Generative AI for Economic Research: Use Cases and Implications for Economists*

by Anton Korinek†

September 2023. This is a living paper - check back soon for updates.

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.

JEL Codes: A10, B41, J23, O3

---

*Accepted, *Journal of Economic Literature.* The online resources associated with this paper provide instructions for how to get started with using Generative AI in economic research. They will also provide regular updates on the latest capabilities of Generative AI that are useful for economists. They will soon be available on the journal website `https://www.aeaweb.org/journals/JEL` and at `https://www.aeaweb.org/resources/`. An earlier version of this paper was circulated under the title "Language Models and Cognitive Automation for Economic Research."

†Anton Korinek is a Professor at the University of Virginia, David M. Rubenstein Fellow at the Brookings Institution, the Economics of AI Lead at the Centre for the Governance of AI (GovAI), a Research Affiliate of the NBER and CEPR, and a Senior Advisor of the Partnership on AI. Financial support from Brookings and from Longview Philanthropy is gratefully acknowledged. I thank Julian Hazell, Sid Srinivasan, and participants at a number of seminars and webinars for helpful conversations on the topic. Moreover, I thank Max Schnidman, Don Suh, Natasha Swindle, and Davis Taliaferro for excellent research assistance, and GPT-3/4 and Claude/Claude 2 for inspiration and editorial assistance. Moreover, I thank the editor, David Romer, and four anonymous referees for their thoughtful comments.

1 Introduction

Recent advances in generative AI, in particular in the space of large language models (LLMs), have the potential to revolutionize research in economics and other scientific disciplines. Generative AI has crossed the threshold to become useful across a wide range of cognitive tasks. This was illustrated by the viral reception of ChatGPT, which was released by OpenAI in November 2022, gained more than 100m users in its first two months, and was soon estimated to produce a volume of text every 14 days that is equivalent to all of the printed works of humanity (Thompson, 2023). OpenAI and Google DeepMind have since released even more powerful LLMs. Moreover, a growing number of established tech companies and startups have developed their own generative AI systems or adapted them to specific use cases in what some commentators have started to call a 'Cambrian explosion.'

The objective of this article is to describe use cases of modern generative AI to interested economic researchers, based on my exploration of the space.[1] My main emphasis is on LLMs, which are the type of generative AI that is currently most useful for research. I have categorized their use cases into six domains: ideation and feedback, writing, background research, data analysis, coding, and mathematical derivations. I provide general instructions for how to take advantage of each of these capabilities and demonstrate them using specific examples. Moreover, I classify the capabilities of the most commonly used LLMs from experimental to highly useful to provide an overview. Table 2 on page 44 provides a summary at the time of writing, September 2023. I do not put emphasis on other types of generative AI, including image, audio, and video generation tools, as they do not have many use cases in economic research at this time. My hope is that this paper will be a useful guide both for researchers starting to use generative AI and for expert users who are interested in new use cases beyond what they already have experience with to take advantage of the rapidly growing capabilities of LLMs.

The online resources associated with this paper are available at the journal website (see the title footnote on the first page of this article) and will provide regular updates of the capabilities and use cases of the most advanced generative AI tools for economic research. Moreover, they offer a guide on "How do I start?" as well as a page with "Useful Resources on Generative AI for Economists."

At present, I view generative AI to be most helpful as an assistant that can automate small "micro tasks" that researchers engage in numerous times during the day but that are often too small to be assigned to human research assistants. Generative AI tools are useful for such tasks because of their high speed and the low transaction cost. Moreover, they are also useful as tutors in coding and data analysis tasks as well as in ideation and writing. I posit that researchers can significantly increase their productivity by incorporating generative AI into their workflow.

---

[1] Please email me at `akorinek@virginia.edu` to suggest additional use cases and resources that I may incorporate in the online resources.

Closely following the current capabilities of generative AI is also valuable because it foreshadows what future generations of generative AI systems will be able to do. In recent years, the amount of computational power employed in training cutting-edge LLMs has doubled on average every six months, delivering rapid increases in capabilities (Sevilla et al., 2022). There is widespread anticipation that these advances will continue in the near future, and that generative AI systems will continue to become more powerful. It is useful for researchers to familiarize themselves even with experimental capabilities because of the rapid pace of advances. In the longer term, I hypothesize that generative AI may usher in an era of cognitive automation that may have profound implications for scientific progress in economics and other disciplines. Additionally, such cognitive automation may also have stark effects on the value of cognitive labor.

There has been a burst of literature on generative AI in economics in recent months, focused primarily on LLMs. Cowen and Tabarrok (2023) and Mollick and Mollick (2023) describe strategies for how to deploy LLMs for teaching and learning. Dowling and Lucey (2023) show use cases for ChatGPT in finance research ranging from idea generation to data identification. Eloundou et al. (2023) and Felten et al. (2023) analyze how different occupations and industries will be affected by LLMs. Horton (2022) explores the use of LLMs as simulated economic agents. Lopez-Lira and Tang (2023) show that LLMs can be used for sentiment analysis to predict stock price movements. Noy and Zhang (2023) demonstrate in a controlled experiment that ChatGPT made writers 40% faster while improving output quality and helping writers with weaker skills more, thereby reducing output inequality. Peng et al. (2023) show that access to the LLM-based coding assistant GitHub Copilot allows programmers to complete coding tasks by 56% faster.

The reactions to the release of generative AI tools such as ChatGPT have been sharply divided: One camp of commentators label LLMs as nothing but "stochastic parrots" (Bender et al., 2021) or "advanced autocomplete." Another camp equated GPT-4 with the "first sparks of artificial general intelligence" (Bubeck et al., 2023), i.e., artificial intelligence that possesses human-level intelligence across all domains. One of the reasons for such divergent views is that the capabilities or "intelligence" of LLMs are so different from human intelligence, making it hard for humans to relate and to compare. I therefore want to express two warnings before I proceed:

1. It is easy – and dangerous – to overestimate the capabilities of LLMs. They can product text that sounds highly authoritative – even when they "hallucinate," i.e., when the content is completely wrong. In human-written texts, there is a strong correlation between authoritative style and insightful content, but LLMs have learned the former without being reliable on the latter. Users need to watch out not to anthropomorphize LLMs and to exert critical judgment when using the results they generate. In many ways, the capabilities of LLMs feel alien to humans. Their primary objective is to generate text; their creators are still

working on ensuring that the content they generate is consistently truthful and appropriate.

2. It is also easy – and dangerous – to underestimate the capabilities of LLMs. Since they regularly hallucinate and make blatant mistakes, it is easy to dismiss LLMs. However, a former Chair for Mensa International reports that ChatGPT has obtained an IQ score of 147 (99.9th percentile) on a verbal-linguistic IQ test (Thompson, 2023). Moreover, whereas the level of human intelligence is relatively static, LLMs are advancing rapidly, becoming more accurate and powerful with every new iteration.

Ultimately, I believe that the most useful attitude towards generative AI is to heed the lessons of comparative advantage that Ricardo (1817) taught us more than two centuries ago: Generative AI systems increasingly have comparative advantage in generating content; humans have comparative advantage in evaluating and discriminating content (at least for now), as well as in organizing research projects.[2] Moreover, LLMs also have super-human capabilities in processing large amounts of text. All this creates ample space for productive collaboration, as we will explore in the remainder of the paper.

Section 2 describes LLMs, the most useful category of generative AI for economic research, from a technical perspective. It observes that LLMs are deep neural networks that are pre-trained on large amounts of data to create a foundation that is then fine-tuned to follow instructions by human users. LLMs are capable of learning the structure of their training data and forming higher-level abstract representations of concepts. LLMs have been improving according to predictable scaling laws as a function of the amount of computation, parameter count, and size of training data employed, leading to a rapid rise in the capabilities of LLMs. Understanding both the workings and limitations of LLMs is useful to prompt them effectively.

Section 3 describes the most commonly used LLMs at the time of writing and lays out six different areas in which LLMs are useful for research. This is the "living section" of the paper that will be regularly updated in the online resources associated with this paper (see title footnote). In ideation, LLMs can help to brainstorm, evaluate ideas, and provide feedback and counterarguments. In writing, they can synthesize text, provide examples, edit and evaluate text, and generate catchy tweets or titles for a paper. In background research, they are useful for searching and summarizing the literature, translating text, explaining concepts, and formatting references. LLMs are also very capable in coding, writing code based on instructions in natural language, explaining code, translating code between programming languages, and even debugging code. For data analysis, LLMs can create figures, extract data from text, reformat data, classify

---

[2] Agrawal et al. (2018) observed that AI systems had comparative advantage in prediction while humans had comparative advantage in judgement. The generative AI revolution has vastly expanded the capabilities of AI in the years since.

text, extract sentiment, and even simulate humans to generate data. Finally, LLMs are starting to display emergent capabilities in mathematical derivations, starting from setting up models and working through derivations to explaining models. At the end of the section, Table 2 provides a systematic overview of all the described use cases and my rating of their usefulness at the time of writing.

In the final section, I speculate on the medium- and long-run implications of advances in generative AI for research in economics and other disciplines. I hypothesize that in the medium term, AI-based assistants will become increasingly useful for generating more and more of the content that makes up research papers, while human researchers will focus on their comparative advantage, i.e., organizing research projects, prompting, and evaluating generated content. In the long term, we cannot rule out that AI systems may be able to produce and articulate superior economic research by themselves.

2 What Are LLMs?

2.1 Training

Large language models (LLMs) are the type of generative AI that is currently most useful for economic research. They are AI systems trained to predict the next word given preceding text, and typically fine-tuned to follow human instructions and generate responses aligned with human preferences. LLMs are based on deep neural networks with billions or – at the cutting-edge – trillions of parameters. Today's LLMs are built on transformer models, introduced by Vaswani et al. (2017). Transformers introduce an 'attention mechanism' in the processing of text, which endogenously assigns varying degrees of importance to different words, enhancing the model's ability to process complex patterns and dependencies.[3] This mechanism has significantly improved the efficiency of language models and their ability to process the meaning of texts. For example, in the sentence "A currency devaluation can stimulate exports, as it makes goods cheaper," the attention mechanism would associate the word "it" with "currency devaluation," ensuring that the economic concept is correctly represented.

The training of modern LLMs that are used in applications like ChatGPT proceeds in three steps.[4]

The first step, pre-training, occurs via a process called self-supervised learning, which induces the model to represent the conditional probability distribution over words given

---

[3] Earlier language models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, processed textual data sequentially. This characteristic imposed an autoregressive 'forgetting curve', whereby the model's attention to earlier words decreased with the addition of new ones. This led to challenges in capturing long-range dependencies between words, as the length of the processed text expanded, making it more difficult for models to process the meaning contained in sentences and paragraphs.

[4] Wolfram (2023) provides an excellent in-depth description of how LLMs work.

the preceding words, based on its training data.[5] The model is fed text fragments, and the model's parameters are adjusted so that it better predicts what the continuation is. For example, when fed the fragment "the sky is ___" the model learns to predict that likely continuations are "blue" or "cloudy." This process is performed on terabytes of data from Wikipedia, scientific articles, books, and other sources on the internet. To predict the training data in a loss-minimizing way, the model needs to learn syntactic structures, relationships between words and the concepts they represent, the context of sentences and how different words might interact in that context, and how different sentences are related to each other. For example, the system learns that the object in "she loves cats and dogs" refers to animals whereas "it's raining cats and dogs" refers to precipitation. The model also learns any biases contained in its training dataset during pre-training. For example, it may learn from training data that goes back many decades that "doctor" is more likely to refer to men and "nurse" to women.

The second step is instruction fine-tuning, which makes the model better at following human instructions. According to the pre-trained model, a likely continuation of "What's your name?" may be "And how old are you?" But a user who enters "What's your name?" in a chatbot presumably does not want another question but an answer. Instruction fine-tuning makes the model learn how to respond to the user's instructions. It is performed using supervised learning, by feeding the model millions of examples for how to respond to thousands of different instructions for tasks like summarization, question-answering, brainstorming, etc.

The third step is reinforcement learning from human feedback (RLHF), by which feedback from human raters tells the model how different responses compare. This process makes the system better aligned with human preferences, in particular along domains that are difficult to define via instruction fine-tuning. For example, human raters typically prefer responses that are more truthful and polite, and they penalize the system for hateful responses (which may be generated occasionally based on the probability distribution learned in pre-training). However, RLHF is a noisy process. For example, it is part of the reason why LLMs have learned to sound authoritative even when they hallucinate.

From an economic perspective, the pre-training step is by far the most expensive. The user cost of computational resources employed to pre-train models such as GPT-4 is estimated to be in excess of $100m – and this does not include the cost of the human capital employed in developing the model, which is likely of similar magnitude (Knight, 2023). By comparison, once an LLM is trained, the cost of inference, i.e., of using the model to generate output, is quite low, costing fractions of a cent per user query.

---

[5] Technically speaking, LLMs transform text into so-called "tokens" that can represent words, numbers, or punctuation marks before processing them. For ChatGPT, for example, one token encodes on average 4 characters or 3/4 of a word in the English language.

2.2 Scaling and Emergent Capabilities

The performance of LLMs improves predictably according to so-called scaling laws, which are empirical regularities that have held for several generations of machine learning models (Kaplan et al., 2020). These scaling laws observe that the goodness-of-fit of LLMs, or the cross-entropy log-loss, which measures how well the conditional probability distribution of words represented by the model reproduces the probability distribution of words in the training data, improves according to a power law function of the number of parameters of the model and the amount of training data.[6] These relationships allow researchers both to determine how to optimally allocate computational resources (or "compute," to use the terminology employed by AI researchers) and to predict how a given amount of compute translates into the goodness-of-fit of LLMs as measured by the log-loss (Hoffmann et al., 2022). As discussed earlier, the compute devoted to the training runs of top-end models, measured by the number of floating point operations (FLOPs) performed, has doubled on average every six months over the past decade, implying a thousand-fold increase every five years (Sevilla et al., 2022). Given the resources that are currently being invested in the advancement of LLMs, this trend is likely to continue for at least several more years.

As the log-loss of LLMs improves, their ability to produce useful text grows alongside. However, in addition, new capabilities arise at discrete thresholds of the log-loss, in the sense that the capabilities are absent in smaller models, suddenly emerge once a certain threshold is crossed, then improve quickly, and eventually mature. For example, Wei et al. (2022a) report that once a certain threshold of training compute is crossed, LLMs almost predictably develop the ability to perform certain arithmetic computations, to unscramble words, or to perform Q&A. Other significant capabilities that have emerged from language models include the ability to translate, code, and rhyme. In fact, most of the useful capabilities for researchers that we document below have emerged only in recent years. An interesting phenomenon about many of these emergent capabilities is that they initially even surprise the creators of these systems (Ganguli et al., 2022). Some of them are discovered by chance after the systems have been released to the public. This suggests that LLMs may exhibit greater capabilities than what is known, a phenomenon that is termed "capabilities overhang."

Whereas the discussion so far has focused on how well of a representation of their training data LLMs develop, a related philosophical question is whether they display any form of 'understanding' of that data or are merely stochastic 'stochastic parrots' that mimic understanding, as Bender et al. (2021) have argued. Ultimately, this debate is closely related to Turing (1950)'s famous question 'Can machines think?' For the

---

[6] Denoting by L the log-loss of the goodness-of-fit of the training data and by N and D the number of parameters and the amount of data employed to train the model, Hoffmann et al. (2022) find that $L \backsimeq AN^{-0.34} + BD^{-0.28}$ where A and B are constants (see appendix D.2 of their paper). Since the exponents are close to each other, these laws imply that it is optimal to use increases in the amount of "training compute," i.e., the number of computations performed to train the model, to scale the parameter count and the size of the training dataset of LLMs in approximately equal proportions.

purposes of this article, let me observe that LLMs do develop increasingly higher-level abstract representations of concepts and their relationships during pre-training, since having a model of the world makes them better at predicting their training data (Li et al., 2023). These representations of the world are then also applied when LLMs respond to novel questions that do not show up in their training data. The world models of LLMs are becoming increasingly broad, complex, and nuanced with each new generation, making them more and more powerful and useful for researchers and other cognitive workers. For our purposes in this article, it is ultimately irrelevant whether having a world model corresponds to understanding or not.

2.3 Prompts

As with any computational system, the quality of output obtained from an LLM depends on the quality of input. The text that a user enters into an LLM-powered assistant is converted into a vector in a high-dimensional latent space and then processed to generate a sequence of output tokens. A lot of attention has thus been dedicated to how to best prompt LLMs for the information that they are asked to generate. The process of devising good prompts for generative AI systems has come to be called "prompt engineering." It is essentially a form of programming in natural language.

In order to make efficient use of the high-dimensional representation of input data, it is advisable to provide LLMs with context and instructions for style when prompting it to generate content. For example, for many prompts in research-related tasks, it is useful to start with texts like the following: "I am an economist working on a research paper. Provide responses in an academic but engaging style." Some LLM chat interfaces, for example ChatGPT, allow users to set a so-called "system message" like what I just described that is automatically applied to all conversations. More generally, modern LLMs have become better and better at detecting a user's intent – even when the user's initial prompt is insufficient or somewhat unspecific. This means that the importance of "prompt engineering" may have been somewhat overstated.

A useful model for interacting with LLMs is to treat them like an intern who is highly motivated, eager to help, and smart in specific domains, but who has just walked into the job, lacks the context of what you are doing, and is prone to certain types of errors. Based on this, I advise that users provide context, iterate, and be patient in order to obtain the best possible results from LLMs.

2.4 Limitations

There are several important limitation of LLMs that stem from the nature of such models and that users need to be aware of. First, LLMs have a tendency to "hallucinate" or "confabulate," i.e., to produce outputs that are inaccurate. The text they generate is ultimately based on the probability distribution of words that they have learned during pre-training, which may induce LLMs to generate sentences that sound plausible but do not make sense. Dziri et al. (2023) show that LLMs also have difficulties in handling tasks that demand intricate multi-step reasoning. Although hallucination has somewhat declined with newer generations of LLMs, it is an inherent characteristic of LLMs, making human oversight of their outputs crucial.

Second, LLMs may also generate privacy concerns, which come in two forms. They are trained on data that may contain sensitive or private information, leading them to generate outputs that may compromise privacy. Moreover, depending on the model and settings employed, the providers of LLMs may retain the data inputted by users for future training purposes, compromising user privacy. However, at the time of writing, there are commercially available LLMs that safeguard the data that is entered, allowing for use in applications for which safety and confidentiality is critical.

Third, LLMs learn the stereotypes and biases present in their training data. When they generate new text, they build on the probability distribution of text in their training data and may replicate and perpetuate biased, harmful, prejudiced, or inappropriate language. Although the providers of LLMs typically attempt to counteract such biases, it is crucial for users to exert appropriate oversight in this domain.

Finally, LLMs are subject to significant data limitations. The pre-training process uses data up to a certain point in time, and the knowledge of the LLM cuts off at that point. The architecture of modern LLMs does not allow them to learn in real time, although some LLMs can access the internet for up-to-date information and generate responses based on what they learn on the web.

Despite these limitations, LLMs exhibit such quick response times and low transaction cost that they are useful for automating a wide range of micro tasks in which they are still error-prone and in which similarly capable human research assistants would not be competitive. For example, I would not resort to human research assistance for micro-tasks such as spelling out the first-order conditions of an optimization problem while I am writing a paper – the associated delay would be too large. But the instantaneous response of LLMs makes it useful to outsource this task, even if there are occasional mistakes. Similarly, I would not hire a human research assistant who regularly commits basic logical fallacies while presenting results with great confidence – I would consider them too unreliable. But after a short adjustment period, I have found it useful to incorporate LLMs that do precisely that into my workflow.

3 Applications of Generative AI in Economic Research

This section demonstrates use cases of cutting-edge LLMs in economic research, classified along six domains: ideation and feedback, writing, background research, coding, data analysis, and mathematical derivations. For each domain, I provide a general description and a few specific use cases for how to take advantage of LLM capabilities. I illustrate both the capabilities and failures of the LLMs at the time of writing to provide a balanced version of the usefulness of LLMs.

Product	Model	Company	Released	Tokens	Data Cutoff	URL
ChatGPT	GPT-3.5	OpenAI	11/2022	4k	9/2021	chat.openai.com
New Bing	GPT-4	Microsoft	1/2023	4k	web access	bing.com/chat
ChatGPT+	GPT-4	OpenAI	1/2023	4k	9/2021	chat.openai.com
Bard	PaLM-2	Google	3/2023	2k	web access	bard.google.com
Claude	Claude 2	Anthropic	7/2023	100k	early 2023	claude.ai
LlaMA	LlaMA 2	Meta	7/2023	4k	early 2023	open source

Table 1: Overview of commonly used LLMs, specifications as of September 2023

Since the landscape of generative AI described in this section is evolving so fast, the journal website `https://www.aeaweb.org/articles?id=10.1257/jel.20231736` and the website `https://www.aeaweb.org/resources/` will provide regular updates of this section.

3.1 Overview of Commonly Used LLMs

Table 1 provides an overview of commonly used LLMs as of September 2023, together with some of their key properties and limitations, including their release date, the maximum token limit that they can process, and the date as of which the training data cut off. It also lists the URLs at which chatbots powered by these LLMs can be accessed.

OpenAI's ChatGPT is by far the most popular LLM. It comes in a free version that is based on OpenAI's GPT-3.5 model as well as a paid version for $20/month. Since March 2023, the paid version has offered access to GPT-4, which is currently the most powerful LLM that is publicly available. Both GPT-3.5 and GPT-4 were pre-trained on data that cut off in Sept. 2021 so they have no knowledge of more recent events. They have a context window of 4000 tokens, amounting to about 3000 words in English, with the limit applying to the sum of the user prompt and the completion that is generated. Aside from the ChatGPT web interface, OpenAI also offers access to its models using an Application Programming Interface (API) that enables programmers to query a range of different OpenAI LLMs while setting several model parameters that affect the result. The models are available on a pay-per-use-basis and come in different sizes, with smaller models executing fast and cheaply whereas larger models are more powerful but slower and more expensive. The API also offers access to a version of GPT-4 with a context window of 32k tokens.

Microsoft's New Bing chat engine is also based on GPT-3.5/4 and can browse the web in real time, serving users results that are based on the most recent information available on the internet. It also allows users to follow the links to the sources that it has identified. It allows users to choose from 3 modes, Precise, Balanced, or Creative, the latter of which provides users with free access to GPT-4.

Google's Bard is based on its PaLM-2 Bison model as of June 2023, which offers functionality at a similar level to GPT3.5. Like Bing, it can also search the web to include real-time information in its response to user queries and allows users to follow links to its sources. It allows users to pick from multiple answers and makes it easy for users to export the results into spreadsheets. Like OpenAI, Google also offers API access to a range of PaLM-2 models of different sizes and capabilities, although it excludes its most powerful PaLM-2 Unicorn model from public access.

Anthropic's Claude 2 is an LLM that brands itself as being helpful, honest, and harmless. It employs a process called constitutional AI to train the LLM to follow a set of high-level ethical principles (Bai et al., 2022). One of the highlights of Claude is that it has a context window of 100k token, meaning that it can process about 75,000 words at once. This is far beyond the other models and implies that Claude can process most academic papers in one go, as we will explore further below. Unfortunately Claude 2 is currently only available in the US and UK. Anthropic also allows API access to their underlying models to process LLM requests in bulk.

Meta's LlaMA 2 series is a set of models with 7B, 14B and 70B parameters released in July 2023 as well as a code-generation model named Code LlaMA released in August 2023. Meta has freely distributed the underlying code and the weights of the trained models while withholding the data used to train the model. The most powerful 70B parameter version is on par with GPT-3.5 and is available on the leading cloud computing platforms, including Microsoft Azure, AWS, and Hugging Face. LlaMA 2 comes with a license that allows both researchers and (with minor limitations) corporations to run the LLMs on their own computers and to fine-tune and improve the pre-trained models. This is highly beneficial from an economic perspective, as it distributes the social surplus created by LLMs and stimulates innovation. However, as these moels become more powerful, it also poses growing safety risks (Anderljung et al., 2023). For 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).

A website that provides occasional users with a user-friendly interface with access to all leading LLMs is `https://poe.com`.

Plugins The capabilities of base LLM can be significantly enhanced with plugins that allow the LLM to perform additional tasks that LLMs by themselves are not good at. For economists, the plugin that is perhaps most noteworthy at the time of writing is ChatGPT's Advanced Data Analysis, which is available to ChatGPT Plus subscribers. The plugin allows ChatGPT to write and execute computer code in a sandboxed environment and to display the results as well as to build and iterate on them. Advanced Data Analysis also allows users to upload files and perform data processing tasks on them, ranging from complex analysis like regressions to file conversions. We will cover several of these capabilities below. Google Bard also runs code in the background to perform certain mathematical tasks.

Another ChatGPT plugin that is useful for economists is Wolfram's Alpha, which can be activated in the plugin store that is available to ChatGPT Plus subscribers. The site `https://www.wolfram.com/wolfram-plugin-chatgpt/` describes a range of examples for how to use this plugin.

.

Vision-Language Models (VLMs) combine LLMs with the ability to process visual information and integrate the two. A version of GPT-4, which is not publicly available at the time of writing, can incorporate visual information in its prompts. Bard can display images from Google Search in its responses. This is an area with a lot of potential for future use cases. For example, early demonstrations suggest that VLMs are able to produce complex outputs based on hand-drawn back-of-the-envelope drafts.

Reproducibility Most of the applications in the remainder of this section use the leading publicly available LLM at the time of writing, 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 the results by calling OpenAI's API. The code sets the parameter "Temperature" to zero, which makes the LLM responses close to deterministic. For non-programmers, a user-friendly way to replicate the results is the OpenAI web interface `https://platform.openai.com/playground`, in which "Temperature" can also be set to zero. Both the OpenAI API and the Playground require a paid subscription to access GPT-4.[7]

There are two factors that limit the reproducibility of my results. First, OpenAI states that "setting temperature to 0 will make the outputs mostly deterministic, but a small amount of variability will remain." I have observed these limits to reproducibility in particular for examples with responses that span multiple sentences.[8] Second, OpenAI states that "as we launch safer and more capable models, we regularly retire older models." Moreover, "after a new version is launched, older versions will typically be deprecated 3 months later." If the `gpt4-0613` model is retired, my results may no longer be reproducible.[9]

---

[7] Executing all of the examples labeled GPT3.5/GPT-4 below required a bit over 5k of input and 5k of output tokens each. At the time of writing, the total cost was slightly below 50 cents. Further pricing information is available at `https://openai.com/pricing`.

[8] See `https://platform.openai.com/docs/guides/gpt/why-are-model-outputs-inconsistent` for further information on the inconsistency of model output, even 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.

[9] Moreover, see `https://platform.openai.com/docs/deprecations` on OpenAI's policy of

The most convenient user interface is ChatGPT, available at `https://chat.openai.com/`, which employs a “Temperature” parameter greater than zero, which introduces more variation into the model’s responses. Accessing GPT-4 via this interface requires a paid subscription to ChatGPT Plus. This allows users to try out the spirit of all the examples employing GPT-4 below, but the extra variability implies that the exact results will differ every time a prompt is executed. The same applies to ChatGPT Advanced Data Analysis and the Wolfram plugin, which both rely on ChatGPT, and to Claude 2, which offers the ability to upload files. My reproduction code therefore exlcudes the results of the latter three models.

3.2 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 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.

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. Whereas any researcher who uses LLMs for ideation and feedback will naturally be careful about which points they use and which points they reject in any given use case – just as we do when we discuss ideas with colleagues – there may 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 GPT-4 for brainstorming or feedback, it is important to keep in mind that its training data cuts off in Fall 2021.

---

model deprecations as well as the current timelines for how long existing models are guaranteed to remain available.

3.2.1 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. Throughout the remainder of this section, I will present all examples generated by LLMs in boxes, with the prompt in bold in the header and the LLM's generated response in the body of the box. 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 purposes here:

Chat 1 (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?

3.2.2 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 2 (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.

Speaking from experience, I very much agree with this assessment.

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 into Claude 2 and asked it the following:

Chat 3 (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 overrelying 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. To provide an example, Online Appendix A.1 provides the referee report that Claude 2 drafted for the Feb. 2023 working paper version of this paper. The LLM offered several comments that resemble comments made by the actual (presumably human) referees of the paper and writes that a revised version of this paper would be a good fit for the *Journal of Economic Literature*. However, interestingly, the comments in the draft referee report are considerably shallower than the comments in Chat 3, which directly prompted Claude 2 for ways to improve the paper.

To check whether this simply reflects a positivity bias, I also asked Claude 2 whether the paper would be suitable for the *American Economic Review*. As shown in the online appendix, it argued (correctly) 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." 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.

3.2.3 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 4 (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.

3.3 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) and that suggests corrections and stylistic improvements as the user is typing is grammarly.com.

3.3.1 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 5 (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.

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. [10] However, 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

---

[10] This 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.

- in a style that is comprehensible to non-economists
- in the style of [enter your favorite economist, politician, etc.][11]

LLMs can also write text in LaTeX format, which automatically surrounds variables by the $ signs required for LaTeX, 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.

3.3.2 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:

Chat 6 (GPT-4): Can you correct the following sentence?
These paper describe six domain in which large language models (LLMs) displays emergant capbilities 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 6 (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.

---

[11] When 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...”

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.”

3.3.3 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 7 (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 7 (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?

3.3.4 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 8 (GPT-4): Brainstorm two or three catchy titles for a paper with the following abstract:
[Abstract from Chat 7]

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 8 (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.

3.3.5 Generating tweets to promote a paper

After having written a paper, the next step is to promote it. The following exercise is similar to the previous one but converts the abstract I pasted into chunks that would work well for Twitter or similar social networks:

Chat 9 (GPT-4): Write three or four tweets that summarize the following text. Use simple sentences and make them catchy:
[Abstract from Chat 7]

1. "Generative AI, especially large language models like ChatGPT, are set to revolutionize research! They're becoming handy as research assistants and tutors across six domains. #AIRevolution #ChatGPT"
2. "From ideation to coding, generative AI is proving useful in a variety of tasks. Economists, it's time to automate your micro tasks and boost productivity! #AIProductivity #Economics"
3. "The future of AI-powered cognitive automation in economic research looks promising. Stay tuned for regular updates on the latest capabilities of generative AI. #FutureOfAI #EconomicResearch"
4. "Want to get started with generative AI? Check out the online resources associated with my paper for instructions and updates. #GenerativeAI #AIResources"

Upon request, LLMs can also incorporate suitable emoticons or links.

3.4 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.

3.4.1 Summarizing Text

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 10 (GPT-4): Summarize the following text in one sentence:
[Abstract from Chat 7]

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. As discussed above, GPT-3.5/GPT-4 can process only up to 4000 tokens or about 3000 words, limiting the ability to produce summaries of longer texts.

Paper Summaries By contrast, the LLM Claude 2 can process texts with up to 100,000 tokens or about 75,000 words at the time of writing, which covers the length of almost all 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, 2023) into Claude 2 and asked for a short summary – "short" in order to not take up too much space here:

Chat 11 (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 2 implies that it can keep the entire text of a typical paper in its working memory and that it 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.

3.4.2 Literature Research

At the time of writing, LLMs are of limited usefulness for searching and referencing the academic literature. When asked for citations or references, they frequently hallucinate and generate authoritative-sounding papers that do not actually exist. Although this has recently improved for cutting-edge LLMs – GPT-4 is significantly better in this respect than GPT-3.5 and is often able to provide standard references that are frequently cited in the literature – all references that LLMs generate should be carefully double-checked. The following is an example:

Chat 12 (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 that has close to 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 and Shapoval (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.

The reason why LLMs struggle with academic references is that they do not 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.

However, there are alternative tools available that employ LLMs to help researchers find academic references. For example, 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. Moreover, 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.

3.4.3 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 13 (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 Research},
author={Korinek, Anton},
year={2023},
institution={NBER},
type={Working Paper},
number={30957},
month={February}
}
```

Chat 13 (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 for all but the most highly cited papers.

3.4.4 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 14 (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 teko aaly, 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.

3.4.5 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, 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 15 (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 is also accessible via the free version of ChatGPT:

Chat 16 (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.

3.5 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. In fact, coding may be one of the areas where current LLMs lead to the greatest productivity gains: Peng et al. (2023) report a controlled experiment in which programmers using OpenAI-powered GitHub Copilot completed their assignments on average 55.8% faster, amounting to a 126% productivity increase.

In the OpenAI ecosystem, Advanced Data Analysis has made the use of ChatGPT Plus for coding tasks in the programming language python even more 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 offers seamless integration into Visual Studio and several other integrated development environments for programmers, and works for many different programming and markup languages.[12] Meta's Code LlaMA, released in August 2023, offers powerful coding assistance on an open-source basis.

3.5.1 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. Because of their length, the results are given in the online appendix.

Chat 17 (ChatGPT Advanced Data Analysis): Simulate the Solow growth model and plot the results in a four chart.

(See Figure 1 for the four chart and Online Appendix A.2 for the code and description generated.)

The model performed the simulation task as requested. As detailed in Online Appendix A.2, 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 hallucinated the description.) 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.

---

[12] While 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.

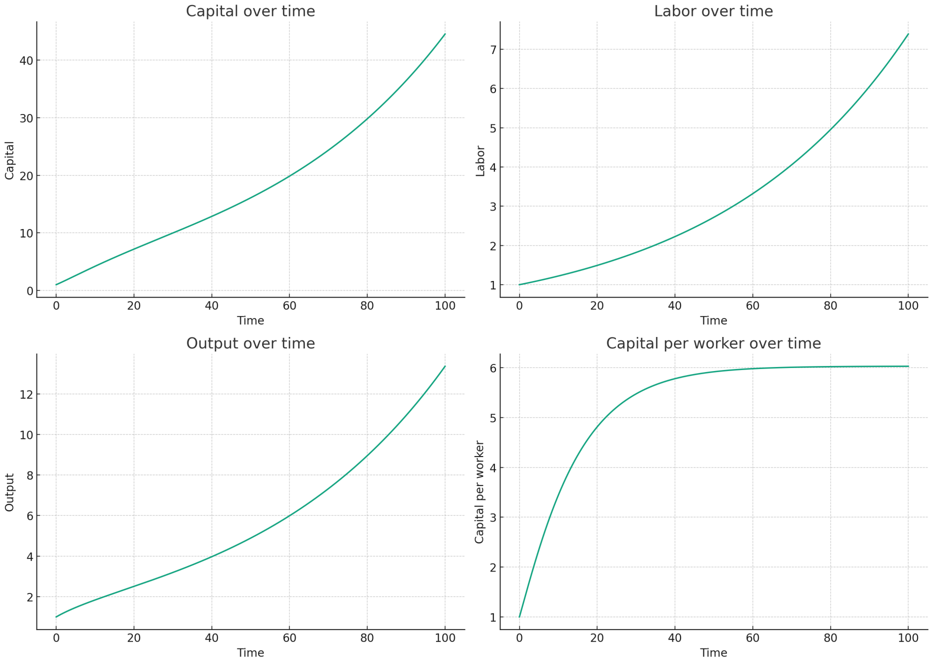


Figure 1: Simulation of Solow growth model (Source: ChatGPT Advanced Data Analysis)

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.

3.5.2 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 18 (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.

3.5.3 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 directly on the conversation in the preceding Section 3.5.2, provides an example:

Chat 18 (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 ‘.ˆ‘, not ‘**‘.

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

3.5.4 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 19 (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 of Section 3.5.2 with an error introduced, illustrates how LLMs can help to catch mistakes in code such as typos:

Chat 20 (GPT-4): Can you debug the following function?
[Code from Chat 18, 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.

3.6 Data Analysis

LLMs can 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.

3.6.1 Creating figures

One of the most recent and perhaps most useful functions of ChatGPT for economists is Advanced Data Analysis, which builds on the coding capabilities of GPT-4 to create versatile figures and graphs. 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.[13] In the following, I show my own prompts as well as the final figures displayed by ChatGPT

---

[13] To compile the underlying stock market data, I asked ChatGPT to write a script to download the data, as described in Online Appendix A.3.

Advanced Data Analysis. A full list of the output produced by Advanced Data Analysis is given in Online Appendix A.3.

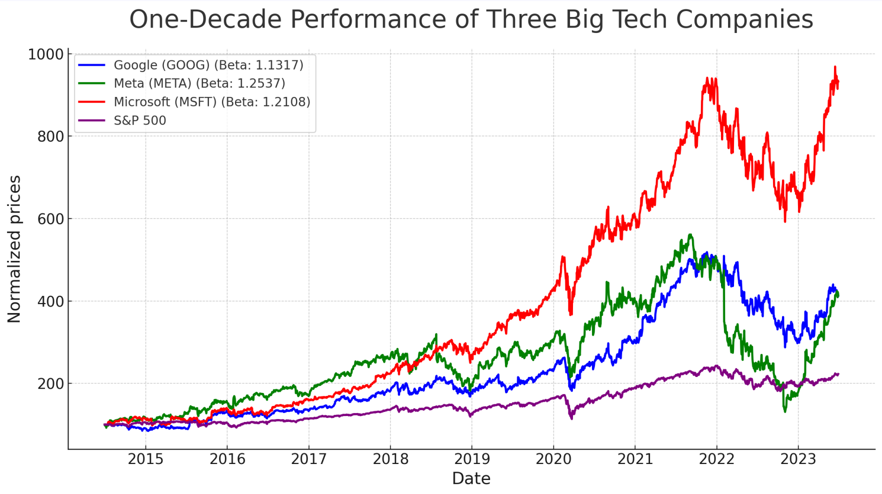
Chat 21 (ChatGPT Advanced Data Analysis): Can you display a graph that shows the performance of these securities normalized to start at 100? **[Uploaded stock_data.csv]**

[first version of figure displayed]

Chat 21 (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".

[second version of figure displayed]

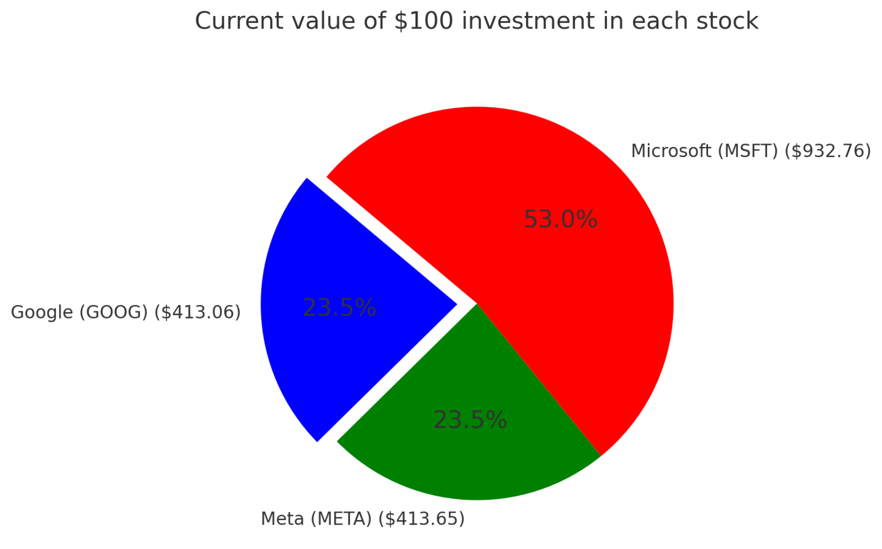
Chat 21 (ChatGPT Advanced Data Analysis) ctd.: Double the size of the title, legend, axes labels and ticks. Label "^GSPC" in the legend as "S&P 500".



Chat 21 (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?

[first version of figure displayed]

Chat 21 (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?



Advanced Data Analysis can also display more general figures.

3.6.2 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 22 (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

Notice that the LLM correctly processes the meaning of the expression "both in econ and math" for Sally and extract the grades accordingly.

Other applications 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. Dunn et al. (2022) 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.

3.6.3 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 22 (GPT-4) ctd.: Reformat in CSV format.

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

Chat 22 (GPT-4) ctd.: Write as a LaTeX table.

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

It excelled at both these and other similar tasks.

3.6.4 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.[14] Following the principle of chain-of-thought prompting suggested by Wei et al. (2022b): 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.

---

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

Chat 23 (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 random circumstances, the system'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.

3.6.5 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 24 (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.[15] 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.

3.6.6 Simulating human subjects

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, using simulated test subjects to replicate and extend upon several behavioral experiments. The following example illustrates the concept:

---

[15] See, for example, `https://www.cnbc.com/2023/02/01/live-updates-fed-rate-hike-february.html`

Chat 25 (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.

3.7 Mathematical Derivations

The ability of general-purpose LLMs to perform mathematical derivations are still limited. 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. (2023) 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 noticable 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. (2023) 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.

3.7.1 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 26 (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:

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

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.

3.7.2 Deriving equations

At the time of writing, 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 26 (GPT-4) ctd.: Solve the optimization problem.

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

$$\mathcal{L}(x_1, x_2, \lambda) = (\alpha x_1^{\rho} + (1-\alpha) x_2^{\rho})^{\frac{1}{\rho}} + \lambda(M - p_1 x_1 - p_2 x_2)$$

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:

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

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 26 (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 $M = p_1 x_1 + p_2 x_2$ to get:

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

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.

3.7.3 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 27 (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?

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

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

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

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.

In this example, 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.

3.8 Summary

Table 2 summarizes all the sample tasks illustrated in this paper, categorized by the six described domains of application of LLMs. In the third column of the table, I report my subjective rating of how useful I found the described LLM capabilities as of September 2023. My ratings come in three categories, where an empty circle (○) describes capabilities that I currently consider more experimental and that deliver inconsistent results, requiring significant human oversight; a half-full circle (◐) 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 (●) 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.

Category	**Task**	**Usefulness**
Ideation and Feedback	Brainstorming	●
	Feedback	◐
	Providing counterarguments	◐
Writing	Synthesizing text	●
	Editing text	●
	Evaluating text	●
	Generating catchy titles & headlines	●
	Generating tweets to promote a paper	●
Background Research	Summarizing Text	●
	Literature Research	○
	Formatting References	●
	Translating Text	●
	Explaining Concepts	◐
Coding	Writing code	◐
	Explaining code	◐
	Translating code	●
	Debugging code	◐
Data Analysis	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	◐

The third column reports my subjective rating of LLM capabilities as of September 2023:

○: 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

Table 2: Summary of LLM capabilities and rating of usefulness

I will provide regular updates of this summary table – together with the remainder of Section 3 – in the online resources associated with this paper (see title footnote) so as to offer an up-to-date assessment of the usefulness of generative AI for economic research.

4 Outlook and Concluding Thoughts

LLMs have become useful research tools for tasks ranging from ideation and feedback, writing and background research to data analysis, coding, and mathematical derivations. Cognitive automation via LLMs is already making researchers significantly more productive. I expect that a growing number of researchers will incorporate LLMs into their workflows. This is likely to help to increase the overall speed of progress in economics, although it risks leaving behind those who do not take advantage of LLMs or do not have access, creating a new digital divide.[16]

In the medium term, I anticipate that LLM-based assistants and tutors will become increasingly useful for generating the content that makes up research papers. Human researchers will focus on their comparative advantage – by posing the questions, suggesting directions for obtaining answers, discriminating which parts of the produced content are useful, editing, and providing feedback, akin to an advisor. Moreover, they will also continue to play an important role in organizing research efforts – for example, by coordinating teams and procuring data sources, akin to a research manager.

Over time, further advances will imply that LLMs are performing their tasks better and better so that the need for humans to provide inputs, edits, and feedback will diminish. There will be a growing range of research activities for which we simply rubber-stamp the output produced by ever-more advanced LLMs. Eventually, our AI research assistants may graduate and become researchers of their own. The experience will be deflating.

It is difficult to predict whether and how different areas of research will be differentially affected by cognitive automation – for example, will theorists be the last ones standing because their abilities prove difficult to replicate by LLMs, or will a more advanced LLM fine-tuned for mathematical applications outperform humans and automate theory work more quickly than other branches of economics? Will empiricists have a leg up because the process of collecting novel data involves steps that are difficult to automate?

For the longer term, I sympathize with the beliefs of many of the pioneers of AI, including the godfather of deep learning, Geoffrey Hinton, and the founders of both OpenAI and DeepMind, Sam Altman and Demis Hasabis, that AI will ultimately reach artificial general intelligence and surpass human intelligence across all domains.

---

[16] To the extent that longer and more complex papers are the result of a positional arms race among researchers without commensurate improvements in insight, greater productivity in generating text may also lead to further bloating of research papers without improving depth or quality (see, e.g. Frank, 1991).

If their premise is correct, it would imply that AI will ultimately also be better at solving scientific problems, including the problems we encounter in economics.

This poses the question of how to best harness AI to solve the economic problems that plague humanity, including the problems that are too difficult to solve for humans, or humans alone. Demis Hasabis famously founded DeepMind with the goal of "solving intelligence, and then using that to solve everything else." Yet how can we train AI systems to master a discipline such as economics, which is sometimes regarded as much an art as a science? Max Planck famously declared that he was originally interested in studying economics but found it too difficult, presumably referring to the way in which economists use simple regularities that are fuzzy and imperfect but still insightful to distill the complex behaviors of economic subjects and systems. Economists have long used computers to process the mathematical laws we use to describe economies. And as we explored in this paper, recent advances in LLMs have gotten us closer to the point where AI systems can deal with economic concepts and ideas that capture the art of economics. Yet combining the two in a productive manner still seems elusive at the time of writing. If our objective is to solve the hardest problems in economics, how much should we invest in developing AI systems that can do that, and is now the point to start doing so?

Richard Sutton (2019) suggests that the ultimate "Bitter Lesson" from progress in AI is that general approaches that leverage advances in computing power ultimately trump domain-specific expertise. He observes that for most of the history of AI, researchers worked on making their AI systems smarter and more powerful by programming domain-specific knowledge into them – for example, teaching a chess computer the wisdom accumulated by generations of chess players. He observed that this strategy always helped in the short term, but the benefits of it eventually plateaued. In the long term, Sutton suggests that general approaches that take advantage of brute scaling of computing power are always the more successful strategy – for example, when DeepMind developed AlphaZero, a chess computer that used massive computing power to learn chess by itself without any human input, it learned the game well enough within 24 hours to beat all other chess computers in the world – and of course all humans (Silver et al., 2017).

A similar bitter lesson may apply to economics. In our work as economists, we spend a lot of our time and effort on similar strategies to what Sutton describes, honing our domain-specific knowledge and expending tremendous resources to solve economic problems. Yet sufficiently advanced general AI systems may be able to produce and articulate superior economic models, and the cognitive work of human economists – like that of all other researchers – may eventually become redundant.[17]

Garry Kasparov (2017) distills the lessons he learned from observing decades of progress in chess computers, with important milestones including his 1997 defeat to Deep Blue

---

[17] Today's cutting-edge AI systems are just a few generations from reaching our best estimates of the complexity of human brains, making it perhaps unsurprising that they are starting to exhibit eerily similar capabilities in a growing range of areas (Carlsmith, 2020).

and the 2017 release of AlphaZero, as follows (p. 254-255):

> "Thousands of years of status quo human dominance, a few decades of weak competition, a few years of struggle for supremacy. Then, game over. For the rest of human history, [...] machines will be better than humans at chess. The competition period is a tiny dot on the historical timeline. This is the unavoidable one-way street of technological progress in everything from the cotton gin to manufacturing robots to intelligent agents.
>
> The competition dot gets all the attention because we feel it intensely when it occurs during our lifetimes. The struggle phase often has a direct impact on our lives in real time, so we over-inflate its relevance in the big picture. [...] it is almost always better to start looking for alternatives and how to advance the change into something better instead of trying to fight it and hold on to the dying status quo."

In Kasparov's terminology, LLMs have entered the period of "weak competition" with cognitive workers, including economic researchers. We are currently at the competition dot, and LLMs are garnering a lot of attention. Yet just like the chess champions of the 1990s, we should not let our anthropocentric bias blind us to the rise of AI, and we should remind ourselves that the competition period, which we may feel intensely in coming years, may just be a tiny dot on the historical timeline.

Whereas my long-term predictions are clearly speculative, I am quite confident about my predictions on the short- and medium-term implications of LLMs. I also believe that the cognitive automation ushered in by the rapid rise of LLMs poses important and urgent new research questions to economists, of which I will brainstorm a few:

1. What will cognitive automation imply for labor markets? Will it also accelerate the development of robots that automate physical tasks? How can our society best prepare for the impending changes?

2. What are the implications of cognitive automation for education? Will human capital be devalued? Will humanity become, on average, less intelligent, just like we became, on average, less strong after the mechanization of agriculture?

3. How can we ensure that relying on a small set of LLMs does not make our thinking more homogeneous and less creative?

4. How will cognitive automation affect technological progress and economic growth? If human labor can be automated, what will be the bottlenecks to growth in the future?

5. ...

6. Finally, but perhaps most importantly, how can we best address the AI alignment problem, i.e., ensure that ever-more advanced and potentially super-intelligent AI systems pursue objectives that are aligned with human objectives?

Continuing on the last question, economists have the tools to translate concepts from the social sciences and humanities, such as "human objectives," into analytic concepts like preferences that are more easily accessible to machines. And we have experience analyzing agency and control problems and their solutions. Their contribution is urgently needed. In fact, there are two channels through which economists can make important contributions to this line of work: First, we can directly work on AI alignment; see, e.g., Korinek and Balwit (2023), for some tentative research directions. Second, our work will affect the concepts and representations through which future AI systems will view economic questions and, ultimately, through which they will view the world – just like our work influences that of our human students, whether they work as economists or policymakers, it will also influence future LLMs that perform economic research and that impact economic policy. As Keynes (1936) described so powerfully at the conclusion of his general theory,

> "... the ideas of economists and political philosophers, both when they are right and when they are wrong, are more powerful than is commonly understood. Indeed the world is ruled by little else. [...] I am sure that the power of vested interests is vastly exaggerated compared with the gradual encroachment of ideas. [...] soon or late, it is ideas, not vested interests, which are dangerous for good or evil."

At this point, human researchers, especially when AI-assisted, are still the best technology around for generating economic research!

References

Agrawal, A., Gans, J., and Goldfarb, A. (2018). *Prediction Machines: The Simple Economics of Artificial Intelligence.* Harvard Business Review Press.

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.*

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., and others (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv:2212.08073.*

Bender, E. M., Gebru, T., McMillan-Major, A., and Mitchell, M. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., and others (2021). On the opportunities and risks of foundation models. *arXiv:2108.07258.*

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. and Shapoval, O. (2023). GPT-3.5 hallucinates nonexistent citations: Evidence from economics. *SSRN Working Paper.*

Carlsmith, J. (2020). How much computational power does it take to match the human brain? Technical report.

Charness, G., Jabarian, B., and List, J. A. (2023). Generation next: Experimentation with ai. *Working Paper, University of Chicago.*

Cowen, T. and Tabarrok, A. T. (2023). How to learn and teach economics with large language models, including GPT. GMU Working Paper in Economics No. 23-18.

Dowling, M. and Lucey, B. (2023). ChatGPT for (finance) research: The Bananarama conjecture. *Finance Research Letters*, page 103662.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., and Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv:2212.05238.*

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., West, P., Bhagavatula, C., Bras, R. L., Hwang, J. D., Sanyal, S., Welleck, S., Ren, X., Ettinger, A., Harchaoui, Z., and Choi, Y. (2023). Faith and fate: Limits of transformers on compositionality.

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.

Felten, E. W., Raj, M., and Seamans, R. (2023). How will language modelers like ChatGPT affect occupations and industries? SSRN Working Paper.

Frank, R. H. (1991). Positional externalities. In Zeckhauser, R., editor, *Strategy and Choice*, pages 25–47. MIT Press, Cambridge, MA.

Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P. C., Chevalier, A., and Berner, J. (2023). Mathematical capabilities of ChatGPT. *arXiv:2301.13867*.

Ganguli, D., Hernandez, D., Lovitt, L., Askell, A., Bai, Y., and others (2022). Predictability and surprise in large generative models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. ACM.

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.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. (2022). Training compute-optimal large language models. *arXiv:2203.15556*.

Horton, J. J. (2022). Large language models as simulated economic agents: What can we learn from homo silicus? 31122. NBER Working Paper.

Jiao, W., Wang, W., Huang, J.-t., Wang, X., and Tu, Z. (2023). Is ChatGPT a good translator? A preliminary study. *arXiv:2301.08745*.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv:2001.08361*.

Kasparov, G. (2017). *Deep thinking: where machine intelligence ends and human creativity begins*. PublicAffairs.

Keynes, J. M. (1936). *The General Theory of Employment, Interest and Money*. Palgrave Macmillan.

Knight, W. (2023). Openai's ceo says the age of giant ai models is already over. *Wired*.

Korinek, A. (2023). Language models and cognitive automation for economic research. *NBER Working Paper*, 30957.

Korinek, A. and Balwit, A. (2023). Aligned with whom? Direct and social goals for AI systems. In Bullock, J. and others, editors, *Oxford Handbook of AI Governance*. Oxford University Press.

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., and Wattenberg, M. (2023). Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv:2210.13382*.

Lopez-Lira, A. and Tang, Y. (2023). Can ChatGPT forecast stock price movements? return predictability and large language models.

Mollick, E. R. and Mollick, L. (2023). Using AI to implement effective teaching strategies in classrooms: Five strategies, including prompts. SSRN Working Paper.

Noorbakhsh, K., Sulaiman, M., Sharifi, M., Roy, K., and Jamshidi, P. (2021). Pretrained language models are symbolic mathematics solvers too! arXiv:2110.03501.

Noy, S. and Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654).

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot.

Ricardo, D. (1817). *On the Principles of Political Economy and Taxation*. John Murray, London.

Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., and Villalobos, P. (2022). Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv:1712.01815*.

Sutton, R. (2019). The bitter lesson. Incomplete Ideas (blog). http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

Thompson, A. D. (2023). GPT-3.5 + ChatGPT: An illustrated overview. Technical report, LifeArchitect.ai.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, LIX(236):433–460.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022a). Emergent abilities of large language models. *arXiv:2206.07682*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *arXiv:2201.11903*.

Wolfram, S. (2023). What is ChatGPT doing ... and why does it work?

Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. *arXiv:2307.15043*.

A Online Appendix: Additional Examples

A.1 Claude 2's Referee Report on This Paper

The following reproduces Claude 2's editorial opinion on the first working paper version of this paper (Korinek, 2023):

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]

It is well known that Claude 2 is programmed to be friendly and upbeat. To evaluate whether it would provide an equally positive assessment for other journals, I continued with the following prompt:

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.

A.2 Simulating the Solow Model

The following prompt uses ChatGPT Advanced Data Analysis to simulate the Solow model:

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?

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)
```

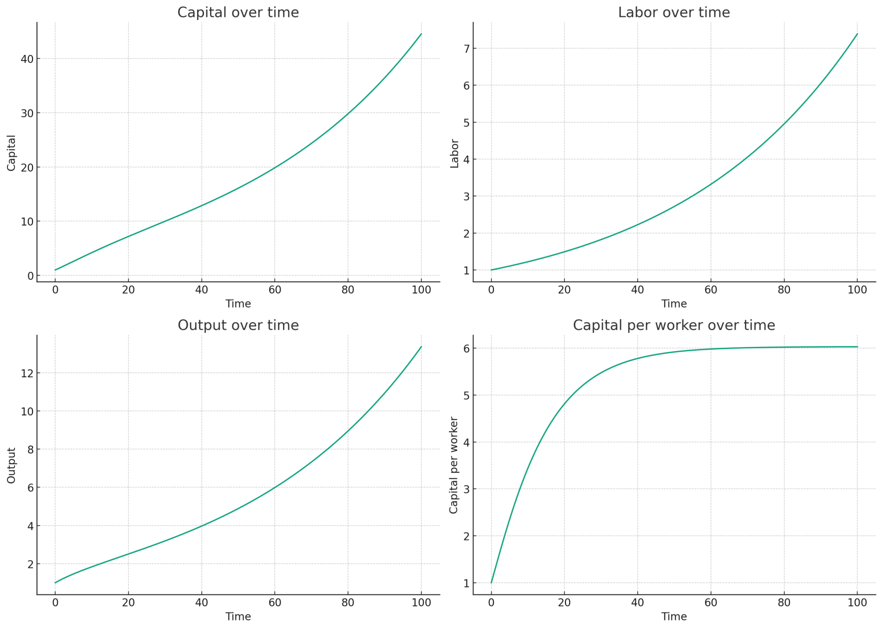
```
axs[0, 1].set_title(’Labor over time’)
axs[0, 1].set_xlabel(’Time’)
axs[0, 1].set_ylabel(’Labor’)

# 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.

A.3 Creating figures

Since Advanced Data Analysis is not connected to the Internet, I started by asking GPT-4 to write code that downloads stock price data and saves it in a CSV file:

Write code to download adjusted close of GOOG, META and MSFT as well as the S&P500 index from 7/1/2014 to 6/30/2023 and save it in a CSV file.

To download the adjusted close of GOOG, META (previously FB), and MSFT as well as the S&P500 index, we can use the ‘yfinance‘ library in Python. This library allows us to download stock data from Yahoo Finance.

Please note that the end date you provided is in the future (6/30/2023), so the code will download data up to the current date.

Here is the Python code:

```
import yfinance as yf
import pandas as pd

# Define the ticker list
tickers_list = [’GOOG’, ’META’, ’MSFT’, ’^GSPC’]

# Fetch the data
data = yf.download(tickers_list, start="2014-07-01", end="2023-06-30")[’Adj
Close’]

# Print first 5 rows of the data
print(data.head())

# Save data to a CSV file
data.to_csv(’stock_data.csv’)
```

This code will download the adjusted close prices for the specified stocks and the S&P500 index from July 1, 2014, to June 30, 2023, and save it in a CSV file named ’stock_data.csv’.

Please note that you need to install the ‘yfinance‘ library before running this code. You can install it using pip:

```
pip install yfinance
```

Also, please note that the ticker symbol for Facebook has changed to ’META’ due to the company’s recent rebranding. If you want to download data for Facebook before the name change, you should use ’FB’ instead of ’META’.

I executed the code in python on my local computer.[18] Then I uploaded this CSV file in ChatGPT Advanced Data Analysis and prompted it as follows:

Can you display a graph that shows the performance of these securities normalized to start at 100?
[Uploaded stock_data.csv]

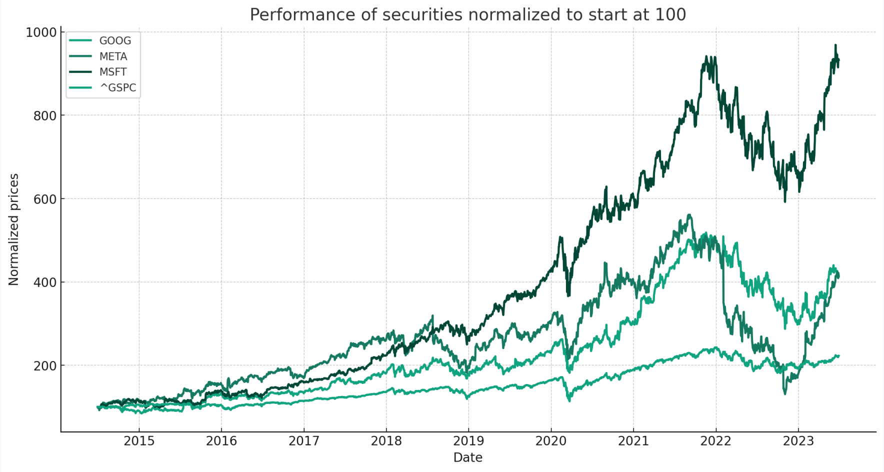
[Code generated and executed]
The dataset contains the closing prices for the securities Google (GOOG), Meta (META), Microsoft (MSFT), and the S&P 500 index (^GSPC) over a certain period of time.

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]

---

[18] ChatGPT Plus can also provide very helpful advice on questions such as “How do I install python on my computer?”



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

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]

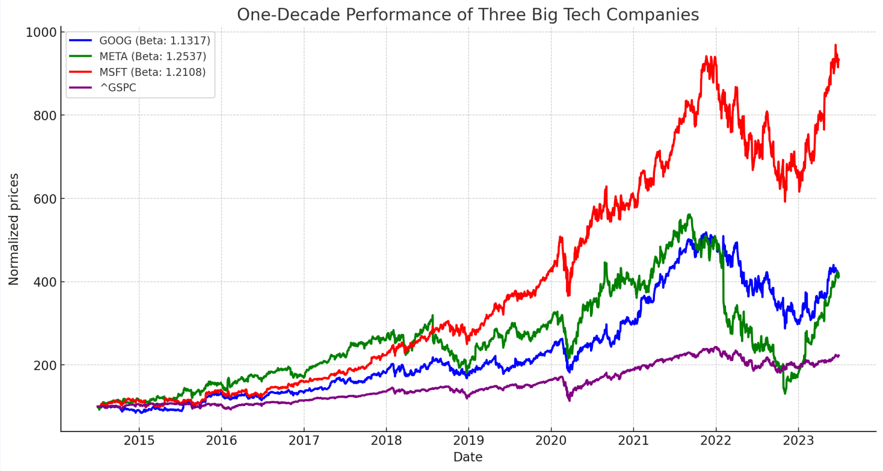
Here are the beta values for the stocks:

- 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]



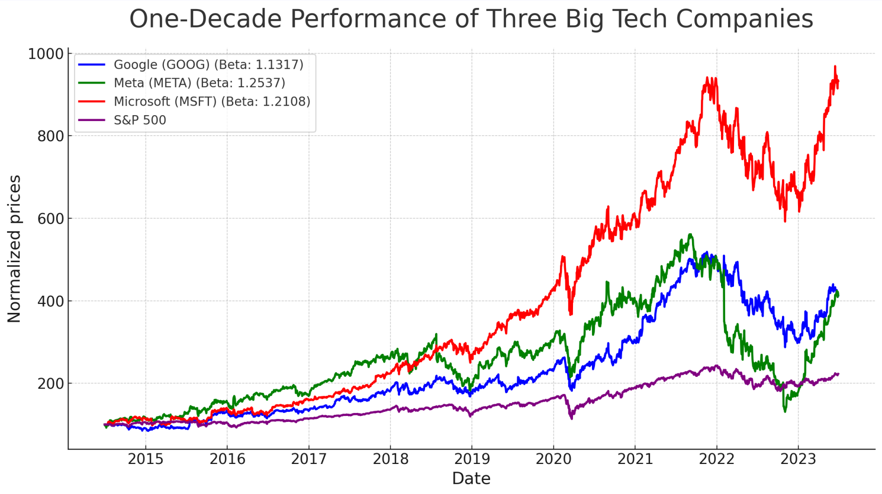
Here is the updated graph with the new title, colors, and beta values included in the legend.

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.

Double the size of the title, legend, axes labels and ticks. Label "^GSPC" in the legend as "S&P 500".

[Code generated and executed]



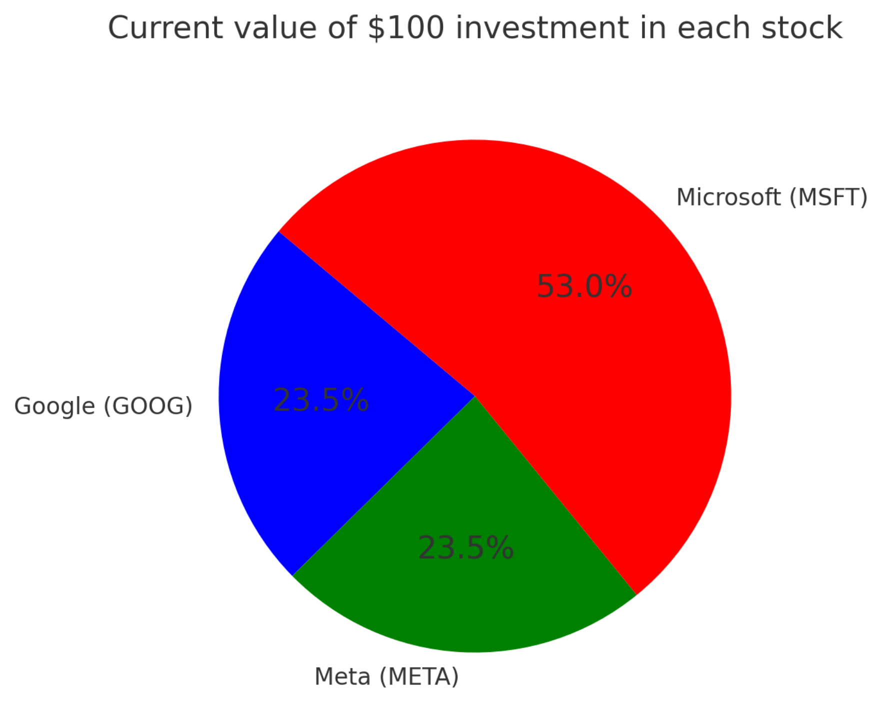
Here is the updated graph with the requested changes:

- The title, legend, axes labels, and ticks have all been doubled in size.
- The "^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.

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]

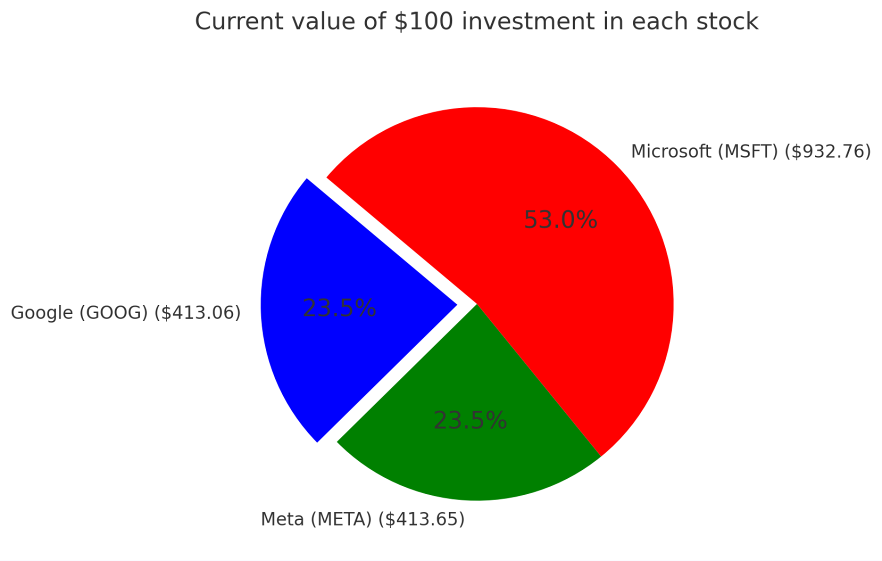


Here is a pie chart that shows the current value of a $100 investment made in each of the three big tech stocks (Google, Meta, and Microsoft) at the start of the period represented in the data.

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.

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.)