

LLMs Level Up—Better, Faster, Cheaper: June 2024 Update to Section 3 of “Generative AI for Economic Research: Use Cases and Implications for Economists,” Published in the *Journal of Economic Literature* 61(4)[†]

ANTON KORINEK

Applications of Generative AI in Economic Research

The landscape of generative artificial intelligence (AI) has evolved rapidly since the publication of the original version of this article in the *Journal of Economic Literature* (Korinek 2023b). This section summarizes the main innovations and demonstrates updated use cases of cutting-edge large language models (LLMs) in economic research, classified along the same six domains as the original article: 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.

[†]Please cite this as Korinek, Anton. 2024. “LLMs Level Up—Better, Faster, Cheaper: June 2024 Update to Section 3 of ‘Generative AI for Economic Research: Use Cases and Implications for Economists,’ published in the *Journal of Economic Literature* 61(4).” <https://www.aeaweb.org/articles?id=10.1257/jel.20231736#additionalMaterials>.

Summary of New Developments—Recent progress in LLMs has been characterized by better performance, growing context windows so LLMs can process more data at once, better recall, faster processing, and falling costs—all music to the ears of economists. Since the first version of this article was published in December 2023, the three leading frontier AI labs have each released significant updates to their LLM offerings incorporating, among other features, vision capabilities and real-time sound processing. At the time of writing in May 2024, OpenAI’s GPT-4o is the most capable publicly available LLM, and the one I use the most. OpenAI also offers a GPT store inside its ChatGPT app that offers useful customized versions of GPT-4. Moreover, I regularly use Anthropic’s Claude 3, released in March 2024, for its writing capabilities. Google DeepMind released Gemini Pro 1.5 in January 2024, which currently offers the largest context window—the publicly available version can process up to 1 million tokens (about 1500 pages of text) simultaneously. Its sister model, Gemini Ultra 1.0, released in January 2024, is Google DeepMind’s most capable model but is not publicly accessible.

TABLE 1 OVERVIEW OF COMMONLY USED LLMs, SPECIFICATIONS AS OF MAY 15, 2024

AI Lab	Model	Best Version	Released	Tokens	Data cutoff	URL
OpenAI	GPT-4o	GPT-4o	May 2024	128k	Dec 2023	chatgpt.com*
Anthropic	Claude 3	Opus	Mar 2024	200k	Aug 2023	claude.ai
Google	Gemini	Ultra	Feb 2024	32k	Jul 2023	gemini.google.com*
Mistral	Mixtral	8x22b	Apr 2024	65k	Sep 2021	OS, chat.mistral.ai
Meta	LLaMA 3	Llama-3	Apr 2024	8k	Dec 2023	OS, meta.ai

Note: A * denotes chatbots that can also browse the internet; OS denotes open-source models.

The performance of the frontier models of all three labs is similar, although OpenAI has a slight lead.

Whereas the described frontier models are proprietary and only available via the internet, a growing number of smaller (albeit less capable) models have been released open source, allowing users to employ them on their own servers for security- or privacy-sensitive applications. Some of these models are small enough that they can be run on laptops and even cell phones.

In addition to describing these new developments in section 3.1, I also cover several novel examples and use cases for LLMs in the subsections below:

- Automated prompt generation
- Feedback on research via voice chats
- Converting hand-written equations to LaTeX
- Generating presentation slides
- Summarizing YouTube videos
- Outlining a plan for a coding project
- Finding data sources

In economics, most AEA journals will soon require authors to declare whether and how they have employed LLMs in their research. It goes without saying that authors are solely accountable for the content they submit.

Data Confidentiality.—An important issue for researchers is how to ensure the confiden-

tiality of the data that they enter into LLMs. OpenAI offers a “temporary chat” option in its ChatGPT app as well as a privacy option in the user settings (turn off “Improve the model for everyone”) to let users opt out of having their inputs used for training future LLMs. OpenAI does not employ user data that are entered via its application programming interface (API) for training purposes. Anthropic does not employ user data for future training except with an explicit opt in or, in rare circumstances, if it is flagged for safety review. Google advises users against entering confidential information into its Gemini apps, since input data may be used for future training purposes.

3.1 Overview of Cutting-Edge LLMs

Table 1 provides an overview of commonly used LLMs as of May 2024, together with some of their key properties and limitations, including their release date, context window, and training data cutoff. The context window reflects the maximum number of tokens that the model can process, with the limit typically applying to the sum of the user prompt and the completion that is generated. On average, four tokens correspond to three words of text in the English language. The training data cutoff reflects how recent the data is on which the LLM was pretrained. Models generally do not have knowledge of facts that occurred past this date, except if they have the capacity to access the internet.

3.1.1 Modes of Access

The listed LLMs can be accessed in four different ways:

- **Web-based Chatbots:** The models in table 1 are all accessible as chatbots under the URLs listed in the last column. These chatbot interfaces allow users to prompt the models as assistants or tutors. Most of the use cases documented below illustrate this mode of interaction. Note that most chatbots are programmed to introduce a certain amount of random variation in their responses, so that results in chat interfaces are not perfectly reproducible. The free version of these chatbots comes either with usage restrictions or provides access only to less powerful model versions. In my experience, this makes it worthwhile to pay the \$20 monthly subscription fee required to access one or several of the top three models with fewer restrictions—ChatGPT 4o, Claude 3 Opus, or Gemini Advanced—to avoid frustration and to derive meaningful productivity gains from LLM use in day-to-day work.
- **Real-time Voice Assistants:** OpenAI offers an interactive voice assistant that allows for natural interaction with its latest model, GPT-4o, in spoken language and via camera and/or screenshots, available via its ChatGPT phone app and on iOS. Some users report that they keep the ChatGPT app open on their phone in voice mode throughout certain work tasks, or even throughout the day, so that they can easily draw on the their digital assistant at any point without raising a finger. Google has released a demo of its Project Astra (**advanced seeing and talking responsive agent**) assistant with similar functionality that will be publicly released later in 2024.
- **Web-based Experimentation Platforms:** All major LLM providers also offer web-based interfaces that offer greater functionality and flexibility than chatbots but do not require programming knowledge. These platforms, such as OpenAI Playground (<https://playground.openai.com>), Anthropic Console (<https://console.anthropic.com>), and Google AI Studio (<https://aistudio.google.com>), allow users to experiment with different model settings, like temperature and top-p sampling, and provide more control over the input and output formats compared to chatbots. Such experimentation platforms are particularly useful for exploring the capabilities of LLMs, testing prompts, and fine-tuning models for specific tasks.
- **APIs:** For the maximum level of customization and integration, the listed models are also accessible through APIs, which allow programmers to integrate LLMs directly into their own software applications. This enables a wide range of more advanced and customized use cases, such as automating repetitive tasks or analyzing large datasets using natural language processing techniques. APIs provide more flexibility and control compared to above two options and can be employed on a pay-per-use basis, but they also require a higher level of technical expertise to use effectively. Accessing LLMs through APIs typically involves signing up for an API key from the model provider, installing a client library in the programming language of choice, and writing code to interact with the API endpoints. While this process may be more complex than using a chatbot, it unlocks the full potential of LLMs for those with the necessary programming skills. The replication package for this paper demonstrates how to use APIs to automatically query LLMs.

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

3.1.2 *Leading Proprietary Models*

The first three models in the table are proprietary, which means that they are owned by the labs that developed them, and that their source code, architecture, and model weights are not freely available for public use, modification, or distribution. They can only be accessed on the labs’ servers subject to the conditions and controls that they stipulate.

All of the leading models combine language processing with the ability to process visual information, implying that they are vision-language models (VLMs). This allows users to upload images in the chat interface or APIs and ask for responses based on these images, as we explore in some of the use cases in the ensuing sections. We will nonetheless continue to refer to them using the common term LLMs.

OpenAI’s GPT-4 was first released in March 2023 as the leading model in its class and is by far the most popular LLM. The most recent version is GPT-4o, released in May 2024, where the suffix “o” stands for “omni” and reflects that the model can simultaneously process text, images, and sound to act as a real-time digital assistant. GPT-4o also has access to several extensions that enhance the capabilities of the model, including data

analysis, the ability to browse the web, and the GPT Store described below. GPT-4o is available for free but subject to usage limits in ChatGPT that depend in part on demand. Paying users can obtain more reliable access. GPT-3.5 Turbo is a smaller sibling of GPT-4 and is also accessible via the free version of ChatGPT. Alternatively, OpenAI’s models are all accessible through the web-based OpenAI Playground as well as via API.

A feature of ChatGPT 4o that is particularly useful for economists is Advanced Data Analysis, a tool that can write and execute Python code in a sandboxed environment directly within the ChatGPT interface and that can display the results and build and iterate on them. Advanced Data Analysis allows users to upload files and perform data processing tasks on them, ranging from file conversions to complex analysis like regressions. We will cover several of these capabilities below, for example, in our simulation of the Solow model.

OpenAI’s GPT Store is a digital marketplace integrated within the ChatGPT interface, offering users the ability to browse, install, and use a wide range of extensions and applications that are developed by third-party creators. These “custom GPTs” enhance GPT-4o’s functionality by adding specialized knowledge, interactive tools, and customized experiences tailored to individual user needs, ranging from writing to economic analysis tools. (They replace the ChatGPT plug-ins that used to be available until March 2024.) To use a custom GPT, it can be located by clicking on “Explore GPTs” in the menu bar on the left side of the ChatGPT interface and searching for it. A few custom GPTs that economists may find useful include the following:

- **Data Analyst** is a custom GPT built by OpenAI that allows users to upload

spreadsheets and analyze or visualize the data contained.

- **Wolfram** provides access to computation, math, curated knowledge, and real-time data from Wolfram Alpha, the maker of Mathematica.
- **Consensus** offers an AI-based research assistant that searches 200 million academic papers to offer science-based answers with citations to the underlying articles.

After you have used a custom GPT for the first time, you can call upon it in any ChatGPT conversation by including it with the @-sign, for example, by writing “@Wolfram.” Users can also make their own custom GPTs by clicking “Explore GPTs” and the “Create” button, and going through the ensuing process step by step. The resulting custom GPTs can be either used privately or shared publicly.

Microsoft Copilot (formerly Bing Chat) is Microsoft’s chat engine based on the GPT-3.5/4 family of OpenAI models. Microsoft has integrated Copilot into all of its office products. Copilot can also browse the web, serving users results that are based on the most recent information available on the internet. It allows users to choose from 3 modes, Precise, Balanced, or Creative.

Anthropic’s Claude 3 was the best publicly available LLM at its release in March 2024 but has now been eclipsed by GPT-4o. Claude brands itself as being helpful, honest, and harmless, employing a process called constitutional AI to train the LLM to follow a set of high-level ethical principles (Bai et al. 2022). Claude comes in three versions: *Claude 3 Opus* is the most powerful model, which I regularly use for writing since I like its writing style; *Claude 3 Sonnet* offers a compromise between speed and skills; *Claude 3 Haiku* is the smallest

and fastest model, which is useful for bulk research applications that do not require the most sophisticated model available. Claude 3 has a context window of 200k tokens, which makes it able to process about 150,000 words in one go—for example, the user can upload and simultaneously process several academic papers in the Claude chat interface. The models are also available via the web-based Anthropic Console and via API, which also offers users the ability to allow the language model to draw on external tools.

Google DeepMind’s Gemini series of LLMs was first released in December 2023 and is accessible via an eponymous chatbot that can also access the internet to include real-time information in its responses and allows users to cross-check results and follow links to its sources. The Gemini family of models comes in four sizes. *Gemini Ultra 1.0* is the most powerful model but is not publicly available. *Gemini Pro 1.5* was released in February 2024 and carries the distinction of having a 1m token context window in its publicly available version—the longest of all publicly available LLMs. This offers new use cases—for example, it allows researchers to upload a significant body of their work all at once and process queries based on it. *Gemini Flash 1.5* was released in May 2024 and offers faster speeds at lower cost but slightly lower performance. Both *Pro 1.5* and *Flash 1.5* are available via the web-based Google AI studio platform as well as via API. Gemini Nano 1.0 is small enough to operate on cell phones.

3.1.3 Leading Open-Source Models

The last two LLMs listed in table 1 are open source, which means that they are freely available to use, modify, and distribute.¹ This

¹Note that many publicly released LLMs do not entirely satisfy the traditional strict definition of the term “open source,” which would imply that the source code, training data, and resulting parameter weights would be

offers several benefits for economic research. Firstly, the transparency of open-source models allows researchers to examine the underlying architecture, enabling them to better understand the model's structure and judge the generated outputs. Secondly, open-source projects allow for collaborative innovation and may accelerate the development of LLMs tailored to specific needs, such as economic research. Researchers can contribute their expertise, suggest improvements, and adapt the models to their particular needs. Thirdly, if researchers have access to low-cost computing resources, they can leverage open-source models for their work without incurring financial costs. Finally, open-source models allow for greater reproducibility, which is helpful for ensuring scientific integrity in economic research, as it enables other researchers to verify and build upon the reported results. These benefits make open-source language models an attractive choice for economic researchers seeking to harness the power of natural language processing in their work.

From an economic perspective open-source models are highly beneficial, as they freely distribute the economic social surplus created by LLMs and stimulate innovation. On the downside, as open-source LLMs become more capable, they also pose 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). Seger et al. (2023) discuss the pros and cons of open sourcing LLMs as well as intermediate solutions that may be desirable as such

models become more capable and pose growing safety risks.

Mistral's Mixtral 8x22b model is one of the leading open-source models, released in April 2024. Mistral is a Paris-based startup founded by former Google DeepMind and Meta employees. The name of the model is a play-on-words on the company's name as well as the model's underlying mixture-of-experts (MoE) architecture, which combines eight specialized sub-models, called experts, to solve complex tasks efficiently.² Mistral also offers a suite of proprietary models and smaller open-source models as well as a chat interface, a web-based platform, and API access to their models.

Meta's LLaMA 3 series encompasses a family of models with 8B, 70B, and 400B parameters, the first two of which were released in April 2024. The largest model is expected to be released shortly. Meta has freely distributed the underlying code and the weights of the trained models while withholding the data used to train the model. The 70B parameter version outperforms GPT-3.5 and is available on the leading cloud computing platforms, including Microsoft Azure, AWS, and Hugging Face.

Locally Operating LLMs.—The emergence of open-source models has also allowed researchers to run LLMs on their own computers, offering advantages such as

freely shared without restrictions. For example, the models listed in the table do not share their training source code and data. Moreover, some of them, for example Meta's LLaMA 3, impose limits on the large-scale commercial use of their models. The AI lab TogetherAI has a project named RedPajama to reproduce and distribute an open source version of the LLaMA dataset.

²In contrast to traditional dense LLMs (for example, GPT-3), where all parameters are used for every token generated, MoE models dynamically select different subsets of parameters for each input, allowing for more efficient computation and better scaling to larger model sizes. This approach is more difficult to implement but enables MoE models to achieve state-of-the-art performance, while being more computationally efficient compared to their dense counterparts. Although the three leading proprietary labs (Anthropic, Google DeepMind, OpenAI) are silent on the exact architecture of their leading models, experts believe that they all use MoE architectures.

data privacy, cost-effectiveness, customization, and offline accessibility. The computational resource requirements imply that only small models can be executed at a reasonable speed on most desktop computers. However, advances in computational capacity as well as rapid efficiency gains of LLMs that allow greater capabilities of smaller models are rapidly making the local use of LLMs more attractive. There are currently two solutions that make it easy to deploy LLMs locally:

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

The open-source LLM Gemma 2b was developed by Google DeepMind for deployment on cell phones. Microsoft’s phi-3-mini 3.8b model is also optimized for use in local machines.

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

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

- <https://docs.anthropic.com/claude/docs/intro-to-prompting> and an associated prompt library at <https://docs.anthropic.com/en/prompt-library/library>
- <https://platform.openai.com/docs/guides/prompt-engineering>
- https://ai.google.dev/docs/prompt_intro
- <https://llama.meta.com/get-started/#prompting>

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

Chat 1 (Anthropic Prompt Generator): *I have written an economics paper and would like to create a draft of a blog for Brookings.*

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

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

The target audience for this blog post is:

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

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

Next, write a short outline for the blog post in a <outline> section. Keep the target audience in

mind as you decide what to focus on and how to present the information.

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

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

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

The generated prompt provides a remarkable level of detail that employs several cutting-edge prompting techniques to obtain the best possible results: it asks the model to start by taking notes in a “scratchpad” section, which induces the model to synthesize the main lessons of the text. Then it asks the model to prepare an outline, which induces the model to generate a plan. Only after these steps does the prompt ask the model to write a draft of the text. The prompt also leaves two locations for the users to insert the paper in question and to define the audience. I simply substituted these by writing “The full text of the paper is attached” and inserting “policymakers” as the target audience. The prompt generated useful results on all three of the leading LLMs that would save considerable time in such a project, although the generated blog post was not quite up to the depth and quality of a think tank like Brookings.³

³I asked Anthropic Prompt Generator to create prompts for blog posts for several different outlets. For some reason, asking for a blog targeted at Brookings gave rise to a higher-quality prompt than asking for any other

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

3.1.4 Reproducibility

Most of the applications in the remainder of this article use the leading publicly available LLM at the time the December 2023 *JEL* version of the article was written, OpenAI’s GPT-4, version gpt4-0613. In the online materials associated with this article (see the footnote on the front page 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. Nonprogrammers can replicate the results on the web-based experimentation platform <https://platform.openai.com/playground>. Both the OpenAI API and the Playground platform require a paid subscription to access GPT-4.⁴

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

outlet I attempted. My best explanation is that the model internalizes that Brookings is known for its high-quality analysis. (Disclaimer: the author is a nonresident fellow at Brookings.)

⁴Executing all of the examples labeled GPT3.5/GPT-4 below required a bit over 5k of input and 5k of output tokens each. In October 2023, the total cost was slightly below 50 cents, but this cost has declined substantially since. Using GPT-4o, the same token count costs about 10 cents. Up-to-date pricing information for OpenAI’s models is available at <https://openai.com/pricing>.

span multiple sentences.⁵ 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.⁶

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, which is part of ChatGPT, and to Claude 2/3, which offers the ability to upload files. My reproduction code therefore excludes the results of the latter models.

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

⁵ 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 calculates 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.

⁶ Moreover, see <https://platform.openai.com/docs/deprecations> on OpenAI’s policy of model deprecations as well as the current timelines for how long existing models are guaranteed to remain available.

3.2 Ideation and Feedback

Research starts with the process of ideation, that is, 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, Terwiesch, and Ulrich (2010) observes that creative performance depends on (i) the quantity of ideas, (ii) the average quality of ideas, and (iii) the variance that 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. Any researcher who uses LLMs for ideation and feedback must, of course, be careful about which points they use and which points they reject in a given use case—just as we do when we discuss ideas with colleagues. There may also be subtle downsides that materialize over time. The reliance on LLM-generated ideas may make individual researchers rely more on automation and practice less critical thinking of their own. Moreover, if more and more economists rely on the same one or two cutting-edge LLMs to generate ideas and obtain feedback, there is a risk that the

ideas that economists work on will become more and more homogeneous and include fewer truly novel ideas. This risk of homogenization is also discussed in Bommasani et al. (2021). Moreover, when using LLMs for brainstorming or feedback, it is important to keep in mind when their training data cuts off (see table 1).

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, so LLMs are best suited for brainstorming in areas in which one is not an expert.

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

Chat 2 (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 five 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 3 (GPT-4): *What are the top three pros and cons of working on the economics of AGI?*

Pros of Working on the Economics of AGI:

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

Cons of Working on the Economics of AGI:

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

it difficult to test hypotheses or draw definitive conclusions.

Another example of a useful prompt for eliciting feedback is:

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

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

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

Chat 4 (Claude 2): *Can you suggest ways to improve this paper?* [Attachment: w30957.pdf]

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

- Provide more specific details and examples when describing the different use cases. For some of them, the explanations are a bit vague. Adding more concrete illustrations would strengthen those sections.
- Expand the discussion of the limitations and potential downsides of over-relying on LLMs. The paper focuses mostly on the benefits, but more balance discussing the risks would be helpful.
- Strengthen the analysis of the long-term implications. The outlook section is thought-provoking but remains speculative.

Adding more rigorous analysis to back up the predictions could make it more convincing.

- Consider testing the use cases on multiple LLMs and comparing their capabilities. The examples all seem to use a single LLM, but testing different ones could reveal variation in strengths/weaknesses.
- Provide guidance on how to properly evaluate LLM outputs before using them. Since LLMs can make mistakes, some tips on double-checking the results would be useful.
- (additional suggestions omitted for space reasons)

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

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

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

- What are the main strengths and weaknesses of this paper?
- What are the main novel ideas in the paper that are not sufficiently emphasized?
- Can you identify any instances of bias in this paper?
- How could I improve section [insert number]?
- Can you draft a referee report for this paper for the *Journal of Economic Literature*?

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

Chat 5 (Claude 2): *Can you draft a referee report for this paper for the Journal of Economic Literature?* [Attachment: w30957.pdf]

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

To the editors of the *Journal of Economic Literature*:

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

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

Summary:

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

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

Comments:

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

I have only minor recommendations to further strengthen the paper:

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

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

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

Sincerely,
[Your name here]

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

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

Chat 5 (Claude 2) ctd.: *Would this paper be suitable for the American Economic Review?*

Based on my review, I do not think this paper would be suitable for publication in the *American*

Economic Review [AER] in its current form. Here is my assessment:

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

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

Some factors to consider regarding *AER* suitability:

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

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

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

Claude 2 argued that the paper would not be suitable since “it is more expository and demonstrative rather than presenting novel research or analysis.” It continued: “As such, it may not rise to the level of contributing major new economic insights demanded

by AER.” This (correct) assessment suggests that the model already possesses the rudimentary capacity to judge what makes for suitable research papers in different journals.

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

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

3.2.3 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 6 (GPT-4): *I am working on an economic*

model that predicts that cognitive automation will increase inequality. What are the top five 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 counterarguments 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 to changing the style of text, editing text, evaluating styles, and 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 earlier 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 plug-ins for many

common applications (including web browsers and office packages) and that suggests corrections and stylistic improvements as the user is typing is grammarly.com.

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

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

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

us to prepare for this eventuality. This could involve developing a deeper understanding of these models to leverage their capabilities effectively, diversifying our skill sets to include areas not easily automated, and advocating for policies that ensure a just transition for those potentially displaced by these technological advancements.

Utilizing this allows researchers to concentrate their energy on the ideas in their text as opposed to the mundane writing process. The text is generally high-quality, although some claim that it is a bit sterile and may lack the idiosyncrasies and elements of surprise that characterize human writing.⁷

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

- in a colloquial style
- in a style that is comprehensible to noneconomists
- in the style of [enter your favorite economist, politician, etc.]⁸

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

Writing is of course a skill that is not only useful for research—cognitive workers in many areas report using, for example, ChatGPT to draft emails, memos, performance reviews, or summaries based on a few bullet points. Likewise, professors report

⁷This 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.

⁸When 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 . . .”

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 8 (GPT-4): *Can you correct the following sentence? These paper describe six domain in which large language models (LLMs) displays emergant capabilities that are useful to the economic research.*

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

Chat 8 (GPT-4) ctd.: *Can you explain the main corrections you made?*

Sure, here are the main corrections:

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

As illustrated in the follow-up question, the LLM can also largely explain its edits so that writers can learn from any corrections it has

made. This latter capability has improved significantly going from GPT-3.5 to GPT-4. Notice, however, that it left out a seventh correction from the list—it also corrected a typo in the word “capabilities.” This highlights the need to carefully check the results of LLMs.

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

3.3.3 *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 9 (GPT-4): *Give a concise list of the top five 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 9 (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 Converting equations to LaTeX [All New in June 2024 Version]

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

Chat 10 (GPT-4): *Convert the attached image to LaTeX code.*

The image shows a handwritten mathematical equation on a light blue background. The equation is $Y = F(K, L) = A[K^\alpha + L^\beta]^\rho, \rho < 0$. The variables K and L are in italics, and the exponents α and β are also in italics. The Greek letter ρ is also in italics.

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

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

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

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

3.3.5 Generating Titles and 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 11 (GPT-4): *Brainstorm two or three catchy titles for a paper with the following abstract: [Abstract from chat 9 inserted]*

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

Chat 11 (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.

3.3.6 Generating X Posts 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 X (formerly Twitter) or similar social networks:

Chat 12 (GPT-4): *Write three or four [posts] that summarize the following text. Use simple sentences and make them catchy: [Abstract from chat 9 inserted].*

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.

3.3.7 Generating Presentation Slides [All New in June 2024 Version]

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

Chat 13 (Claude 3): *Can you create a 20-slide deck for the attached economic research paper? [Attachment: w30957.pdf]*

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

Slide 1: Title Slide

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

Slide 2: Introduction

- Large language models (LLMs) can revolutionize economic research

- Describes 25 use cases of LLMs across 6 domains
- Provides instructions and examples for each use case
- Classifies usefulness of capabilities from experimental to highly useful

Slide 3: What are LLMs?

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

[Slides 4–18 omitted for space reasons]

Slide 19: Importance of Economic Ideas

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

Slide 20: Conclusion

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

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

3.4 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 Summarization [Updated in June 2024 Version]

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 14 (GPT-4): Summarize the following text in one sentence: [Abstract of this paper from chat 9 inserted].

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

The summary sentence it provided covers all the important bases.

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

Chat 15 (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.

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

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

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

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

Chat 16 (Gemini): Provide a one-paragraph summary of this: <https://www.youtube.com/watch?v=sjL6Gl6ZIqs>.

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

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

3.4.2 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 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 17 (GPT-4): Provide citation information for Anton Korinek’s paper with Zach Bethune.

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

The paper title is correct but the publication information is hallucinated—and this is for a paper with over 200 citations. For small variations of my prompt (e.g., putting the authors in alphabetical 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 percent 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, LLMs with web connections such as Bing or Bard can be prompted to look up specific papers on the internet and generate reference information from their findings. Moreover, the website <https://elicit.org> allows searches of the academic literature and displays a list of papers (that really exist) based on simple user prompts. It also includes a brief summary generated by an LLM.

3.4.3 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 18 (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 18 (GPT-4) ctd.: Format it in Chicago style.

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

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

3.4.4 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 19 (GPT-4): Translate the following sentence into Spanish, German, and Finnish: *Generative AI, in particular large language models (LLMs) such as ChatGPT, has the potential to revolutionize research in many disciplines.*

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

German: Generative KI, insbesondere grosse Sprachmodelle (LLMs) wie ChatGPT, hat das

Potenzial, die Forschung in vielen Disziplinen zu revolutionieren.

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

During pretraining, 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 materials, and even for more advanced researchers who are branching out beyond their main area of expertise. For good explanations, it is critical to employ the most cutting-edge LLMs.

Chat 20 (GPT-4): *What are instrumental variables useful for?*

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

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

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

When employing LLMs in this way, follow-up questions that go into further detail can be very useful and allow the user to obtain personalized tutoring. Other concept questions that the LLM answered well included the following:

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

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

Chat 21 (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 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 provide 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 percent faster, amounting to a 126 percent productivity increase.

On ChatGPT Plus, Advanced Data Analysis has made coding tasks in Python very convenient, as it can write code, execute it, learn from it, write follow-up code, and display the requested results. It also allows users to upload code, ask for specific modifications, and download it again to continue work on it. GitHub Copilot and Tabnine

offer seamless integration into Visual Studio and several other integrated development environments for programmers, and they work for many different programming and markup languages.⁹ 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. The model occasionally asks the user for confirmation and feedback. The following reports my exchange:

Chat 22 (ChatGPT Advanced Data Analysis):
Simulate the Solow growth model and plot the results in a four chart.

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

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

where:

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

Assumptions for the model:

⁹While 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.

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

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

- $s = 0.4$,
- $\delta = 0.1$,
- $n = 0.02$,
- $\alpha = 0.33$,
- initial capital stock $K(0) = 1$,
- initial labor $L(0) = 1$.

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

Then, we will plot the following four charts:

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

Does that sound good?

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

Chat 22 (ChatGPT Advanced Data Analysis) ctd.: *Go ahead.*

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

# Model parameters
s = 0.4
delta = 0.1
n = 0.02
alpha = 0.33
KO = 1
LO = 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] = KO
L[0] = LO
Y[0] = KO**alpha * LO**(1-alpha)

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

# Capital per worker
K_L = K / L

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

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

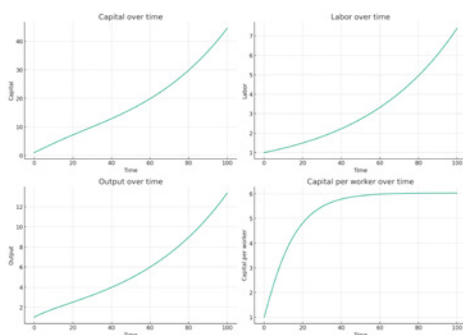
# Plot labor over time
axs[0, 1].plot(times, L)
axs[0, 1].set_title('Labor over time')
axs[0, 1].set_xlabel('Time')
axs[0, 1].set_ylabel('Labor')

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

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

# Make the layout tight
plt.tight_layout()

plt.show()
```



Here are the four plots for the Solow growth model:

- 1. Capital over time:** The capital stock increases over time and approaches a steady state. This steady state is the point at which investment (saving) equals depreciation, so the capital stock no longer changes. [Author’s note: although the numerical simulation is correct, this point of the description is clearly wrong.]
- 2. Labor over time:** Labor grows exponentially at a constant rate, reflecting the assumption of constant population growth in the Solow model.
- 3. Output over time:** Output also increases over time and approaches a steady state. This is because output is a function of capital and labor, which are both increasing. However, due to diminishing returns to capital, the growth of output slows down over time.
- 4. Capital per worker over time:** Capital per worker also increases over time and approaches a steady state. This is the point at which the economy achieves balanced growth in the Solow model. At the steady state, output per worker and capital per worker grow at the same rate, which is the rate of technological progress. In this simplified version of the model without technological progress, this growth rate is zero, so output per worker and capital per worker are constant at the steady state.

These results illustrate the key predictions of the Solow growth model: in the long run, an economy approaches a steady state level of capital (both total and per worker), and this steady state level is determined by the savings rate, the depreciation rate, and the rate of population growth. Changes

in these parameters can lead to a higher or lower steady state, but they cannot sustain long-run economic growth. According to the model, long-run economic growth can only be sustained by technological progress.

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

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

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

Outlining a Plan for a Coding Project [All New in June 2024].—Cutting-edge LLMs, in particular Claude 3, also have the ability to outline a plan for how to approach coding projects. In the spirit of chain-of-thought prompting, they frequently deliver better results if first asked to outline a plan on how to code a programming task and then execute the plan.

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 23 (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 on the preceding conversation, provides an example:

Chat 23 (GPT-4) ctd.: *Can you translate the function into Matlab?*

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

```
function [beta1, beta2] = regression(X, y)
    X = [ones(size(X, 1), 1), X];
    XtXinv = inv(X' * X);
    beta1 = XtXinv * X' * y;
    residuals = y - X * beta1;
    V = diag(residuals.^2);
    Vinv = inv(V);
    XtVinvXinv = inv(X' * (Vinv * X));
    beta2 = XtVinvXinv * X' * (Vinv * y);
end
```

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

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

3.5.4 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 24 (GPT-4): *Traceback (most recent call last): File "C:/.../test.py", line 20, in <module> X = np.array([[1, 2], [3, 4], [5, 6], [7, 8]]) NameError: name 'np' is not defined.*

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

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

```
“python
import numpy as np
”
```

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

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

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

Chat 25 (GPT-4): *Can you debug the following function? [Code from preceding example, with a 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 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 help locate data sources, format data, extract data from text, classify and score text, create figures, extract sentiment, and even simulate human test subjects. Most of these capabilities can be accessed not only through a web interface as shown in the demonstrations below, but also via an API that allows large amounts of data to be formatted, extracted, classified, et cetera. 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 Locating Data Sources [All New in June 2024 Version]

LLMs can be useful starting from the first step of data analysis, that is, in identifying and locating appropriate data sources. The following example illustrates the role

that Claude 3 can play in advising users on what data sources to consider:

Chat 26 (Claude 3): *What data would be useful to analyze the effects of AI on the labor market and where can I find it?*

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

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

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

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

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

3.6.2 Creating Figures

One of the most useful functions of ChatGPT for economists is Advanced Data Analysis, which employs the coding capabilities of GPT-4 to create versatile figures and graphs.

In 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.¹⁰

Chat 27 (ChatGPT Advanced Data Analysis): *Can you display a graph that shows the performance of these securities normalized to start at 100?* [Uploaded stock_data.csv]

[Code generated and executed.]

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

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

[Code generated and executed.]

[Code generated and executed.]



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

From the graph, we can see how each of these securities has performed relative to its starting point over time. This type of comparison can be helpful when assessing the relative performance of different securities or indices.

Chat 27 (ChatGPT Advanced Data Analysis) ctd.: *Calculate the beta for each of the stocks and add it to the legend. Change the colors to*

¹⁰To compile the underlying stock market data, I asked ChatGPT to write a script to download the data, as described in the online appendix of the paper.

make them easier to distinguish. Use the title “One-Decade Performance of Three Big Tech Companies.”

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

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

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

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

where:

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

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

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

[Code generated and executed]

Here are the beta values for the stocks:

- Google (GOOG): 1.1317
- Meta (META): 1.2537
- Microsoft (MSFT): 1.2108

These values indicate that all three stocks are generally more volatile than the market (S&P 500), since their beta values are greater than 1. Now let’s update the graph with the new title, colors, and the beta values in the legend.

[Code generated and executed.]

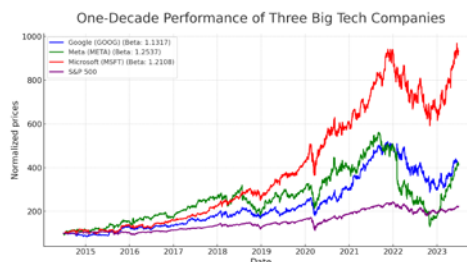


Here is the updated graph with the new title, colors, and beta values included in the legend.

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

Chat 27 (ChatGPT Advanced Data Analysis) ctd.: Double the size of the title, legend, axes labels, and ticks. Label “^GSPC” in the legend as “S&P 500.”

[Code generated and executed.]



Here is the updated graph with the requested changes:

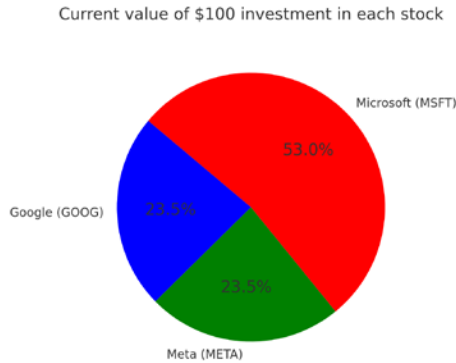
- The title, legend, axes labels, and ticks have all been doubled in size.
- The “^GSPC” label in the legend has been changed to “S&P 500.”

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

Chat 27 (ChatGPT Advanced Data Analysis)

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

[Code generated and executed.]



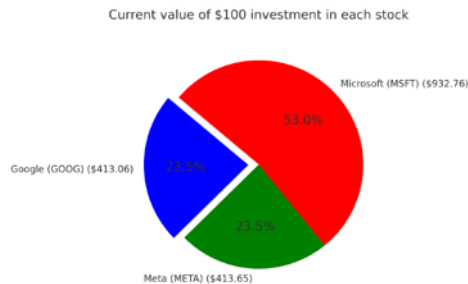
Here is a pie chart that shows the current value of a \$100 investment made in each of the three big tech stocks (Google, Meta, and Microsoft) at the start of the period represented in the data.

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

Chat 27 (ChatGPT Advanced Data Analysis)

ctd.: Can you label each position with its current dollar value in addition to the percentage? Can you make one of the pieces pop out of the pie?

[Code generated and executed.]



Here is the updated pie chart. Each slice now includes both the current dollar value and the percentage of the total portfolio that it represents. The first slice, representing the Google stock, is “exploded” or offset from the pie to highlight it. This visual effect can be useful for emphasizing a particular piece of a pie chart. In this case, it emphasizes that Google has the highest value of the three stocks based on an initial \$100 investment.

[Author’s note: the figure is correct, but ChatGPT’s statement that Google has the highest value is false—this serves as a reminder that Advanced Data Analysis cannot “see” the figures it creates.]

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

3.6.3 Extracting Data from Text

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

Chat 28 (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 and econ grade and math grade.

Mark and A and B+
Sally and A– and A–
Frank and B and C

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

Other applications of extracting data from text include numerical data, for example, 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 be 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.

3.6.4 Reformatting Data

LLMs are equally versatile at 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 28 (GPT-4) ctd.: *Reformat in CSV format.*

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

Chat 28 (GPT-4) ctd.: *Write as a LaTeX table.*

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

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

3.6.5 Classifying and Scoring Text

Social science research frequently employs statistical techniques to represent text as data (Gentzkow, Kelly, and Taddy 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.¹¹ Following the principle of chain-of-thought prompting suggested by Wei, Wang et al. (2022), the prompt asks first for the justification in order to induce the LLM to reason about its response before performing the actual classification. This is akin to asking a student to think before they respond to a question.

Chat 29 (GPT-4): *Here are five 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

¹¹Eloundou et al. (2023) employ GPT-4 in this manner to systematically estimate the labor market impact of LLMs.

4	Maintain and update filing, inventory, mailing, and database systems.	Mostly data entry and retrieval, easily automated.	Easy
5	Compile and prepare reports, charts, and graphs.	Data processing and visualization can be automated.	Easy

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

3.6.6 *Extracting Sentiment*

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

Chat 30 (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.¹² 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.

3.6.7 *Simulating Human Subjects*

Argyle et al. (2023) 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 sociodemographic 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 (2023) showcases applications to economics, using simulated test subjects to replicate and extend upon several behavioral experiments. Brand, Israeli, and Ngwe (2023) demonstrate how to use LLMs to obtain useful information for market research.

¹²See, for example, <https://www.cnbc.com/2023/02/01/live-updates-fed-rate-hike-february.html>.

The following example illustrates the concept:

Chat 31 (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 nonbinary
 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, Jabarian, and List (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 is growing but still limited at this point. The reasons for the relatively poor performance include that the training data for current LLMs is mostly text based and includes comparatively little math, and that math requires high-level abstractions that are still difficult for current

systems. Frieder et al. (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. (2023) show that LLMs can also be fine-tuned for mathematical tasks. Moreover, there have been noticeable performance gains in math going from GPT-3.5 to GPT-4, as documented, for example, 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 32 (GPT-4): *Set up the optimization problem of a consumer with CES preferences over two goods in LaTeX code.*

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

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

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

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

3.7.2 Deriving Equations

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

Chat 32 (GPT-4) ctd.: *Solve the optimization problem.*

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

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

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

$$\frac{\partial \mathcal{L}}{\partial x_1} = \alpha x_1^{\rho-1} (\alpha x_1^\rho + (1 - \alpha)x_2^\rho)^{\frac{1}{\rho}-1} - \lambda p_1 = 0,$$

$$\frac{\partial \mathcal{L}}{\partial x_2} = (1 - \alpha)x_2^{\rho-1} (\alpha x_1^\rho + (1 - \alpha)x_2^\rho)^{\frac{1}{\rho}-1} - \lambda p_2 = 0,$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = M - p_1x_1 - p_2x_2 = 0.$$

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 32 (GPT-4) ctd.: *Derive the demand functions.*

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

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

[Author's note: the term $\frac{\alpha}{1-\alpha}$ is omitted here.]
Rearranging gives:

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

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

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

Solving for x_1 gives the demand function for good 1:

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

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

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

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

As the example illustrates, LLMs still routinely make mistakes in mathematical derivations. I entered “When you divided the two equations, you forgot the alpha terms.” The LLM proceeded to re-derive the results and 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 33 (GPT-4): *Can you explain what this model is, how to get from one step to the next, and what the intuition for each step is?* [Equations pasted in LaTeX format.]

$$\begin{aligned} & \max_{C(t), K(t)} \int_0^\infty e^{-\rho t} U(C(t)) dt \\ \text{s.t.} \quad & \dot{K}(t) = F(K(t), L) - \delta K(t) - C(t)K(0) \\ & \text{given } \lim_{t \rightarrow \infty} K(t)e^{-\rho t} > 0 \\ & \mathcal{H}(t, K(t), C(t), \lambda(t)) = e^{-\rho t} U(C(t)) \\ & \quad + \lambda(t) [F(K(t), L) - \delta K(t) - C(t)] \\ & \quad \frac{\partial \mathcal{H}}{\partial C(t)} = 0 \Rightarrow U'(C(t)) = \lambda(t) \\ & \quad \frac{\partial \mathcal{H}}{\partial C(t)} + \dot{\lambda}(t) = 0 \Rightarrow \lambda(t) \\ & \quad \quad = -\lambda(t) [F'(K(t), L) - \delta] \\ & \quad \lim_{t \rightarrow \infty} \lambda(t)K(t) = 0 \end{aligned}$$

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

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

The Hamiltonian function, denoted by $H(t, K(t), C(t), \lambda(t))$, is a tool used in optimal control theory to solve this kind of dynamic optimization problem. It is a function of time t , state variable $K(t)$, control variable $C(t)$, and costate variable $\lambda(t)$, which is the shadow price of capital. The first-order conditions for the optimal control problem are derived by taking derivatives of the Hamiltonian with respect to the control **and state variables, and setting them equal to zero**. These conditions characterize the optimal paths of consumption and capital over time.

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

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

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

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

TABLE 2
SUMMARY OF LLM CAPABILITIES AND RATING OF USEFULNESS

Category	Task	Usefulness
Ideation and Feedback	Brainstorming	●
	Feedback	◐
	Providing counterarguments	◐
Writing	Synthesizing text	●
	Editing text	●
	Evaluating text	●
	Converting hand-written equations	○
	Generating catchy titles and headlines	●
	Generating tweets to promote a paper	●
Background Research	Generating presentation slides	●
	Summarization	●
	Literature Research	◐
	Formatting References	●
	Translating Text	●
Coding	Explaining Concepts	◐
	Writing code	◐
	Explaining code	◐
	Translating code	●
Data Analysis	Debugging code	◐
	Locating data sources	◐
	Creating figures	◐
	Extracting data from text	●
	Reformatting data	●
	Classifying and scoring text	◐
Math	Extracting sentiment	◐
	Simulating human subjects	◐
	Setting up models	◐
	Deriving equations	○
	Explaining models	◐

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

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

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

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

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 May 2024. 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.

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.

An ongoing strand of research explores how generative AI can be used to automate not only specific micro tasks in research, but broader elements of the research process. For example, Yang et al. (2024) show how to develop an AI agent that can autonomously perform a wide range of coding tasks for a software project. Koehler and Sauermann (2024) examine the ability of LLMs to perform algorithmic management in scientific research, focusing on five key managerial functions that are highlighted in the organizational literature: task division and task allocation, direction, coordination, motivation, and supporting learning.

REFERENCES

- Anderljung, Markus et al. 2023. “Frontier AI Regulation: Managing Emerging Risks to Public Safety.” <https://doi.org/10.48550/arXiv.2307.03718>.
- Ardekani, Aref Mahdavi, Julie Berts, Michael M. Dowling, and Suwan Long. 2023. “EconSentGPT: A Universal Economic Sentiment Engine?” <http://dx.doi.org/10.2139/ssrn.4405779>.
- Argyle, Lisa P., Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. 2023. “Out of One, Many: Using Language Models to Simulate Human Samples.” *Political Analysis* 31 (3): 337–51.
- Bai, Yuntao et al. 2022. “Constitutional AI: Harmlessness from AI Feedback.” <https://doi.org/10.48550/arXiv.2212.08073>.
- Bommasani, Rishi et al. 2021. “On the Opportunities and Risks of Foundation Models.” <https://doi.org/10.48550/arXiv.2108.07258>.
- Brand, James, Ayelet Israeli, and Donald Ngwe. 2023. “Using GPT for Market Research.” Harvard Business School Working Paper 23-062.
- Bsharat, Sondos Mahmoud, Aidar Myrzakhan, and Zhiqiang Shen. 2024. “Principled Instructions are All You Need for Questioning LLaMA–1/2, GPT–3.5/4.” <https://arxiv.org/abs/2312.16171>.
- Bubeck, Sébastien et al. 2023. “Sparks of Artificial General Intelligence: Early Experiments with GPT–4.” <https://doi.org/10.48550/arXiv.2303.12712>.
- Buchanan, Joy, and Olga Shapoval. 2023. “GPT-3.5 Hallucinates Nonexistent Citations: Evidence from Economics.” <http://dx.doi.org/10.2139/ssrn.4467968>.
- Charness, Gary, Brian Jabarian, and John A. List. 2023. “Generation Next: Experimentation with AI.” NBER Working Paper 31679.
- Dunn, Alexander et al. 2022. “Structured Information Extraction from Complex Scientific Text with Fine-Tuned Large Language Models.” <https://doi.org/10.48550/arXiv.2212.05238>.
- Frieder, Simon et al. 2023. “Mathematical Capabilities of ChatGPT.” <https://doi.org/10.48550/arXiv.2301.13867>.
- Gentzkow, Matthew, Bryan Kelly, and Matt Taddy. 2019. “Text as Data.” *Journal of Economic Literature* 57 (3): 535–74.
- Girotra, Karan, Lennart Meincke, Christian Terwiesch, and Karl T. Ulrich. 2023. “Ideas Are Dimes a Dozen: Large Language Models for Idea Generation in Innovation.” <http://dx.doi.org/10.2139/ssrn.4526071>.
- Girotra, Karan, Christian Terwiesch, and Karl T. Ulrich. 2010. “Idea Generation and the Quality of the Best Idea.” *Management Science* 56 (4): 591–605.
- Horton, John J. 2023. “Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?” NBER Working Paper 31122.
- Jiao, Wenxiang, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. “Is ChatGPT a Good Translator? Yes with GPT-4 as the Engine.” <https://doi.org/10.48550/arXiv.2301.08745>.
- Koehler, Maximilian, and Henry Sauermann. 2024. “Algorithmic Management in Scientific Research.” *Research Policy* 53 (4): 104985.
- Korinek, Anton. 2023a. “Language Models and Cognitive Automation for Economic Research.” NBER Working Paper 30957.
- Korinek, Anton. 2023b. “Generative AI for Economic Research: Use Cases and Implications for Economists.” *Journal of Economic Literature* 61 (4): 1281–1317.
- Noorbakhsh, Kimia, Modar Sulaiman, Mahdi Sharifi, Kallol Roy, and Pooyan Jamshidi. 2023. “Pretrained Language Models Are Symbolic Mathematics Solvers Too!” <https://doi.org/10.48550/arXiv.2110.03501>.
- Peng, Sida, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. “The Impact of AI on Developer Productivity: Evidence from GitHub Copilot.” <https://doi.org/10.48550/arXiv.2302.06590>.
- Seeger, Elizabeth et al. 2023. “Open-Sourcing Highly Capable Foundation Models: An Evaluation of Risks, Benefits, and Alternative Methods for Pursuing Open-Source Objectives.” <https://arxiv.org/>

- abs/2311.09227.
- Wei, Jason, Xuezhi Wang et al. 2022. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” In *Advances in Neural Information Processing System (NeurIPS 2022)*, Vol. 35, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Red Hook, NY: Curran Associates Inc.
- Yang, John et al. 2024. “SWE-agent: Agent–Computer Interfaces Enable Automated Software Engineering.” <https://arxiv.org/abs/2405.15793>.
- Zou, Andy, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. “Universal and Transferable Adversarial Attacks on Aligned Language Models.” <https://doi.org/10.48550/arXiv.2307.15043>.