain-of-thought prompting suggested by ?, the prompt asks first for the justification in order to induce the LLM to reason about its response before performing the actual classification. This is akin to asking a student to think before they respond to a question.

Chat 34 (GPT-4): Here are 5 tasks from O*Net:

1. Monitor operational records and reports to ensure accuracy and efficiency.
2. Develop and implement policies and procedures for recordkeeping.
3. Prepare and maintain employee records.
4. Maintain and update filing, inventory, mailing, and database systems.
5. Compile and prepare reports, charts, and graphs.

Provide a 5-10-word description of how easy to automate each task is and classify into "easy" or "hard" to automate. Provide your output in the form of a LaTeX table with the following columns:

no; task; justification; classification

No	Task	Justification	Result
1	Monitor operational records and reports to ensure accuracy and efficiency.	Requires understanding of context and critical thinking.	Hard
2	Develop and implement policies and procedures for recordkeeping.	Requires human judgement and decision making.	Hard
3	Prepare and maintain employee records.	Mostly data entry and retrieval, easily automated.	Easy
4	Maintain and update filing, inventory, mailing, and database systems.	Mostly data entry and retrieval, easily automated.	Easy
5	Compile and prepare reports, charts, and graphs.	Data processing and visualization can be automated.	Easy

The results are reasonable, although I should emphasize that they are not entirely robust. Just like a human evaluator who may make different calls depending on seemingly random circumstances, the model’s answers to what is essentially the same question change when the wording of the prompt is changed – and in some cases even when the order of the listed tasks is modified.

¹⁹Eloundou et al. (2023) employ GPT-4 in this manner to systematically estimate the labor market impact of LLMs.

3.5.6 Extracting sentiment

LLMs can also extract sentiment from text. They can, for example, classify tweets into “positive” or “negative.” Similarly, they can categorize statements by the Federal Open Market Committee (FOMC) that sets US interest rates, as illustrated in the following example:

Chat 35 (GPT-4): Is the following FOMC statement from December 2022 hawkish or dovish?

Recent indicators point to modest growth in spending and production...
[From <https://www.federalreserve.gov/newsevents/pressreleases/monetary20221214a.htm>, remainder omitted for space reasons]

The statement is hawkish. The FOMC is indicating a concern about elevated inflation and is taking action by raising the target range for the federal funds rate and reducing its holdings of Treasury securities and agency debt. The statement also suggests that further increases in the target range may be appropriate in the future to return inflation to the 2 percent objective. This indicates a tightening of monetary policy, which is a characteristic of a hawkish stance.

The assessment is correct and well-argued.

I also explored whether the LLM could identify whether the December 2022 or February 2023 FOMC statement was more hawkish, but its ability to assess Fed-speak was not quite nuanced enough – it focused mainly on the level of interest rates in February 2023 being higher as opposed to the small and nuanced changes in the text of the statement that indicated a potential change in direction. It did so even when I explicitly instructed it to report its assessment while “disregarding the target level for the federal funds rate.” Only when I manually replaced the numbers for the target level by “[range]” did the system correctly replicate the assessment that the February 2023 statement was slightly more dovish, as was widely reported in the financial press at the time.²⁰ Ardekani et al. (2023) develop an economic sentiment prediction model along similar lines and employ it to analyze US economic news and the ECB’s monetary policy announcements.

3.5.7 Simulating human subjects [Updated in Fall 2024]

Argyle et al. (2022) propose the use of LLMs to simulate human subjects, based on the observation that the training data of LLMs contains a large amount of information about humanity. They condition GPT-3 on the socio-demographic backstories of real humans and demonstrate that subsequent answers to survey questions are highly correlated with the actual responses of humans with the described backgrounds, in a nuanced and multifaceted manner. Horton (2022) showcases applications to economics,

²⁰See, for example, <https://www.cnbc.com/2023/02/01/live-updates-fed-rate-hike-february.html>

using simulated test subjects to replicate and extend upon several behavioral experiments. Brand et al. (2023) demonstrate how to use LLMs to obtain useful information for market research.

The following example illustrates the concept:

Chat 36 (GPT-4): For each of the following voters, provide a one-word rating to indicate whether they are more likely to approve or disapprove of a policy proposal to loosen US immigration restrictions on highly-educated workers:

1. A 32-year old female lawyer in Houston, TX who is Hispanic
2. A 73-year old male retired doctor in Rome, GA who is Caucasian
3. A 25-year old software engineer in San Francisco, CA who is Asian and identifies as non-binary
4. A 45-year old female grocery check-out clerk in Brooklyn, NY who is second-generation Italian
5. A 55-year old male insurance agent in Lake City, FL who is Caucasian

1. Approve
2. Disapprove
3. Approve
4. Approve
5. Disapprove

There is a significant risk that the simulated results simply propagate false stereotypes, and they must hence be used with great care. However, they also contain valuable information. If used correctly, they can provide useful insights about our society, from which all the data used to train the LLMs ultimately originate. For experimental economists who prefer keeping to human subjects, Charness et al. (2023) describe how LLMs can help to improve the design and implementation of experiments. Chang et al. (2024) propose 12 best practices for generative AI in experimental research, ranging from the pre-treatment stage to the design and implementation stage, to the analysis stage.

3.6 Mathematical Derivations

The ability of general-purpose LLMs to perform mathematical derivations are growing but still limited at this point. The reasons for the relatively poor performance include that the training data for current LLMs is mostly text-based and includes comparatively little math, and that math requires high-level abstractions that are still difficult for current systems. Frieder et al. (2024) develop a dataset of graduate-level mathematical questions and show that GPT3.5’s mathematical abilities are significantly below those of an average mathematics graduate student – perhaps providing some solace for the short term. However, Noorbakhsh et al. (2021) show that LLMs can also be

fine-tuned for mathematical tasks. Moreover, there have been noticeable performance gains in math going from GPT-3.5 to GPT-4, as documented, e.g., by Bubeck et al. (2023). Moreover, datasets like the one created by Frieder et al. (2024) will certainly be useful for making future LLMs better at math. This is an area in which further progress would be very valuable for researchers.

Fall 2024 Status Update for “Mathematical Derivations” OpenAI’s o1 model marks a breakthrough in this sphere. The mistakes of older LLMs that are illustrated in the text below are no longer a challenge for this cutting-edge model. The ongoing advances in reasoning suggest that LLMs will continue to become better at math.

3.6.1 Setting up models

The following prompt is an example of how LLMs can be useful for setting up economic models. I prompted the LLM to generate LaTeX code that I could directly paste into my editor, generating the results shown below.

Chat 37 (GPT-4): Set up the optimization problem of a consumer with CES preferences over two goods in LaTeX code.

The optimization problem of a consumer with CES (Constant Elasticity of Substitution) preferences over two goods can be set up as follows in LaTeX code:

$$\begin{aligned} &\underset{x_1, x_2}{\text{maximize}} && (\alpha x_1^\rho + (1 - \alpha)x_2^\rho)^{\frac{1}{\rho}} \\ &\text{subject to} && p_1 x_1 + p_2 x_2 \leq M, \end{aligned}$$

In this problem, the consumer is trying to maximize their utility function, which is represented by a CES function. The variables x_1 and x_2 represent the quantities of the two goods that the consumer chooses. The parameters α and ρ are the CES preference parameters, p_1 and p_2 are the prices of the two goods, and M is the consumer’s income. The constraint represents the consumer’s budget constraint.

At the time of writing, cutting-edge LLMs were capable of generating the type of model setups that are commonly used in, for example, undergraduate problem sets. This is useful because the results appear in seconds and save time typing.

3.6.2 Deriving equations

LLMs also have some limited ability to derive equations. Continuing on the example above, I asked the LLM to solve the optimization problem that it set up:

Chat 37 (GPT-4) ctd.: Solve the optimization problem.

To solve this optimization problem, we need to use the Lagrangian method. The Lagrangian function is:

$$\mathcal{L}(x_1, x_2, \lambda) = (\alpha x_1^\rho + (1 - \alpha)x_2^\rho)^{\frac{1}{\rho}} + \lambda(M - p_1x_1 - p_2x_2)$$

We then take the first order conditions by differentiating the Lagrangian with respect to x_1 , x_2 , and λ and setting the derivatives equal to zero:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial x_1} &= \alpha x_1^{\rho-1} (\alpha x_1^\rho + (1 - \alpha)x_2^\rho)^{\frac{1}{\rho}-1} - \lambda p_1 = 0, \\ \frac{\partial \mathcal{L}}{\partial x_2} &= (1 - \alpha)x_2^{\rho-1} (\alpha x_1^\rho + (1 - \alpha)x_2^\rho)^{\frac{1}{\rho}-1} - \lambda p_2 = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= M - p_1x_1 - p_2x_2 = 0.\end{aligned}$$

Solving these equations simultaneously will give the optimal quantities of x_1 and x_2 that maximize the consumer's utility subject to their budget constraint. The solution will depend on the specific values of the parameters and the income and prices.

So far, the first-order conditions obtained are correct.

Chat 37 (GPT-4) ctd.: Derive the demand functions.

To derive the demand functions, we can use the first order conditions. Divide the first equation by the second to get:

$$\frac{x_1^{\rho-1}}{x_2^{\rho-1}} = \frac{p_1}{p_2}$$

(Author's note: the term $\frac{\alpha}{1-\alpha}$ is omitted here.)

Rearranging gives:

$$\frac{x_1}{x_2} = \left(\frac{p_1}{p_2}\right)^{\frac{1}{\rho-1}}$$

Substitute this into the budget constraint $M = p_1x_1 + p_2x_2$ to get:

$$M = p_1x_1 + p_2x_1 \left(\frac{p_2}{p_1}\right)^{\frac{1}{\rho-1}}$$

Solving for x_1 gives the demand function for good 1:

$$x_1 = \frac{M}{p_1 + p_2 \left(\frac{p_2}{p_1}\right)^{\frac{1}{\rho-1}}}$$

Similarly, we can derive the demand function for good 2:

$$x_2 = \frac{M}{p_2 + p_1 \left(\frac{p_1}{p_2} \right)^{\frac{1}{\rho-1}}}$$

These are the Marshallian demand functions for goods 1 and 2.

As the example illustrates, LLMs still routinely make mistakes in mathematical derivations. I entered “When you divided the two equations, you forgot the alpha terms.” The LLM proceeded to re-derive the results, corrected the mistake, but made a mistake in the following line. This underlines that current LLMs do not possess the high-level abstract representations necessary to perform math at a consistent level. They can serve, at best, as a writing assistant that quickly types lots of math but needs very careful oversight and potential editing.

3.6.3 Explaining models

Current LLMs also have some ability to explain simple models. This may be useful – but also risky – for students. In the following example, I pasted LaTeX code into the LLM and asked it to explain the underlying model and the steps in deriving a solution:

Chat 38 (GPT-4): Can you explain what this model is, how to get from one step to the next, and what the intuition for each step is?

[Equations pasted in LaTeX format]

$$\begin{aligned} \max_{C(t), K(t)} \quad & \int_0^\infty e^{-\rho t} U(C(t)) dt \\ \text{s.t.} \quad & \dot{K}(t) = F(K(t), L) - \delta K(t) - C(t) \\ & K(0) \text{ given} \\ & \lim_{t \rightarrow \infty} K(t) e^{-\rho t} \geq 0 \end{aligned}$$

$$\mathcal{H}(t, K(t), C(t), \lambda(t)) = e^{-\rho t} U(C(t)) + \lambda(t) [F(K(t), L) - \delta K(t) - C(t)]$$

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial C(t)} = 0 &\Rightarrow U'(C(t)) = \lambda(t) \\ \frac{\partial \mathcal{H}}{\partial K(t)} + \dot{\lambda}(t) = 0 &\Rightarrow \dot{\lambda}(t) = -\lambda(t) [F'(K(t), L) - \delta] \\ \lim_{t \rightarrow \infty} \lambda(t) K(t) &= 0 \end{aligned}$$

This is a model of optimal consumption and capital accumulation, often referred to as the Ramsey-Cass-Koopmans model in economics. It is a continuous-time version of the neoclassical growth model, where a representative agent chooses consumption and capital accumulation paths to maximize lifetime utility.

The agent's utility is discounted over time at rate ρ , and depends on consumption $C(t)$ according to the utility function $U(C(t))$. The agent's capital stock $K(t)$ evolves according to the law of motion given by the differential equation, where $F(K(t), L)$ is the production function, δ is the depreciation rate of capital, and $C(t)$ is consumption.

The Hamiltonian function, denoted by $H(t, K(t), C(t), \lambda(t))$, is a tool used in optimal control theory to solve this kind of dynamic optimization problem. It is a function of time t , state variable $K(t)$, control variable $C(t)$, and costate variable $\lambda(t)$, which is the shadow price of capital.

The first-order conditions for the optimal control problem are derived by taking derivatives of the Hamiltonian with respect to the control **and state variables, and setting them equal to zero**. These conditions characterize the optimal paths of consumption and capital over time.

The first condition $U'(C(t)) = \lambda(t)$ implies that the marginal utility of consumption is equal to the shadow price of capital. This means that the agent is indifferent between consuming an additional unit of output and saving it to increase the capital stock.

The second condition, the costate equation, describes the evolution of the shadow price of capital over time. It says that the rate of change of the shadow price is equal to the negative of the shadow price times the marginal product of capital net of depreciation. This reflects the trade-off between consuming today and saving for future consumption.

The transversality condition, $\lambda(t)K(t) = 0$ as t goes to infinity, ensures that the agent does not accumulate an infinite amount of capital, which would not be optimal because of the assumption of diminishing marginal utility of consumption. This condition rules out paths where the agent accumulates capital indefinitely and never consumes it.

The LLM correctly identified the model that was described in the math and explained the derivations mostly correctly, except for one small point, which I highlighted in bold: the LLM incorrectly suggested that the derivative of the Hamiltonian with respect to state variables be set to zero. Since most of the explanation is useful, it is likely that many students will resort to LLMs to tutor them on examples like the one presented. Yet this comes with the risk that part of the information will be incorrect – the LLM is not an infallible tutor.

3.7 Promoting Research [New Section in Fall 2024]

After having written a paper, the next step is to promote it. There are few areas in which LLMs have been as impactful as in marketing. This is particularly relevant for research papers, which are highly structured and information-dense self-contained pieces of knowledge—the current generation of LLMs has become capable of processing the content of entire research papers and are excellent at transforming that content into different formats, including formats that are conducive for promotion. Accordingly, there are several excellent use cases of LLMs for research promotion.

3.7.1 Social media posts [Updated in Fall 2024]

For the following exercise, I uploaded the original working paper version of this paper (Korinek, 2023b) to GPT-4o and asked it to generate social media posts, in this particular example for X (formerly Twitter):

Chat 39 (GPT-4o): Write a series of X posts to promote the attached paper. Use simple sentences and make them catchy.

[Attachment: w30957.pdf]

1/Exciting new possibilities! My latest NBER paper explores 25 ways language models like ChatGPT can boost research in economics. From ideation to coding, LLMs are transforming our workflows! #AI #EconResearch #LLM

2/Did you know? Large language models are now capable of automating micro-tasks like data analysis, writing, and even mathematical derivations in economic research. Check out my paper for more insights! #AIinEconomics #ProductivityBoost

3/Researchers, take note! LLMs aren't just about writing—they can also help with background research, coding, and more. Discover 6 key areas where they can assist in my latest paper. #CognitiveAutomation #AI #Economics

4/What's the future of cognitive labor? My paper dives into how LLMs could reshape the role of human economists, making us more efficient while raising new questions about the future of work. #FutureOfWork #AIRevolution

5/Want to stay ahead in research? Learn how to incorporate LLMs into your daily tasks and increase productivity. Don't miss the insights in my NBER paper! #Economics #AItools #GenerativeAI

The LLM adapts to the optimal format for different social media platforms. Upon request, it can also incorporate suitable emoticons or links.

3.7.2 Presentation slides [All New in 2024Q2 version]

Successful promotion leads to invitations to present your paper. The latest generation of LLMs released in 2024 also have the capability to draft slides based on a research paper. This is a difficult exercise because it requires that the model exhibits detailed recall across a long paper.

Chat 40 (Claude 3): Can you create a 20-slide deck for the attached economic research paper?

[Attachment: w30957.pdf]

Here is a 20-slide deck summarizing the key points from the research paper "Language Models and Cognitive Automation for Economic Research" by Anton Korinek:

Slide 1: Title Slide

- Paper title: "Language Models and Cognitive Automation for Economic Research"
- Author: Anton Korinek

Slide 2: Introduction

- Large language models (LLMs) can revolutionize economic research
- Describes 25 use cases of LLMs across 6 domains

- Provides instructions and examples for each use case
- Classifies usefulness of capabilities from experimental to highly useful

Slide 3: What are LLMs?

- LLMs are foundation models
- large deep learning models pre-trained on lots of data
- Form abstract representations and world model from training data
- Capabilities improve according to predictable scaling laws as compute and data increase
- Many capabilities are emergent, arising suddenly at specific thresholds

[Slides 4 – 18 omitted for space reasons]

Slide 19: Importance of Economic Ideas

- Economists can translate human concepts to analytic forms for AIs
- Economic ideas will influence worldview of future AI systems
- "Soon or late, it is ideas which are dangerous for good or evil" (Keynes)

Slide 20: Conclusion

- LLMs are becoming useful for automating research tasks
- Humans still have comparative advantage in generating research for now
- Important to adopt LLMs to boost productivity
- Prepare for potentially transformative long-term impacts

The generated slides provide an excellent first draft on which to build. What is impressive about the latest version of Claude that was used in the example is that it recalled the information contained in the paper in sufficient detail so that it could create 20 slides full of appropriate material without starting to forget content. It also chunked the material appropriately into the requested number of slides.

3.7.3 Blog Posts [All new in Fall 2024]

One of the most effective ways of promoting research papers to a broader audience is to write blog posts that convey their message in simple terms. This is an easy task for modern LLMs—users can upload the underlying paper to provide the content, and the

LLM can create a blog post to the user's specifications. The following is an example that builds on the prompt that generated with Anthropic Prompt Generator in Chat 4 to draft a summary of the first version of this paper (Korinek, 2023b).

Chat 41 (GPT-4o): Your task is to help draft a blog post for the Brookings Institution based on an economics research paper. The full text of the paper is attached.

The target audience for this blog post are economic researchers.

[continuing with the prompt generated in Chat 4]

[Attachment: w30957.pdf]

<scratchpad> [Omitted for space reasons]

<outline> [Omitted for space reasons]

<blog_post_draft>

How AI Is Reshaping Economic Research: Exploring Language Models for Cognitive Automation

Recent advances in artificial intelligence have brought transformative tools like large language models (LLMs) into the toolkit of economic researchers. These models, such as OpenAI's ChatGPT, are not just conversational agents; they represent a major leap in how cognitive tasks can be automated, promising to significantly alter the landscape of research in economics. In his recent paper, "Language Models and Cognitive Automation for Economic Research," Anton Korinek lays out 25 ways that these models can be leveraged, offering insights that are highly relevant for economic researchers looking to understand the potential impact of AI on their field.

Six Domains of LLM Utility in Economic Research

Korinek's paper categorizes the applications of LLMs into six key domains. These applications span from idea generation to coding, offering a comprehensive toolkit for researchers.

- **Ideation and Writing:** LLMs can assist researchers from the very start of a project, helping brainstorm ideas, generate counterarguments, and refine initial drafts.

[Remainder omitted for space reasons]

The draft blog posts generated by the current generation of frontier LLMs are generally well done and, after some editing, save significant time compared to having to write such a post from scratch.

3.7.4 Conducting interviews [All new in Fall 2024]

Large language models (LLMs) have emerged as powerful tools for conducting and assisting with interviews related to academic research. Their capabilities in this domain can be leveraged in two primary ways: answering interview questions and generating insightful questions for various audiences.

LLMs can be used to answer interview questions on behalf of researchers, using their published work as a knowledge base. This approach is particularly useful when researchers are preparing responses for written interviews or need assistance in formulating clear and concise answers. For example, using a research paper as a reference, an LLM could provide detailed answers to questions about the impact of AI on economic research.

Chat 42 (Claude 3.5 Sonnet): I received an interview request on the impact of AI on economic research from The Chronicle of Higher Education. Can you help me draft responses based on the attached paper of mine? I am pasting the questions here:

1. You describe 25 use cases for large language models (LLMs) in economic research across six domains. Which of these capabilities do you think has the most potential to revolutionize how economists conduct research in the near term?
2. In the paper, you speculate on the longer-term implications of cognitive automation via LLMs for economic research. Could you elaborate on how you envision the role of human economists evolving as LLMs become more advanced?
3. You mention that LLMs can be used to simulate human subjects for research. What are the ethical considerations and potential pitfalls of this approach, and how might it change the landscape of behavioral economics?
4. The paper touches on the limitations of current LLMs, particularly in areas like mathematical derivations. How do you see these limitations being addressed in future iterations of LLMs, and what implications might this have for theoretical economics?
5. You draw parallels between the rise of LLMs and Garry Kasparov's observations about chess computers. How do you think economists should prepare for a future where AI systems might produce superior economic models? What skills should economists focus on developing?

[Attachment: w30957.pdf]

Certainly! I'd be happy to help you draft responses based on your paper. I'll provide suggested responses for each question, drawing directly from the content and ideas presented in your paper.

1. You describe 25 use cases for large language models (LLMs) in economic research across six domains. Which of these capabilities do you think has the most potential to revolutionize how economists conduct research in the near term?

Suggested response: "Based on my analysis, I believe the capabilities related

to writing and data analysis have the most potential to revolutionize economic research in the near term. LLMs are particularly adept at tasks like synthesizing text, editing, and generating catchy titles or summaries. These capabilities can significantly streamline the writing process, allowing researchers to focus more on ideas and analysis rather than the mechanics of writing.

In terms of data analysis, LLMs show great promise in extracting data from text, reformatting data, and classifying and scoring text. These capabilities can automate many time-consuming tasks in data preparation and initial analysis, potentially accelerating the research process.

It's worth noting that I've rated several capabilities in these domains as 'highly useful' in my paper, indicating that incorporating them into research workflows can already save significant time and boost productivity. As these tools continue to improve, I expect their impact on how we conduct research to grow substantially."

2. In the paper, you speculate on the longer-term implications of cognitive automation via LLMs for economic research. Could you elaborate on how you envision the role of human economists evolving as LLMs become more advanced?

Suggested response: "As LLMs become more advanced, I anticipate a shift in the role of human economists. In the medium term, I expect LLM-based assistants and tutors to become increasingly useful for generating content that makes up research papers. Human researchers will likely focus more on their comparative advantages - posing questions, suggesting directions for obtaining answers, discriminating which parts of the produced content are useful, editing, and providing feedback.

In essence, the role may evolve to be more akin to that of an advisor or research manager. Economists will likely spend more time organizing research efforts, coordinating teams, and procuring data sources.

However, in the longer term, as LLMs continue to advance, the need for human inputs, edits, and feedback may diminish. We may find ourselves increasingly just rubber-stamping the output produced by ever-more advanced LLMs. Ultimately, it's possible that sufficiently advanced AI systems may be able to produce and articulate superior economic models, potentially making the cognitive work of human economists redundant.

This potential future underscores the importance of economists focusing on tasks that leverage uniquely human traits and adapting to work alongside increasingly capable AI systems."

[Remainder omitted for space reasons.]

The LLM provided well-crafted responses that accurately reflect the perspective provided in the attached background paper from February 2023. In its response to the first question, it even employed the classical human trick of punting the exact question (asking for *one* specific capability) and focusing on a more general question that

summarizes several categories of use cases.

This approach works well because the LLM can draw on the paper it was provided to generate insightful and relevant responses that align with the researcher's work and the interviewer's audience. The model maintains consistency with the original research while adapting the language to suit an interview format, making complex ideas more accessible to a broader audience.

LLMs can also be employed to generate thought-provoking and relevant interview questions. This capability can be tailored to different target audiences, enhancing the impact and reach of research promotion efforts. For instance:

Chat 43 (Claude 3.5 Sonnet): I am a journalist working for the Chronicle of Higher Education. I enjoyed reading the attached paper and scheduled an interview with the author. What would be five good questions to ask him based on the content of the paper?

[Attachment: w30957.pdf]

Based on the content of the paper, here are five good questions you could ask the author in your interview:

1. You describe 25 use cases for large language models (LLMs) in economic research across six domains. Which of these capabilities do you think has the most potential to revolutionize how economists conduct research in the near term?
2. In the paper, you speculate on the longer-term implications of cognitive automation via LLMs for economic research. Could you elaborate on how you envision the role of human economists evolving as LLMs become more advanced?
3. You mention that LLMs can be used to simulate human subjects for research. What are the ethical considerations and potential pitfalls of this approach, and how might it change the landscape of behavioral economics?
4. The paper touches on the limitations of current LLMs, particularly in areas like mathematical derivations. How do you see these limitations being addressed in future iterations of LLMs, and what implications might this have for theoretical economics?
5. You draw parallels between the rise of LLMs and Garry Kasparov's observations about chess computers. How do you think economists should prepare for a future where AI systems might produce superior economic models? What skills should economists focus on developing?

These questions cover a range of topics from the paper, including specific capabilities of LLMs, their potential impact on the field of economics, ethical considerations, current limitations, and future implications for the profession. They should provide a good basis for an in-depth discussion with the author about the key themes of his paper.

The questions—which happen to be the ones answered in Chat 42—are well-chosen and cover the most important parts of the paper. The capability of generating insightful

questions is useful not only for journalists but also for economists for two key reasons. First, economists often find themselves in situations where they are asked to suggest questions that would lead to interesting conversations targeted at specific audiences. Journalists, podcast hosts, and moderators at panel events regularly turn to their interviewees or speakers for such input. LLMs can be an invaluable tool in these scenarios, helping economists quickly generate relevant and engaging questions tailored to different audiences. For instance, the questions generated in the previous example would be markedly different from those aimed at a business newspaper, a policy think tank, or a general interest magazine. By leveraging LLMs, economists can effectively shape the discourse around their research and ensure it resonates with diverse audiences.

Second, LLMs allow researchers to prepare thoroughly for high-stakes interviews. In such cases, the prompt could be modified to generate challenging questions that the researcher might face, for example, by adding a phrase like “What would be some challenging questions that I may face?” This approach can help to anticipate difficult questions (especially if the interviewers themselves turn to LLMs) and prepare thoughtful responses in advance.

3.7.5 Podcasts [All new in Fall 2024]

Perhaps the most impressive tools when it comes to promoting research papers is the ability of Google’s NotebookLM to generate ‘deep dive’ podcasts. The system works by first generating an outline of the source material and identifying linkages, revising that outline, producing a detailed version of the script, performing a round of critique and the associated modifications, and then adding disfluencies to make the conversation sound more natural. An example is reproduced in Chat 3 on page 22.

4 Conclusions [All new in Fall 2024]

This paper has described the rapidly evolving landscape of large language models (LLMs) and their applications in economic research in Fall 2024. We explored dozens of use cases across seven key domains: ideation and feedback, writing, background research, coding, data analysis, mathematical derivations, and research promotion. The capabilities of LLMs have advanced dramatically even in the short time since the first version of this paper was published in December 2023, with new developments like improved reasoning abilities, interactive collaboration workspaces, real-time voice assistants, and powerful LLM-powered search emerging in recent months. These tools are proving ever more useful for automating many tasks in the research process, with the potential to significantly boost productivity.

Experiencing this breakneck pace of progress firsthand serves as a stark reminder that we urgently need to examine the broader economic and societal implications of ever-improving artificial intelligence systems. As economists, we have a responsibility to

help prepare for the myriad policy challenges that the Age of AI will bring Korinek (2024). A simple yet crucial exercise for all researchers is to consider: how would transformative advances in AI affect your specific area of study? By engaging with this question proactively, we can better anticipate and shape the impacts of AI on our disciplines and society at large.

The rapid efficiency gains in AI as an input to research also necessitate rethinking our optimal research methodologies. As LLM capabilities grow, it becomes increasingly beneficial to incorporate these tools into our workflows. However, the pace of advances also raises questions about the optimal timing of certain research projects. Some inquiries may be better postponed, as imminent AI advances could soon make them far easier to tackle. Researchers must weigh the benefits of immediate pursuit against the potential for dramatically improved capabilities for certain tasks in the near future. Lastly, as the production of research papers becomes cheaper and easier with AI assistance, we face an urgent challenge in evaluation. The bottleneck in the research process may shift from generation to assessment of ideas and results. This highlights the pressing need to develop robust methods for evaluating AI-augmented content, including exploring the feasibility of LLM-assisted peer review processes. While this presents significant challenges, it also offers opportunities to reimagine and improve our systems of knowledge validation.

Despite these complexities, the integration of LLMs and other AI tools into economic research also creates some cause for optimism. These technologies have the potential to dramatically accelerate the pace of discovery, allowing us to tackle increasingly complex problems and generate novel insights. As we experience this transition, our collective task is to use the power of AI responsibly, ensuring that it enhances rather than diminishes the quality and impact of our work. By embracing these tools thoughtfully and critically, we may be able to benefit from unprecedented productivity, creativity, and a greater capacity to address the world’s most pressing challenges.

References

- Anderljung, M., Barnhart, J., Korinek, A., Leung, J., O’Keefe, C., Whittlestone, J., and et al. (2023). Frontier AI regulation: Managing emerging risks to public safety. *arXiv:2307.03718*.
- Ardekani, A. M., Bertz, J., Dowling, M. M., and Long, S. (2023). EconSentGPT: a universal economic sentiment engine? SSRN Working Paper.
- Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J., Rytting, C., and Wingate, D. (2022). Out of one, many: Using language models to simulate human samples. *arXiv:2209.06899*.
- Ash, E., Hansen, S., and Muvdi, Y. (2024). Large language models in economics. *CEPR Discussion Paper DP19479*.

- Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., and others (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv:2212.08073*.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., and others (2021). On the opportunities and risks of foundation models. *arXiv:2108.07258*.
- Brand, J., Israeli, A., and Ngwe, D. (2023). Using GPT for market research. *Harvard Business School Working Paper*, 23(062).
- Bsharat, S. M., Myrzakhan, A., and Shen, Z. (2024). Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., and others (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv:2303.12712*.
- Buchanan, J., Hill, S., and Shapoval, O. (2023). ChatGPT hallucinates non-existent citations: Evidence from economics. *The American Economist*.
- Chang, S., Kennedy, A., Leonard, A., and List, J. A. (2024). 12 best practices for leveraging generative AI in experimental research.
- Charness, G., Jabarian, B., and List, J. A. (2023). Generation next: Experimentation with AI. *Working Paper, University of Chicago*.
- Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., and Stoica, I. (2024). Chatbot arena: An open platform for evaluating LLMs by human preference.
- Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., and Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.
- Dathathri, S., See, A., Ghaisas, S., Huang, P.-S., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., Hayes, J., Vyas, N., Al Merey, M., Brown-Cohen, J., Bunel, R., Balle, B., Cemgil, T., Ahmed, Z., Stacpoole, K., Shumailov, I., Baetu, C., Gwal, S., Hassabis, D., and Kohli, P. (2024). Scalable watermarking for identifying large language model outputs. *Nature*, 634:818–823.
- Dell, M. (2024). Deep learning for economists. *Journal of Economic Literature*, forthcoming.
- Dell’Acqua, F., McFowland III, E., Mollick, E., Lifshitz-Assaf, H., Kellogg, K. C., Rajendran, S., Kraymer, L., Candelon, F., and Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality. *Harvard Business School Working Paper*, (24-013).

- Eloundou, T., Manning, S., Mishkin, P., and Rock, D. (2023). GPTs are GPTs: An early look at the labor market impact potential of large language models.
- Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P. C., Chevalier, A., and Berner, J. (2024). Mathematical capabilities of ChatGPT. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23.
- Gentzkow, M., Kelly, B. T., and Taddy, M. (2019). Text as data. *Journal of Economic Literature*, 57(3):535–74.
- Girotra, K., Meincke, L., Terwiesch, C., and Ulrich, K. T. (2023). Ideas are dimes a dozen: Large language models for idea generation in innovation. *Working Paper*.
- Girotra, K., Terwiesch, C., and Ulrich, K. T. (2010). Idea generation and the quality of the best idea. *Management science*, 56(4):591–605.
- Heikkilä, M. (2024). How ChatGPT search paves the way for AI agents: Here’s what needs to be done to make AI assistants truly helpful. *MIT Technology Review*. <https://www.technologyreview.com/>, accessed on November 10, 2024.
- Ho, A., Besiroglu, T., Erdil, E., Owen, D., Rahman, R., Guo, Z. C., Atkinson, D., Thompson, N., and Sevilla, J. (2024). Algorithmic progress in language models.
- Horton, J., Filippas, A., and Horton, R. (2024). EDSL: Expected parrot domain specific language for AI powered social science. Whitepaper, Expected Parrot.
- Horton, J. J. (2022). Large language models as simulated economic agents: What can we learn from homo silicus? *NBER Working Paper w31122*.
- Huang, J. and Chang, K. C.-C. (2023). Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics (ACL)*, page 15.
- Jiao, W., Wang, W., Huang, J.-t., Wang, X., and Tu, Z. (2023). Is ChatGPT a good translator? A preliminary study. *arXiv:2301.08745*.
- Jones, N. (2024). In awe: Scientists impressed by latest ChatGPT model o1. *Nature*, 634:275–276.
- Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux, New York.
- Korinek, A. (2023a). Generative AI for economic research: Use cases and implications for economists. *Journal of Economic Literature*, 61(4):1281–1317.
- Korinek, A. (2023b). Language models and cognitive automation for economic research. *NBER Working Paper*, 30957.

- Korinek, A. (2024). Economic policy challenges for the Age of AI. *NBER Working Paper*, w32980.
- Korinek, A. and Vipra, J. (2025). Concentrating intelligence: Scaling and market structure in artificial intelligence. *Economic Policy*, *forthcoming*.
- Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. (2024). The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv:2408.06292*.
- Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., and Farajtabar, M. (2024). GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv:2410.05229*.
- Noorbakhsh, K., Sulaiman, M., Sharifi, M., Roy, K., and Jamshidi, P. (2021). Pre-trained language models are symbolic mathematics solvers too! *arXiv:2110.03501*.
- OpenAI (2024). Learning to reason with LLMs. *OpenAI blog*, <https://openai.com/index/learning-to-reason-with-llms>. Accessed: 2024-10-08.
- Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot.
- Plaat, A., Wong, A., Verberne, S., Broekens, J., van Stein, N., and Bülçöck, T. (2024). Reasoning with large language models, a survey. *arXiv:2407.11511*.
- Seeger, E., Dreksler, N., Moulange, R., Dardaman, E., Schuett, J., Wei, K., Winter, C., Arnold, M., hEigertaigh, S. O., Korinek, A., Anderljung, M., Bucknall, B., Chan, A., Stafford, E., Koessler, L., Ovadya, A., Garfinkel, B., Bluemke, E., Aird, M., Levermore, P., Hazell, J., and Gupta, A. (2023). Open-sourcing highly capable foundation models: An evaluation of risks, benefits, and alternative methods for pursuing open-source objectives.
- Shorten, C., Pierse, C., Smith, T. B., Cardenas, E., Sharma, A., Trengrove, J., and van Luijt, B. (2024). StructuredRAG: JSON response formatting with large language models.
- Si, C., Yang, D., and Hashimoto, T. (2024). Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. *arXiv:2409.04109*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.
- Wu, T., Lan, J., Yuan, W., Jiao, J., Weston, J., and Sukhbaatar, S. (2024). Thinking llms: General instruction following with thought generation.

- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. (2024). Tree of thoughts: Deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23.
- Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. *arXiv:2307.15043*.