New Tools, New Rules: A Practical Guide to Effective and Responsible GenAI Use for Surveys and Experiments Research

Simon J. Blanchard
Nofar Duani
Aaron M. Garvey
Oded Netzer
Travis Tae Oh

Version Dated May 17th 2025

Note: All the authors contributed equally to this research. They are listed alphabetically.

New Tools, New Rules: A Practical Guide to Effective and Responsible GenAI Use for Surveys and Experiments Research

Generative Artificial Intelligence (GenAI) tools based on Large Language Models (LLMs) are quickly reshaping how researchers conduct surveys and experiments. From reviewing the literature and designing instruments, to administering studies, coding data, and interpreting results, these tools offer substantial opportunities to improve research productivity and advance methodology. Yet with this potential comes a critical challenge: researchers often use these systems without fully understanding how they work. This article aims to provide a practical guide for effective and responsible GenAI use in primary research. We begin by explaining how GenAI systems operate, highlighting the gap between their intuitive interfaces and the underlying model architectures. We then examine different use cases throughout the research process, both the opportunities and associated risks at each stage. Throughout our review, we provide flexible tips for best practice and rules for effective and responsible GenAI use, particularly in areas pertaining to ensuring the validity of GenAI coded responses. In doing so, we hope to help researchers integrate GenAI into their workflows in a transparent, rigorous, and ethically sound manner. Our accompanying website (questionableresearch.ai) provides supporting materials, including reproducible coding templates in R and SPSS and sample pre-registrations.

1. Introduction

Generative Artificial Intelligence (GenAI) systems based on Large Language Models (LLMs) are rapidly emerging as some of the most transformative technologies of our time. Trained on massive datasets and powered by advanced neural network architectures, these models generate coherent, contextually relevant responses in real-time, enabling applications ranging from sophisticated writing and analysis to complex coding and content creation. Their remarkable capabilities have driven widespread adoption across industries and academic disciplines (e.g., Arora, Chakravorty, and Nishimura, 2024; Rathje et al. 2024).

The appeal of these systems is clear: they provide powerful tools for text generation, data analysis, and research support, offering the potential to enhance the research process. However, their growing adoption presents a challenge: those who use GenAI the most frequently often understand it the least (Tully, Longoni, and Appel, 2025). This gap in understanding is especially important because GenAI's performance depends heavily on how it is used (see Blyte et al. 2025; Tomaino, Cooke, and Hoover 2025). Choosing the right model, crafting effective prompts, and rigorously validating outputs can substantially improve the quality and reliability of its results. As these technologies continue to improve, and as researchers become more skilled in working with them, they will learn to better harness their potential and more carefully anticipate and mitigate their shortcomings. In other words, both uncritical enthusiasm and premature dismissal may miss the point: the utility of GenAI in research depends not only on what the system can do, but also on how well researchers understand and deploy it.

The path toward effective and responsible use of GenAI in research, therefore, begins with a deeper understanding of how these systems operate. Behind the intuitive interfaces of tools like ChatGPT, Claude, and Gemini lie sophisticated architectures that incorporate retrieval mechanisms, computational tools, and multimodal processors. Having a basic understanding of how these components interact is crucial, not necessarily for the sake of technical proficiency, but to harness GenAI's full potential while mitigating its risks.

This article provides a guide for researchers seeking to integrate GenAI into their experimental and survey research processes. Section 2 provides an overview of how GenAI models function, starting with LLMs as the foundation and expanding into the system-level enhancements that improve their capabilities. Section 3 builds on these principles to explore their implications for research. Focusing on primary data collection, we review how researchers can utilize GenAI at different stages of the research process, from reviewing literature and designing studies to administering surveys and analyzing data with an eye on validation. Section 4 provides generalized principles for the appropriate use of GenAI systems as the technology evolves.

Finally, to support the responsible and effective adoption of GenAI in marketing research, we provide a range of resources on our companion website, www.questionableresearch.ai. These materials include downloadable codes for coding, pre-registering and validating the coding procedures of GPT-coded responses using SPSS and R. By doing so, we aim to help researchers apply the principles with transparency and confidence.

2. An Overview of LLMs and GenAI Systems

Effectively harnessing any research tool requires a deep understanding of its underlying mechanisms. Just as a researcher must grasp the principles of statistical analysis before interpreting results generated by statistical software, understanding the core technology behind GenAI is essential for its responsible and effective use.

Widely used GenAI systems like ChatGPT, Claude and Gemini consist of multiple interconnected components, yet at their core lies an LLM. These advanced neural networks power the system's ability to generate text, retrieve information, and support computational analysis. Because LLMs serve as the foundation of these systems, this section begins by examining how LLMs function. Section 2.1 explores their training processes, neural architectures, and the mechanisms that enable them to generate contextually relevant text. Understanding how LLMs are developed also highlights their inherent constraints and limitations. Section 2.2 discusses these constraints, which not only shape the behavior of LLMs but also influence the broader GenAI systems in which they are embedded. With this LLM foundation in place, the discussion then broadens to consider the full GenAI ecosystem. Section 2.3 examines how modern systems integrate additional components, such as input processing mechanisms, retrieval systems for real-time information access, and external tools for code execution and data analysis, to extend LLM functionality. This section provides a structured understanding of how the components of GenAI systems interact to support research.

2.1. The Foundational Model: The LLM

2.1.1 Data Gathering and Preparation. LLMs are built from large diverse datasets. While general-purpose LLMs like ChatGPT are trained on a corpus of diverse web and print texts, others have more specific training data (used in fields like law, medicine, or marketing). Once the corpus is assembled, it must be cleaned and organized so that the model can learn its patterns.

The first key step in this process is tokenization, where text is broken down into smaller components known as tokens. These tokens can represent entire words, subwords, or characters, depending on the tokenization strategy. For example, the phrase "Limited-time offer on unbelievably good deals!" is tokenized by GPT-4 into the following sequence:

[75577, 7394, 3085, 389, 40037, 89234, 1695, 12789, 0]

which corresponds to the subword tokens:

$$t_1 = Limited, t_2 = -time, t_3 = offer, t_4 = on, t_5 = unbelie, t_6 = vably, t_7 = good, t_8 = deals, t_9 = !]$$

Common words like “Limited” and “time” appear as single tokens, while less frequent words or morphologically complex words such as “unbelievably” are split into subword tokens like “unbelie”, ”vably”. This subword tokenization helps the model handle rare words while still leveraging patterns learned from more frequent components.

Once tokenized, each word or subword is represented by a numerical token ID drawn from the model’s vocabulary. To make these tokens usable for learning, the model maps each token ID to a high-dimensional embedding vector through a learned embedding matrix. This matrix can be thought of as a giant lookup table, where each row corresponds to a token ID and contains a vector that represents that token’s learned properties. When the model encounters token 75577 (“Limited”), it retrieves the corresponding vector $x_1 = E[75557]$, where E is the embedding matrix. If the embedding size is 4,096, this means that the word “Limited” is now represented by a 4,096-dimensional vector containing real-valued numbers. Importantly, all tokens are embedded in the same continuous vector space, meaning each token is represented using the same dimensions, which is what will allow the model to compare, combine, and contextualize them. This embedding space is also dramatically smaller than the token ID space: for a vocabulary of over 100,000 tokens, the model compresses these sparse, discrete identifiers into dense vectors with thousands (not hundreds of thousands) of dimensions.

Yet, at this stage, these vectors are simply arrays of real numbers. They are not interpretable or semantically meaningful. In fact, the embedding matrix is randomly initialized at the start of training, and each token’s vector (e.g., [0.014, -0.732, 0.085, ...]) reflects no knowledge about language or meaning. It is only through exposure to massive amounts of text, and the iterative refinement of the model’s parameters through training, which comes next, that these vectors come to encode useful linguistic patterns.

2.1.2 Model Training.

The core objective during model training is for the LLM to learn how to use context to predict the next token. Context is essential to language understanding. For example, when encountering the word "model" in this article, most readers infer that it refers to a computational system (not a fashion figure) because of the surrounding words (and possibly familiarity with the authors). LLMs must learn to make such inferences automatically, without external knowledge or explicit definitions. Although each token is eventually represented by a high-dimensional embedding vector, as just discussed, these vectors begin as random numbers with no built-in structure. Through training, the model needs to learn to refine embeddings so that tokens with similar contextual patterns develop similar internal representations.

The learning process unfolds through iterative steps of prediction and feedback. Once inputs have been converted to embedding vectors, the model passes them through a series of layers that help learn how the embeddings relate to one another in context. At the heart of this architecture is a mechanism called self-attention, which enables the model to assign dynamic weights to each token embedding based on its relevance to other token in the sequence. In essence, the model learns to decide which parts of a sentence should inform its interpretation.

For example, in the phrase "Limited-time offer on unbelievably good deals!", the model learns (through exposure to many similar examples), that "unbelievably" should strongly influence the interpretation of "good deals," and amplify its intensity. Likewise, it learns that "Limited-time" contributes important contextual meaning to "offer," by signaling urgency. These relationships are not provided in the training data; they are inferred through prediction error.

As input data passes through these layers, the model generates contextualized hidden states: refined representations of each token that incorporate its meaning and function relative to the surrounding text. Using these final hidden states of the last token in a sequence, the model produces a probability distribution over all possible next tokens. For example, given "Limited-time offer on", the model might assign high probabilities to plausible continuations:

$$P(unbelievably) = 0.65, P(amazing) = 0.20, P(our) = 0.1, P(good) = 0.05$$

In this case, "unbelievably" receives the highest probability because it often appears in promotional language following phrases like "Limited-time offer." If the model instead predicted a less likely word, such as "good", it would compare its prediction to the actual next token in the training data and compute a prediction error.

This error is then used to update all relevant parameters, including not only the layers of the deep neural architecture but also the embeddings themselves, through a process that is known as backpropagation. For example, if the model predicted "amazing" instead of "unbelievably," it might slightly alter the embedding for unbelievably to make it more distinct in similar contexts, while also refining how the model interprets related modifiers like "amazing." Crucially, this learning process does not occur in isolation. The model undergoes this cycle of prediction and correction across billions of training examples, drawn from a wide range of sources including books, websites, news articles, and more. Text is processed in mini-batches, and the model incrementally updates its parameters after each batch. Over time, these iterative updates allow the model to internalize complex aspects of language (e.g., syntax, associations, and nuances).

As models scale, so too does their context window (i.e., the number of tokens they can process at once to infer any specific token). Earlier models like GPT-2 supported up to 1,024 tokens per input sequence. GPT-3.5 increased this to 4,096, and GPT-4o now handles up to 128,000 tokens. These larger context windows enhance the model's ability to maintain coherence across paragraphs, understand long documents, and engage in more context-aware conversation.

2.1.3 Content Generation.

Once an LLM is trained, it can generate new text in response to a prompt: a process known as inference. During inference, the model's parameters remain fixed: there is no learning, no updates, and no feedback loop. The model draws entirely on what it learned during training to predict and generate one token at a time, sampling from a probability distribution over its vocabulary. A common misconception is that prompting changes the model. It does not. Prompts influence which knowledge the model draws on, but they do not alter how the model works. The model's internal weights remain unchanged regardless of what the input might be. This fixed nature is what allows the same model to be deployed consistently across users and use cases. Consider the prompt:

> "Help me think of a good call-to-action statement for my marketing promotion: 'Limited-time offer on our new product'."

After tokenization, the input is converted into a sequence of tokens such as:

$$[t_1 = "help", t_2 = "me", t_3 = "think", t_4 = "of", \ldots, t_{16} = "product"]$$

Each token is embedded and passed through the model's layers to generate a final hidden state. Using this internal representation, the model produces a probability distribution over all possible next tokens. For example, it might assign:

$$P(\text{“}Consider\text{”}) = 0.40,\ P("Try") = 0.20, P(\text{“}Use\text{”}) = 0.15, P(\text{“}Act\text{”}) = 0.10,\ P("Jump") = 0.08, P(\text{“}Explore\text{”}) = 0.07$$

This output is not a single prediction, but a distribution. To generate text, the model must apply a choice function that converts this distribution into a specific token. At the most extreme the model may simply select the token with the highest probability ("Consider"). A more common approach used by LLMs is to sample from the full or a subset of the probability distribution.

There are several ways to adjust how sampling works, depending on the desired output. The *temperature* setting controls how focused or varied the model's responses are: lower values (e.g., 0.2) make the model more predictable by emphasizing high-probability tokens, while higher values (e.g., 1.0+) make responses more diverse by giving lower-probability tokens a better chance. Additionally, one can limit the set of tokens the model samples from using *top-k* sampling (restricting choices to the top *k* most likely tokens) or *top-p* (nucleus) sampling, which selects from the smallest set of tokens whose combined probability exceeds a threshold *p* (e.g., 80%). These settings are often adjustable via chat interfaces or API calls.

Once the next token has been selected the updated sequence is then reprocessed, generating a new hidden state, and the model predicts the next token in the same way. This continues iteratively (predict distribution, select token, append to input sequence) until the model produces a special stopping token or hits a predefined token limit. The final output might unfold:

$$[t_{\{m+1\}} \ = Consider, t_{m+2} \ = using, t_{m+2} :, t_{m+3}\ Limited, \dots, t_{m+k}\ 20!]$$

Which is rendered to the user as: "Consider using: 'Limited-time offer on our new product: Buy now and save 20%!'" At each step, the model generates the next token based on the prompt and all previously generated tokens, ensuring local coherence and contextual relevance. However, it does not retain memory of previous prompts or outputs beyond the current session. Each generation session is stateless: the model starts fresh unless content is provided in the prompt.

2.2 Capability Constraints of LLMs

Despite their impressive capabilities, LLMs possess several inherent limitations that shape their performance, stemming directly from how they are built and trained. These constraints are summarized in Table 1. First, LLMs' knowledge is static, drawn from predetermined training datasets that may be outdated or incomplete; this is a consequence of being trained on specific datasets at a particular point in time (Section 2.1.1) and their core parameters remaining fixed after this training period (Section 2.1.3). Importantly, LLMs do not retrieve data in the traditional sense like a database. Instead, they reconstruct information by synthesizing complex patterns and statistical relationships, a core aspect of their training process where they learn to predict subsequent text based on context (Section 2.1.2). Because LLMs generate outputs based on this pattern-based reconstruction, they prioritize linguistic plausibility rather than guaranteed factual accuracy, and can produce convincing but incorrect statements. This fundamental characteristic, combined with their reliance on a finite context window to process information (Section 2.1.2), means they cannot directly recall or perfectly retrieve past inputs beyond the current context window. This limits their usefulness for summarizing lengthy materials or conducting iterative tasks like multi-session interviews. Their responses may also vary with identical prompts due to the probabilistic nature of their text generation process, where they select from a distribution of possible next tokens (Section 2.1.3), posing challenges for reproducibility. Finally, given that they are centered on understanding and generating language, they inherently lack the ability to run code or perform computations independently.

2.3 Accessing and Expanding the Capabilities of LLMs in GenAI Systems

Researchers typically interact with LLMs through broader GenAI systems, which integrate additional components such as user interfaces, tools for data input (e.g., document uploads) or code execution environments with a core LLM "engine." Researchers can access isolated LLMs or broader GenAI systems in multiple ways, each with different trade-offs in terms of computational requirements, flexibility, and reproducibility. Researchers must consider both the access method and the system's additional capabilities when selecting what GenAI to use. We provide a summary of the core considerations for researchers in Table 2.

Table 1 – Researcher-relevant Implications of How LLMs Work

LLM Constraint	Definition	Example of Impact on Researchers
Predetermined Corpora	LLMs are trained on a fixed dataset; they cannot update their knowledge after training.	A researcher may miss relevant information not present in the training corpora because the model does not contain all the relevant information.
No Retrieval Capability	LLMs do not store training data but instead high-dimensional vector representations.	When summarizing a study, the model may omit or misrepresent findings and have an inability to retrieve the original text.
Text-Only Inputs and Outputs	LLMs process and generate text exclusively, without native support for multimodal inputs or outputs.	A researcher analyzing survey responses with visual stimuli cannot directly process images or tables without additional tools.
Limited Input and Output Size	LLMs have a fixed token limit for processing inputs, and thus sometimes truncate text.	When analyzing long research papers, key sections may be omitted from analyses without warning or explanation.
Lack of Persistent Memory	LLMs process prompts in each session independently without retaining memory beyond the current context.	A researcher conducting an iterative interview may find that the model forgets prior responses unless they are re-included in the prompt.
Optimized for Plausibility	LLMs prioritize generating fluent, coherent text rather.	A literature summary may contain fabricated citations or misinterpreted conclusions.
Probabilistic Output	LLMs generate text by sampling from probability distributions, leading to variations across repeated prompts.	When using LLMs to code or label textual information a researcher may receive different classifications each time they run the model
Inability to Run Code	LLMs can generate code but generally cannot execute or verify it.	A researcher asking the model for statistical analysis may receive a plausible-looking but incorrect result.

Table 2: Comparison of GenAI Access Methods

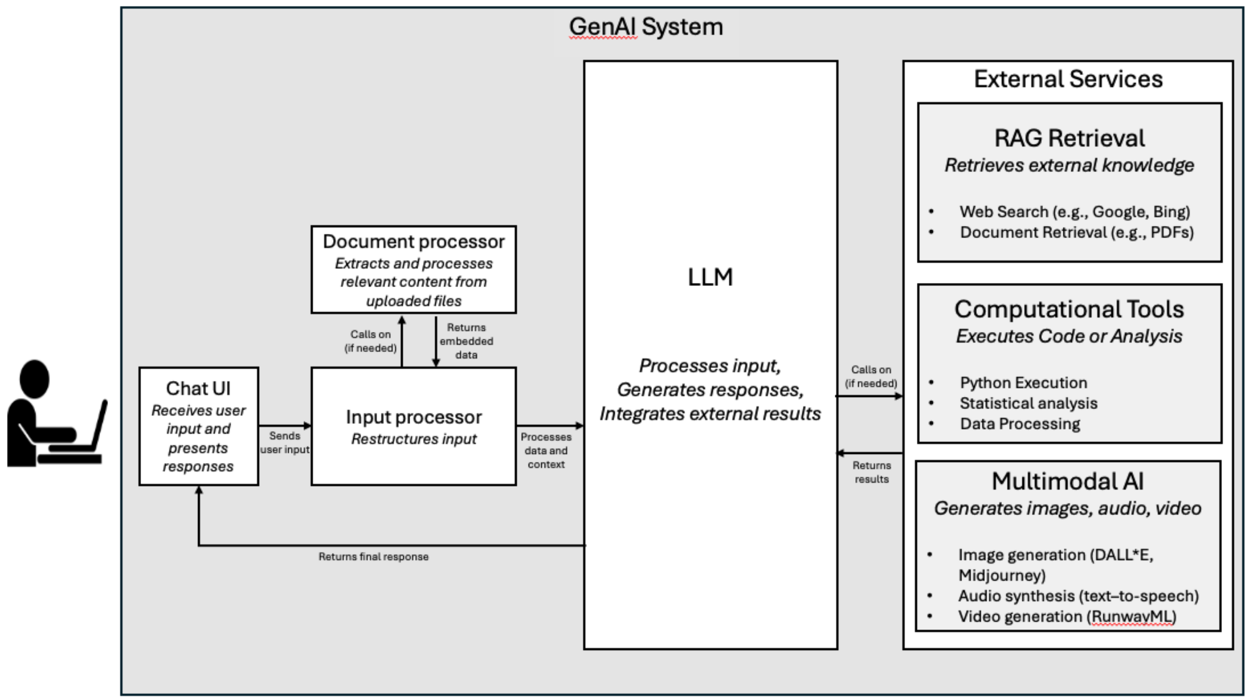
Access Method	Best For	Available Capabilities	Limitations	Data Privacy Considerations
Web-Based Chat Interfaces	Literature review, document analysis, summarization, exploratory research, Writing assistance	Document uploads, web search, image /audio inputs, and real-time code execution	Less reproducible and scalable than API-based access, limited integration with statistical package	Inputs may be stored by providers unless using enterprise accounts.
API-Based Access	Repeated tasks for many text units such as labeling tasks of open text responses, Reproducible workbooks	Allows large-scale text processing, integration into research codes, and fine-tuning.	No web search, limited input types, interactive chat UI, or external outputs. Requires (some) programming knowledge.	Researchers must verify provider policies; some API calls can be used for model training.
Local Model Software	Security-sensitive research, qualitative coding of sensitive human subject data adapting the model.	Full control over data, ability to fine-tune, does not send data to external servers	Requires computational resources, lacks built-in web search, uploads, or multimodality	Full data privacy. Best option for confidential or proprietary research data

2.3.1 Accessing LLM-powered GenAI systems via Web-based Chat Interfaces

Arguably the most common way researchers interact with LLMs is through web-based chat interfaces, such as ChatGPT (OpenAI), Claude (Anthropic), Gemini (Google), and

DeepSeek Chat. These platforms allow users to engage with GenAI in a conversational format, making them particularly appealing for tasks that require iterative refinement, brainstorming, document analysis, or exploratory research. Chat interfaces often incorporate external services such as retrieval, document processing, code environments, and multimodal capabilities for inputting and generating images, audio, or video. Figure 1 illustrates these components.

Figure 1 – A Sample (Simplified) GenAI System's Architecture



One of the major enhancements in a GenAI system versus a core LLM is input processing, which allows users to provide more complex inputs beyond simple text prompts. This is particularly useful in research applications, where users may need to upload documents, datasets, or prior study results. Since LLMs process only linear sequences of text, they cannot natively interpret structured files such as PDFs, or spreadsheets. To enable document understanding, GenAI systems often include a *Document Processor*, which extracts, segments, and embeds uploaded content into a format the LLM can understand. This process is necessary because when a user uploads a document, the system doesn't retrain the model on its contents. Instead, the document is parsed and relevant parts are added to the user's query as additional context for the model to consider. However, since LLMs can only process a limited amount of information at once, the system must selectively extract and prioritize the most relevant sections of the document to include in the prompt and ensure it fits within the model's maximum input size. Moreover, many GenAI systems can incorporate *Multimodal Inputs and Outputs*. While an LLM alone cannot interpret non-text data, a multimodal system converts images into embeddings, allowing the AI to describe, categorize, or create non-text elements.

Once the input is processed, the LLM generates responses based on the information provided. However, unlike standalone LLMs, a GenAI system can enhance this step by incorporating external knowledge sources through retrieval mechanisms. *Retrieval-Augmented Generation (RAG)*, for instance, allows the system to fetch real-time information from proprietary research databases, academic articles, or external web sources. In doing so, a GenAI system reduces, but does not eliminate, the risk of outdated or incorrect responses. Importantly, this is *not* training the LLM, which remains static, but instead expands the current context window to temporarily include additional information. In addition to retrieval, modern GenAI systems improve LLMs' analytical capabilities by integrating external tools. For example, a *Code Environment* can enable the LLM to generate Python syntax, submit it to the code environment, wait for the code environment to return the results, and add the outputs of the code environment back to the context to generate a user response. By doing so, the GenAI runs codes.

A key challenge when using web-based chat interfaces is that it is not always obvious which LLM powers the system, nor what that means for research applications. Unlike API-based or local installations, where researchers explicitly choose a model, chat-based interfaces often obscure the underlying model, its update frequency, and its full range of capabilities. This can make it difficult for researchers to determine which GenAI system best fits their research needs, especially as the choice of model and access method directly influences supported context window size, multimodal processing, customization, and data security (see Table 1).

2.3.2 Accessing LLMs via API.

While web-based chat interfaces offer an intuitive and readily accessible entry point for interacting with GenAI systems, researchers with more technical needs or requirements for integration into larger workflows may benefit from the structured capabilities of API-based access. With API-based access, users send queries directly to externally hosted models using custom code, enabling the incorporation of LLM functionalities into applications. This akin to the difference between batch versus real-time processing in code execution. Major LLM providers such as OpenAI, or Anthropic offer API-based access, charge based on token usage.

To illustrate how a researcher can access an LLM via the API, consider the sample R code provided in Table 3 to submit our sample prompt from Section 2.1.3 to OpenAI's GPT-4o via API. After loading two libraries, the researcher sets an API key (i.e., a unique identifier that allows OpenAI to know who to charge) and specifies the API endpoint (i.e., where the researcher needs to send their request to have a prompt response). After setting a few key parameters (i.e., the model to use, the prompt, the maximum length of content to generate, and the model's temperature), the code sends the request to the API and collects the response.

API-based access is particularly valuable for automating large-scale text processing, integrating AI into research practices, and ensuring replicable workflows. For example, using APIs in programming environments like Python or R allows users to design custom scripts to extract, code, or summarize textual data. This makes APIs a powerful tool for automated content analysis, coding large datasets, and generating structured outputs.

Table 3 – Simple R Code to Send a Request to the OpenAI API

```
library(httr) # necessary libraries
library(jsonlite) # necessary libraries

# 1. Set an API key to identify yourself with OpenAI
api_key <- "YOUR_API_KEY_HERE"
# 2. Set the API endpoint – where we reach the OpenAI's API
url <- "https://api.openai.com/v1/chat/completions"
# 3. Prepare our identification to OpenAI
headers <- add_headers("Content-Type" = "application/json", "Authorization" = paste("Bearer", api_key))
# 4. Define the request for the LLM
data <- list(model = "gpt-4o", messages = list(list(role = "user", content = "Help me think of a good call-to-action
statement for my marketing promotion: 'Limited-time offer on our new product")), max_tokens = 1000,
temperature = 0.7)
# 5. Send the request to the LLM and collect its response
response <- POST(url, headers, body = toJSON(data, auto_unbox = TRUE))
```

One important aspect is that API requests are stateless: each prompt is processed independently without retaining memory of past interactions. As a result, for iterative tasks, researchers need to explicitly include all necessary context in each request. In other cases, this feature may be beneficial like when analyzing open-text responses from a survey where one wishes to treat each response independently. Moreover, LLM API calls generally do not support features like document uploads, web search, or real-time code execution. That means that researchers need to ensure that the submitted prompts do not send too much information in a single prompt (i.e., exceed the context window limits). Finally, the information sent by API calls may be logged by the service provider, so it is important to review data retention policies carefully, especially when handling confidential or proprietary data.

2.3.3 Local Model Installation: Running LLMs Privately.

For researchers who need maximum control and privacy, local model installations offer an effective solution despite their higher technical demands. A local LLM is created by downloading the model and its associated weights, similar to how you might download a statistical software to your local machine. While managing these downloads and configurations can be complex for the average researcher, software like Ollama simplifies the process by providing an interface to select, download, and run models locally.

Local models offer significant advantages for researchers, particularly regarding security, customization, and offline functionality. Since all computations occur on a researcher's own machine or an institutional server, data remains private. This makes local models ideal for handling sensitive datasets such as proprietary industry data, confidential human-subject responses, or unpublished research findings. Additionally, locally installed models can be fine-tuned on domain-specific data, enhancing performance on specialized research tasks. Offline functionality is another key benefit, as it allows consistent access without concerns about internet connectivity, server downtime, or data transmission risks.

However, running local models also comes with notable limitations. One major drawback is the computational cost; high-performance language models require significant GPU resources, which can be challenging for individual researchers or small teams. While smaller versions of models, such as LLaMA 3 and Mistral 7B, can operate on personal computers, larger models often demand cloud-based infrastructure with high-performance hardware. Additionally, local models typically lack built-in document processing and web search capabilities, meaning researchers must manually integrate additional tools to extend the model's functionality. Despite these challenges, local models remain a strong choice for those who prioritize privacy, fine-tuning capabilities, and complete control over their AI systems.

3. Potential Uses of GenAI Systems Throughout the Primary Data Research Process

GenAI systems have the potential to significantly enhance the research process. Their inherent versatility allows them to summarize vast bodies of literature, generate survey measures, refine experimental manipulations, conduct interviews, and assist with data analysis. However, alongside these benefits, these systems also introduce significant risks. Without a clear understanding of how they function, researchers may inadvertently rely on them in ways that compromise scientific rigor, introduce biases, or distort findings. Fortunately, a deeper understanding of how these models work (how they are trained, their constraints, and the broader system-level enhancements) can help researchers use GenAI more effectively and responsibly. Building on the foundation laid out in Section 2, this section examines the implications of GenAI technologies across different stages of the research process, from literature review and study design to data collection, analysis, and interpretation. Rather than offering an exhaustive list of possible applications, our goal is to provide a roadmap for critically assessing the risks and benefits of using GenAI at each stage of research.

To illustrate these points, we refer throughout to a single running example based on Grewal et al. (2019), which examined how consumers devalue unattractive but edible produce due to the negative self-perceptions associated with its consumption.

3.1 Stage 1: Interacting with the Literature

A comprehensive literature review is a crucial part of any research paper, enabling scholars to learn from and integrate relevant existing knowledge. GenAI systems, leveraging LLMs, can help summarize information, highlight connections, and navigate extensive bodies of academic work. By drawing on their training data and any integrated retrieval mechanisms such as retrieval-augmented generation (RAG) or document upload support, these systems can surface articles, themes, and patterns that might not be immediately evident through traditional searches.

However, these strengths come with a critical limitation: GenAI systems are not optimized for factual precision. Because LLMs generate responses based on patterns learned from training data rather than direct retrieval of verified information, they can produce summaries that are fluent but factually inaccurate or even fabricate citations altogether. This accuracy gap is further compounded by the fact that LLMs do not have real-time access to academic databases or proprietary sources unless explicitly integrated. As a result, while GenAI can accelerate early-stage exploration and thematic synthesis, its outputs in this domain require especially careful verification and methodological safeguards.

3.1.1 Exploring the Literature

There are numerous ways researchers can use GenAI to explore literature, from conducting broad literature syntheses to extracting key themes and summarizing findings relevant to a given research question. A researcher studying the impact of food product attractiveness on consumer preferences, for example, might start by asking a GenAI chatbot to "identify relevant academic literature about consumers' perceptions of unattractive food products." Through iterative interactions, the researcher can request a summary of key findings, filter articles by discipline, or ask for an extraction of recurring theoretical frameworks.

However, as highlighted in Section 2.2, LLMs have inherent constraints which can impact the accuracy and completeness of their output. To illustrate, we used OpenAI's chatbot without access to web search (i.e., o1). Specifically, we asked: "Identify relevant academic literature about consumers' perceptions of unattractive food products." Among the papers identified, one was listed as:

> de Hooge, I. E., van Dulm, E., & van Trijp, H. C. M. (2018). "Cosmetic Specifications in the Food Waste Issue: Supply Chain Considerations and Consumer Preferences." *Food Quality and Preference*, 56, 126-139.

While the research exists, the model provided the wrong journal. The article was published in *Journal of Cleaner Production* (Volume 193, issue 10, pages 698-709). This hallucination is entirely consistent with the model's property of generating probable responses, even if at times incorrect. The authors and topic are associated with academic research in food waste and consumer preferences and the *Food Quality and Preference* is a well-known journal in this field. However, the chatbot lacked access to a database or validation mechanism to confirm the exact journal, volume, and issue of the reference, and ultimately generated the most probable (though incorrect) combination of these elements.

These risks can be mitigated, by using GenAI systems with retrieval-augmented generation (RAG) which integrate external search. For example, when we repeated the same query with web search enabled (this time with GPT4o), we retrieved the correct citation for the de Hooge et al. article after ChatGPT searched the web. Unfortunately, though RAG reduces the likelihood of purely probabilistic hallucinations, it does not fully eliminate accuracy issues. In this case, when asked for a summary of the article, the system provided two links, neither of which led to the correct publication in the *Journal of Cleaner Production.* Instead, one linked to a similar *de Hooge et al.* (2017) paper in *Food Quality and Preference*, while the other directed to a *Vogue Business* summary of that article. Though RAG does not guarantee accuracy, it can help researchers to trace the origins of information, helping to catch and correct some errors.

3.1.2 Analyzing a Single Article

One of the strengths of LLMs is their ability to contextualize any given text. As such, it is sensible that many researchers would use LLMs to interact with a specific research article, such as summarizing the main contributions of an article, extracting measurement scales, outlining experimental designs, or distilling findings into structured formats like tables for easier comparison across studies. These capabilities could save considerable time and effort. However, relying on LLMs alone for these tasks comes with limitations. Because LLMs are trained on predetermined corpora, they may not have encountered the specific article in question during training. And even if they have, the conversion of tokens to embedding space during model training means the system cannot retrieve the article verbatim or with guaranteed fidelity. As a result, attempts to summarize or extract details from a particular article using only the LLM may yield incomplete or inaccurate outputs. GenAI systems that support RAG, however, can address this issue by enabling document uploads. By complementing and incorporating the full text of a specific article into the model's active context window, these systems allow researchers to ground the LLM's output in the content of the article and can reduce hallucinations.

For example, consider a researcher interested in creating a table summarizing studies from Grewal et al. (2019). Instead of querying a LLM to summarize that research, we uploaded the full article into OpenAI's ChatGPT (model GPT-4o), and requested GenAI to create a table summarizing the studies in the paper: the model extracted manipulations, sample characteristics, dependent variables, and moderators/mediators, producing Table 4.[1] The model produced a useful and well-structured table, which was far more accurate than the table produced without directly uploading the full text, and yet the accuracy of the results is still not guaranteed. For example, for Study 2, the diagnostic value of choice was listed as both the independent variable and the moderator. In reality, the independent variable was produce attractiveness, while the diagnostic value of choice was a moderator. Additionally, the model failed to list negative self-perceptions as a mediator, even though it was measured and assessed in the study. These inaccuracies illustrate how, despite improvements through document processing and structured chunking, GenAI systems do not always extract with perfect fidelity.[2]

---

[1] The exact prompt was "I am a researcher interested in understanding this academic article. I would like to create a summary table of studies, in which, for each study in the article, you identify: The independent variable (be precise as to whether it is measured or manipulated and how), The dependent variable, Any moderator or mediator, The sample source and context, The main finding."

[2] It is important to note that the accuracy of the table is substantially superior to that of relying exclusively on an LLM to generate the same table. We provide one, for the sake of illustration, in Web Appendix A.

Table 4 – Sample Summary of Studies: Grewal et al. (2019) using GPT4o and RAG

Study	Independent Variable	Dependent Variable	Moderator/Mediator	Sample Source & Context	Main Finding
Study 1	Produce attractiveness (manipulated; unattractive vs. attractive images)	Willingness to pay (WTP) for produce	Mediator: Negative self-perceptions	MTurk participants (n = 304) imagining a grocery shopping scenario	Participants valued unattractive produce less, mediated by negative self-perceptions.
Study 2	Diagnostic value of choice (manipulated: self-diagnostic vs. non-self-diagnostic)	WTP for unattractive produce	Moderator: Diagnostic value of choice	Undergraduate students (n = 301) in a simulated shopping task	When choices were not self-diagnostic, the devaluation of unattractive produce was mitigated.
Study 3	Self-esteem boosting task (manipulated; high vs. control)	WTP for unattractive produce	Mediator: Negative self-perceptions	Undergraduate students (n = 191) in a lab-based task with real monetary stakes	Boosting self-esteem eliminated the difference in WTP between attractive and unattractive produce.
Study 4A	In-store ad messaging (manipulated; self-esteem boosting vs. control)	Real choice of unattractive produce	Mediator: Negative self-perceptions	Field study in a Swedish grocery store (n = 130 shoppers)	Self-esteem boosting ads significantly increased the choice of unattractive apples.
Study 4B	In-store ad messaging (manipulated; self-esteem boosting vs. control)	Real choice of unattractive produce	Mediator: Negative self-perceptions	MTurk participants (n = 201) in a simulated retail choice	Replicated Study 4A findings in a controlled environment; negative self-perceptions mediated the effect.

For tasks involving literature review, whether for a specific paper or an entire literature it is recommended to go beyond the use of LLMs and leverage the RAG component of the GenAI system, whether through access to the web, databases of academic research or individual paper upload to increase accuracy. However, given the factual nature of the task of literature reviews, it is still crucial to verify the accuracy of the information retrieved or generated.

3.1.3 Literature review best practices

Researchers using GenAI for literature review should tailor their approach to the task at hand. When the goal is to educate yourself about a new topic and explore existing knowledge (but not necessarily specific papers), interacting with an LLM alone can be a productive way to surface patterns, frameworks, and ideas. Because these models are trained on massive text corpora and excel at recognizing complex patterns (Section 2.1), they can quickly summarize relevant knowledge and make it more accessible, even across different disciplines. This makes them ideal for early-stage ideation and learning.

However, LLMs are not optimized for precision (Section 2.2). When accuracy is important (e.g., citing prior work, identifying whether your research question has already been studied, or determining which papers to include in a meta-analysis), responses must be treated with caution. Researchers should rely on retrieval-augmented GenAI (RAG) systems that support real-time web access and provide sources link, or on document upload workflows that allow GenAI systems to summarize and extract information. Uploading relevant papers ground the system's output in the actual content of the article rather than in secondary summaries, abstracts, or outdated versions of the research. These features can help minimize–though they cannot outright eliminate–the risk of hallucinations.

Researchers should also be mindful of the limitations of GenAI's knowledge constraints. As discussed in Section 2.2, LLMs have fixed predetermined knowledge cutoffs, which means they may not reflect the latest findings in fast-moving domains, and they may be unable to access content hidden behind paywalls. While uploading academic papers into GenAI systems can help with these knowledge constraints, data handling concerns must be considered. Unless researchers are using a GenAI system that guarantees user data will not be used for model training (e.g., local LLM or enterprise-level), they should avoid uploading copyrighted texts or proprietary content (see section 2.3 and Table 2).

We also recommend that researchers use iterative refinement practices when conducting literature review tasks. For example, when working with long academic articles, the system's ability to accurately extract content may deteriorate over the course of a long session due to context window limitations (see Section 2.2). To mitigate this, it is useful to periodically re-upload the document or split it into sections to refresh the model's context.

3.2 Stage 2: Research Design

In the research design stages, researchers develop studies, including measurement instruments and experimental manipulations, to test their hypotheses. GenAI systems can offer valuable assistance in these tasks by recognizing linguistic patterns, adapting existing content, and generating new text in a contextually appropriate manner. These capabilities enable researchers to identify relevant scales, modify items for new contexts, and generate realistic stimuli for experimental conditions. However, as with literature exploration, these capabilities come with important limitations. Unlike human researchers, GenAI does not engage in deductive reasoning or theoretical interpretation. Instead, it generates content based on statistical associations between words in its training data or retrieved sources. This can introduce risks such as construct validity issues in measurement and confounding variables in experimental manipulations. This section explores how GenAI can assist in study development while emphasizing best practices for maintaining rigor in research design.

3.2.1 Developing measurements

Operationalizing the measurement of a construct is a critical step in any research study involving primary data (Churchill 1979). GenAI systems can provide significant advantages, such as helping researchers quickly identify existing measurements by surveying the literature or extract scale items from articles or their web appendices. Yet, one of the most promising uses is the ability of GenAI to quickly adapt scales to new contexts or to generate new scale items. These adaptation tasks align with LLM's strengths: the models can easily produce variations on a given item by substituting words, adjusting tenses, and incorporating new contextual details to transform, for example, from self-focused items to other-focused evaluations, or state-based measurements to trait-based ones. As such, it should be effective to generate promising items.

Consider a study on consumer perceptions of produce attractiveness. A researcher seeking to measure the visual attractiveness of produce may not find a validated psychometric instrument tailored to this specific construct. In this case, they might use a web-based GenAI chat to generate candidate scale items by providing a simple instruction such as: “suggest survey items to measure produce visual attractiveness.” The system might generate Likert-scale items such as: “This produce looks visually appealing to me,” “The color of this produce is vibrant and attractive,” “This produce appears fresh and high-quality,” and “This produce is something I would feel proud to display or serve to others.” These items have clear linguistic coherence and contextual relevance, showing just how efficient GenAI can be as a tool for generating a large set of potential items. However, items produced are not necessarily free of measurement error.

In the example instruction above, we did not define what constitutes visual attractiveness (defined in Grewal et al as the degree of natural aesthetic deviation from a prototypical category exemplar, specifically focusing on natural variation in shape or appearance that arises during growth, and excluding deviations due to damage or spoilage). Without this context, GenAI relies entirely on its attention mechanism to determine relevance (see Section 2.1.2). The model prioritizes terms that frequently co-occur with “produce,” “visual,” or “attractiveness.” As such, the model might interpret “produce” broadly, encompassing any attributes associated with fruits and vegetables, such as freshness, quality, or appeal. Specifically, “attractiveness” may reflect general consumer appeal rather than specific aesthetic properties implied in “visual attractiveness.” As a result, items like “The produce appears fresh and high-quality” creates a departure in measurement from the intended construct.

Even when the generated items seem reasonable, they may introduce systematic errors. The initial set of items includes: 1) Presumptive wording ("This produce is vibrant and attractive"), which assumes vibrancy is universally perceived as attractive, 2) Double-barreled phrasing ("This produce looks fresh and high-quality"), which combines two distinct attributes into a single statement, and 3) Overlapping constructs ("This produce is something I would feel proud to display or serve to others"), which introduced an element of social desirability.

One way to mitigate these limitations is to recommend the inclusion of operational definitions and context when engaging in item generation, and the use of iterative refinement to mitigate such issues when using GenAI for measurement development. For example, the researcher could opt to generate items while including, in the prompt, the operational definition of visual attractiveness provided in the text of Grewal et al. (2019).[3] In doing so, we find that the model generated items that better capture visual attractiveness as intended. For example, it included items such as: "This produce closely resembles what I expect an ideal [produce type] to look like," "There is little to no natural variation in the appearance of this produce compared to what I would consider a standard example," and "The physical appearance of this produce is consistent with my mental image of a perfect [produce type]."

However, one may note that even including this operational context did not eliminate systematic measurement error entirely. For example, double-barreled items were still produced, and some recommended semantic differential items with endpoints that were not opposites (e.g.,

[3] We asked for revision based on: "What if we conceptualize produce attractiveness in terms of the degree of natural aesthetic deviation from the prototypical category exemplar of physical appearance. As such, unattractive produce is defined as having significant natural variation from prototypicality, whereas attractive produce is defined as having limited (if any) variation from prototypicality. Given this conceptualization, we limit our focus to the natural variation in physical appearance that arises during a product's growth (e.g., an apple's odd shape while growing on a tree). This excludes deviations in appearance due to damage, disease, or other sources of external aesthetic divergence that may rationally raise safety or health concerns (e.g., due to pests or consumer mishandling)."

unbalanced–symmetric, unconventional–prototypical). To improve upon items, iterative refinement through asking for revisions based on definitions and examples of potential sources of measurement error (e.g., saying that some items are double-barreled) is useful.

3.2.2 Developing manipulations

GenAI offers researchers a powerful tool for developing experimental manipulations, allowing for the creation of context-rich stimuli that enhance ecological validity. By leveraging LLMs for text-based manipulations and GenAI systems for multimodal AI for visual stimuli, researchers can generate immersive scenarios and images that more closely resemble real-world experiences. However, using GenAI for experimental manipulations introduces methodological challenges. Just as poorly defined measurement items can lead to construct validity issues in survey research, inadequately controlled manipulations can create confounds that undermine causal inference. Because GenAI models generate texts based on linguistic plausibility, they can introduce systematic differences between experimental conditions beyond the intended manipulation. To illustrate, we prompted GPT-4o to generate vignettes for a study manipulating the visual attractiveness of an apple.[4] The model produced the following descriptions:

> Visually Attractive Condition: "Imagine you are shopping for produce at a grocery store. As you walk through the vibrant produce section, your eyes are drawn to a bin of apples. Among them, you notice a particular apple that catches your attention. This apple has a flawless, shiny red surface that gleams under the store lights, as if it has just been polished. It is perfectly round and without any blemishes or bruises. The bright color and smooth texture make it look fresh, juicy, and irresistible. You pick it up, feeling its firm, cool surface in your hand."
>
> Visually Unattractive Condition: "Imagine you are shopping for produce at a grocery store. As you walk through the produce section, you come across a bin of apples. Among them, you notice a particular apple that stands out for the wrong reasons. This apple has a dull, uneven surface with patches of discoloration and spots of brown. It is misshapen, with a slightly lumpy appearance, and you can see small blemishes and bruises scattered across its skin. The rough texture and faded color make it look less appealing and not as fresh. You pick it up, feeling its uneven, slightly soft surface in your hand."

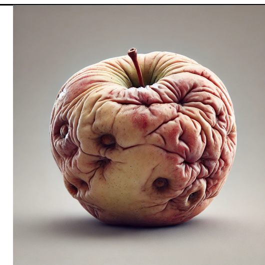
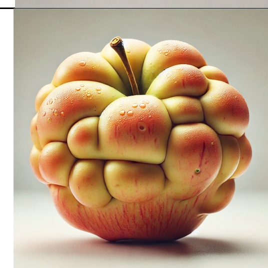
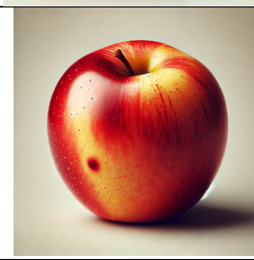
[4] The exact prompt was: "I want to run an experiment where I tell people to imagine shopping for produce at a grocery store. I want everyone to imagine finding an apple. Participants in the visually attractive condition should imagine a visually attractive apple and participants in the visually unattractive condition should imagine a visually unattractive apple. Can you help me create the vignettes for these two conditions?"

While these two vignettes are largely parallel, one can notice that the visually attractive condition uses “flawless,” “gleams,” and “irresistible,” while the unattractive condition employs less vivid and explicitly negative terms such as “blemishes,” “bruises,” and “not as fresh.” This effectively conflates visual attractiveness with freshness and perishability.

One of the most valuable capabilities of modern GenAI systems is their ability to integrate LLMs with external multimodal tools, enabling researchers to generate both text-based and visual stimuli. However, in the absence of precise definitions, GenAI systems may introduce visual confounds, just as they do in text-based manipulations (Sisodia, Burnap, and Kumar 2024). To illustrate, we used ChatGPT GPT-4o, in which the core LLM can access an external DALL-E model to generate images. We prompted the model to generate a visually attractive apple and a visually unattractive apple in two separate images. It created the images in the first row of Table 5. The issue with this initial attempt was that the LLM associated visual unattractiveness with spoilage, damage, and discoloration, rather than shape irregularity alone. While adding the context from footnote 3 does improve the results in terms of avoiding showing damage (see row 2), it led to an unrealistic unattractive apple.

To understand why this occurred and improve the results, it is important to note that LLMs in chat-based GenAI systems do not generate images. Instead, they generate commands for external services, which are rendered in the chat user interface. By requesting the exact commands used by the LLM, we were able to manually refine them to remove unwanted confounds and focus strictly on shape asymmetry while keeping freshness and realism constant for the unattractive apple (see third row in Table 5). Thus, while GenAI can serve as a good partner for the researcher in exploring possible manipulation designs, the researcher should be as specific as possible in describing the manipulation and then iteratively modify the prompts.

Table 5 – Illustration of Confounds in Image Generation via Chat

Prompt approach	Attractive	Unattractive
"Create a visually attractive and a visually unattractive apple in two separate images."		
Adding to row 1 the definition and context in footnote 3		
Asking GenAI to generate images, with a more precise prompt for the LLM[5]		

3.2.3 Study design best practices.

GenAI can be a great brainstorming partner in designing a study, particularly in the development of measures and experimental manipulations. However, because GenAI systems generate content based on probabilistic token generation and do not reflect conceptual understanding, we provide steps through which researchers can maintain methodological rigor.

When the goal is to generate measures or manipulations that map to a predetermined theoretical construct, we recommend that researchers begin by clearly defining the focal

[5] e.g., unattractive: "A high-resolution, close-up photograph of a single red apple on a plain white background, taken at eye level. The apple has a very asymmetrical shape and is atypical. It features a mix of red and orange-yellow tones, with one large area near the middle where the red pigmentation fades into strong natural discoloration with less polish. Small, scattered discoloration dots are visible across the skin, giving it an organic, naturally imperfect look. The surface is smooth with very visible texture variations, but the apple remains fresh and undamaged. The lighting is soft and even, emphasizing its natural imperfections without making it appear bruised or spoiled."

construct in their prompt. Simply including the name of a construct (e.g., visual attractiveness) is not enough as it does not ensure that the generated content will capture the researcher's intended meaning. Departures from common academic meaning is likely, as GenAI systems rely upon vast training corpora that include both academic and non-academic text, and may conflate the focal construct with closely related but theoretically distinct concepts (e.g., freshness or cleanliness). Fortunately, GenAI's adaptability means that the researchers can improve the potential validity and usefulness by specifying their requirements more precisely. Beyond including clear definitions, specifying which constructs should be distinct (e.g., mediators or downstream consequences) can help the generation of items that avoid overlap. Nevertheless, researchers should not assume that the use of GenAI in item generation or manipulation development absolves them of the responsibility to demonstrate validity.

Researchers can also experiment with different prompting strategies to further improve the quality and consistency of generated content. For example, when designing experimental manipulations, prompting the model to generate both conditions (e.g., high vs. low empathy; attractive vs. unattractive product) in the same prompt can encourage parallel structure and reduce the risk of unintended confounds. Alternatively, generating each condition in a separate prompt may allow for more tailored elaboration, albeit with greater risk of asymmetry. Researchers can also increase control by explicitly stating that the content is for use in an experiment, like priming the model to prioritize consistency across conditions and to hold constant any information not intended to vary. These types of strategic prompts can help guide the model's output toward greater rigor and better alignment with the intended design.

While the variability in generation can lead to challenges for scale development and creation of manipulations, it is important to highlight that it can be particularly valuable during earlier stages of the research process. One particularly useful ability is to generate dozens of variations on how a construct might be operationalized, and thus provide researchers with a rich space for theoretical exploration and refinement. This can open the door to more systematic forms of stimulus sampling, such as Mix-and-Match (Simonsohn et al., 2025), which aims to ensure confound management and design transparency when testing conceptual variables through diverse instantiations. Additionally, by experimenting with the prompts and responses (e.g., alternative definitions, examples), researchers can use GenAI as a partner in refining their construct development. Moreover, it is also possible to bring the model's attention to scaling principles (e.g., avoiding double-barreled questions), or things to hold constant in scenarios.

3.3 Stage 3: Study Administration

GenAI is opening up exciting new possibilities for how researchers can administer studies, particularly in the way conversations are used to collect data and deliver experimental treatments. Traditionally, surveys and experimental manipulations have predominantly been administered through static and pre-scripted designs which can either limit adaptability or constrain scalability to the sample sizes often required in experimental and survey-based research. GenAI can now enable dynamic, responsive interactions that fluidly respond to participant input. In other words, GenAI can act as a virtual research assistant, engaging participants in naturalistic dialog. However, these capabilities come with caveats. These models' associative, pattern-based understanding means that they can deviate from instructions.

To illustrate, we conducted a study using Garvey and Blanchard's (2025) toolkit, which embeds a GPT-powered chatbot within Qualtrics.[6] Five hundred and three CloudResearch

[6] Garvey and Blanchard's system uses a chat user interface (built with JavaScript and CSS within Qualtrics) and a serverless function that communicates securely with the ChatGPT API. After participants input text, the system updates the conversation history and forwards it to the LLM, before retrieving responses and displaying them.

Connect participants engaged in an AI-facilitated conversation about their fresh produce purchasing decisions. After providing informed consent, participants were introduced to a live chat embedded within the Qualtrics interface. The AI interviewer initiated the conversation with a neutral, open-ended question, prompting participants to describe their approach to selecting fresh produce at a grocery store. The interview proceeded dynamically, with real-time follow-up questions tailored to participant responses. A sample interview is in Figure 2.

Figure 2 – Sample Semi-Structured Interview Conducted Using GPT-powered Agent

assistant: Hello. Nice to meet you and thank you for participating in this interview. To start, can you tell me about how you usually approach buying fresh produce at a grocery store?
user: Hi, I normally go and pick out what I like, making sure it doesn't have defects.
assistant: That makes sense. When you're selecting produce and checking for defects, what specific things do you look for or avoid?
user: Brown spots, dents, mold, etc
assistant: Got it. Are there any particular types of produce where you're especially careful about these defects, or is it a general practice for all the produce you buy?
user: I would say a general practice
assistant: Understood. When deciding which produce to buy, do factors like price or organic labels play a big role in your choice, or is the appearance the primary concern?
user: Mostly appearance, only price if it's too high
assistant: Thanks for sharing that. How do you feel about purchasing unattractive produce, specifically, produce that appears irregular or oddly shaped but is edible and safe to eat?
user: I would likely not purchase it
assistant: Interesting to hear. Could you explain a bit more about why you might decide not to purchase unattractive produce? Is it more about perception, or are there other considerations?
user: I would feel it isn't safe and would be turned off by it
assistant: Thanks for explaining. When it comes to ensuring the produce is safe and of good quality, do you rely on any particular strategies, like shopping at specific stores or looking for certain quality assurances?
user: Generally no
assistant: Alright. Lastly, when you shop for fresh produce, do you plan ahead and have a list, or do you tend to make decisions spontaneously once you are in the store?
user: I normally would have a list but may buy something spontaneously also
assistant: Thank you for sharing your approach with me. That concludes our interview about your decision-making process when purchasing fresh produce at a grocery store. You can now proceed with the study.

At this point, it is important to note that the system's behavior was influenced by instructions designed to simulate the role of a neutral research interviewer. Because LLMs are trained on a large but non-exhaustive corpora, we must direct the model's attention toward our use case: interview about purchasing visually unattractive produce. We do this via a prompt:

> ChatGPT, I am the researcher and you are about to enter into a conversation with a user who is a participant in my experiment. The user will just have read 'Hello. Nice to meet you and thank you for participating in this interview. To start, can you tell me about how you usually approach buying fresh produce at a grocery store?' and their first message will likely be related to these instructions. Your goal is to conduct an

> interview about consumer decision making processes related to purchasing fresh produce at a grocery store. You should focus solely on asking questions about the user's decision making process when purchasing produce. Your questions should flow naturally and directly relate to the user's responses. The session should collect exactly 8 inputs from the user before you tell the user that the interview is over and they can proceed with the study. Toward the middle of the interview, ask the participant how they feel about purchasing visually unattractive produce (explicitly describe this as produce that appears irregular or oddly shaped but is edible and safe to eat). Based on the participant's response to this question, ask follow-up questions to probe why the participant does or does not prefer to purchase visually unattractive produce.

Each component of the initial prompt serves a critical function in shaping the behavior of the GenAI interviewer. The opening sentence ("ChatGPT, I am the researcher and you are about to enter into a conversation…") establishes the experimental context and clarifies the AI's role, ensuring it behaves as a research assistant rather than a general-purpose chatbot. The reference to the participant's first message ("The user will just have read…") helps orient the AI to the conversational timeline and ensures coherent continuity. The explicit instruction ("Your goal is to conduct an interview…") focuses the AI's attention on a specific objective, constraining its otherwise general-purpose capabilities. Directives such as "focus solely on asking questions" and "questions should flow naturally…" help maintain a conversational tone while reinforcing topical relevance and conversational alignment. The specification ("collect exactly 8 inputs from the user…") is useful for expecting standardization and knowing when to deliver the question about visually unattractive produce. Finally, the instruction to introduce and probe the topic midway through the interaction provides a consistent manipulation across interviews, while allowing a follow-up that maintains the naturalistic dialogue.

While the prompt provides essential guidance for shaping the model's behavior, researchers should be aware that the model may still deviate from its original instructions . As discussed in Section 2, these systems generate responses based on probabilistic associations rather than conceptual understanding. Consequently, even well-structured prompts function more like strong suggestions than rigid instructions: models may still ignore or drift from the intended task, especially during extended interactions. For example, if a participant wants to avoid discussing produce and instead unexpectedly states, "I like to shop for electronics," the model may follow this tangent rather than redirecting to produce-related decision-making. These deviations are not failures of logic but reflections of the LLM's lack of persistent memory.

Moreover, it's important to recall that LLMs generate responses probabilistically. This means that, as with human interviewers, even when two participants write the same prompt, the LLM might generate two different responses, which in turn, might create conversational drift. To illustrate, when we generated the same opening text in two surveys ("Hi, I normally go and pick out what I like, making sure it doesn't have defects."), we obtained two different responses:

1. "When you're selecting produce, are there specific qualities or characteristics you look for to determine if something is free of defects?"
2. "Thank you for sharing that! When you say you make sure it doesn't have defects, what specific traits or qualities do you look for in fresh produce?"

Although both responses align with the study's purpose, the first emphasizes defects, while the second shifts focus to broader freshness cues. Over multiple exchanges, such subtle variations may accumulate, leading to divergent experiences across participants.

3.3.2 Best practices for administering a study using generative AI.

Like using human or lab assistants to conduct studies, using GenAI requires careful design of study protocol, monitoring, and logging. First, research should log complete transcripts of all AI-user exchanges. In the training and refining stage, examining conversations can help researchers tweak the model instructions and check for edge cases (for example, how does the model respond when participants veer off topic, or say something inappropriate). In later stages of the research, capturing the full context window becomes crucial for replicability.

Researchers should also keep a record of all AI-related model specifications. This includes the exact model version used (e.g., GPT-4o), system parameters (e.g., temperature, seed), prompt texts (initialization and reinforcement), constraints on the number of interaction rounds or time of interaction if applicable, and participant exclusion rules related to AI performance (e.g., if the AI fails to respond or breaks character) or technical issues (e.g., failure interactions due to network connectivity). We also recommend that researchers include post-session survey items asking participants to report any technical difficulties, confusion, or irregularities experienced during the conversation. These issues may become more pronounced during longer exchanges with the model because of its limited persistent memory and context-window constraints (Section 2.2).

In other applications of the GenAI's ability to act in open conversations with participants, researchers may want to use it like a confederate, to administer experimental manipulations (such as varying tone, empathy, assertiveness, or other psychologically relevant constructs; see Garvey and Blanchard 2025). In these instances, researchers must not assume that the intended manipulation is always faithfully realized, and should therefore test their prompts in the same way that they would for any other manipulation (i.e., pre-testing and manipulation checks).

Several additional best practices can further enhance the quality and reliability of AI-mediated study administration. For example, reinforcement prompts (i.e., brief hidden injections passed to the LLM during the conversation) can help guide the AI back to its intended role or behavior when deviations occur (Garvey and Blanchard 2025). For example, take our prior example of a participant that tries to go off-topic by stating, *"I like to shop for electronics."* To help further mitigate this risk, one can add a reinforcement prompt to sentences submitted by participants (e.g., "*Keep the interview on the topic of the user's decision-making process when purchasing produce at a grocery store.")* When submitted to the system, the user's original message and the reinforcement prompt are concatenated into a single input such that the model receives, along with distinct roles interpretable by the LLM: *"[user]I like to shop for electronics.*

[system]Keep the interview on the topic…" That way, researchers improve the odds of remaining focused while the participant is not aware of the intervention.

Finally, researchers must take active steps to protect participant autonomy and privacy when administering studies through GenAI systems. Informed consent should clearly describe the nature of the interaction and caution participants against disclosing sensitive or personally identifiable information during open-ended responses. Just as importantly, researchers must ensure that GenAI services used do not retain the right to use submissions for model training.

3.4 Stage 4: Data Analysis & Interpretation

In the final stage of the research process, researchers analyze and interpret data to test hypotheses and extract insights. This often involves structuring datasets, coding qualitative responses, and conducting statistical analyses, tasks that traditionally require manual effort and technical expertise. GenAI systems offer valuable assistance by generating executable code, automating portions of data processing, and helping to streamline and interpret complex analyses. They can help novice users to close the analytics gap by assisting novice users in programming or sophisticated data analysis. This section explores how GenAI can assist in data analysis, focusing on statistical computations and qualitative coding. Using data from the participants who engaged in the study described in Section 3.3, we illustrate both the benefits and limitations, and highlight trade-offs between automation, interpretability, and rigor.

3.4.1 Interacting and Running Code on Data Files

One of the most useful applications of GenAI system in the business world is in helping programmers to code. As such, these tools could help researchers analyze their data, whether for exploratory analysis purposes or for more sophisticated analysis. As discussed in Section 2.2, LLMs alone cannot directly handle file uploads or execute computations. While LLMs can process text-based prompts, they do not inherently interact with structured data files such as CSV, Excel, or JSON files. To illustrate, we asked OpenAI's API (which leverages only the LLM and not the full GenAI system) to "Conduct a one-sample t-test and report its t-statistic and p-value, against a null hypothesis that the mean age is equal to 42. The values are: [all values copy-pasted]." The LLM generated-response outlined the steps for conducting a t-test, reported the mean (42.70), standard deviation (13.02), number of observations (341), t-statistic (1.12) and p-value (0.263). However, these statistics were incorrect because the LLM did not actually compute the values: the output merely mimicked the format of a real analysis (the correct values were 40.89, 13.90, 503, -1.78, 0.075, respectively).

While standalone LLMs cannot perform calculations, modern GenAI systems integrate computational tools that allow them to process structured data and execute real code. As illustrated in Figure 1, these systems include input processors and document handlers that detect when a request involves structured data analysis. Instead of attempting to approximate statistical results, the LLM generates executable Python code, which is then processed in a dedicated code execution environment. For instance, using the same prompt in OpenAI's chatbot using ChatGPT 4o generated this Python script (e.g., this is displayed by clicking the " in ChatGPT):

```
import pandas as pd
from scipy import stats
# Load the dataset (handled by the system's document processor)
df = pd.read_csv("uploaded_file.csv")
# Extract the age column and drop missing values
age_data = df['age'].dropna().astype(float)
# Define the null hypothesis mean
mu = 42
# Conduct the one-sample t-test
t_stat, p_value = stats.ttest_1samp(age_data, mu)
# Print results
t_stat, p_value
```

Once the computation is complete, the results are returned to the researcher. In our example, the system correctly calculated all the parameters. This hybrid approach allows researchers to interact naturally with the AI while ensuring that statistical analyses are conducted rigorously. It also enables dynamic and flexible analysis, as the LLM can refine, modify, or generate additional code based on researcher feedback (e.g., exclude participants who fail an attention check). However, researchers must carefully validate the tests performed. For instance, the Python function stats.ttest_1samp(age_data, mu) defaults to a two-tailed test, which may not align with a study's specific needs. Additionally, unlike conversations with LLMs, which can be saved, data uploads and computations are retained only during the session (often an hour).

3.4.2 Coding and Labeling Data

The scalability of GenAI systems to process large amounts of data and do it in a repeated manner makes them a valuable tool to code or label both structured and unstructured data. GenAI offers multiple ways to code complete such tasks, each with trade-offs in researcher control, contextual accuracy, and reproducibility. The most straightforward but least structured approach is copy-pasting multiple textual responses to a chat-based GenAI system and requesting coded outputs.[7] While the ease of this method may be particularly tempting, it is severely limited in its ability to provide valid and reliable coding of responses. First, recall that context-size is limited such that we could only submit 40 conversations at a time (ChatGPT 4o). Second, as an LLM does not distinguish between elements in the context, we have no guarantee that one's data is coded independently of the others. We do not recommend this approach.[8]

---

[7] For example, "Your task is to act as an independent coder for a study. In that study, participants chatted with an AI interviewer about their preferences regarding fresh produce. Please analyze each of the following conversations to give me, on a score of 0 to 9, the extent to which the participant (denoted as user) focuses on risk (i.e., references to dangers, concerns, things to avoid). For each conversation, return only the response as a string in the format [X] (e.g., [2] for 2) without any additional explanation or text."

[8] Moreover, while one may be tempted to ask the chatbot to code responses one at a time, or to start fresh, recall that the information previously submitted to the LLM remains part of the models' input sequence. A better naïve chat-based approach would be to copy a single textual input to code, along with coding instructions, each time in a completely new conversation with the LLM. However, this approach can be quite taxing.

Rather, we recommend that researchers use one of two approaches. The first is a sequential API-based approach, in which a computer script sequentially submits each participant's response (along with standardized instructions) to the API and captures the value returned. The second is a file-upload approach with code execution, where a structured dataset (e.g., CSV with one participant response to code per row) is uploaded to a chat-based GenAI system, along with instructions. The GenAI generates coding rules, which can be saved. However, the two approaches lead to very different by which the LLM processes the data.

To illustrate the implications and differences between these approaches, we used the conversations from the produce purchase interviews described in Section 3.3.1, and asked the model to create a measure of the extent to which participants referenced risk-related concerns when purchasing fresh produce. For both GenAI approaches, the standardized instructions were:

> Your task is to act as an independent coder for a study. In that study, participants chatted with an AI interviewer about their preferences regarding fresh produce. Please analyze the following conversation to give me, on a score of 0 to 9, the extent to which the participant (denoted as user) focuses on risk (i.e., references to dangers, concerns, things to avoid). Return only the response as a string in the format [X] (e.g., [2] for 2) without any additional explanation or text. Here's the conversation:

For the API approach, we used one coder to process each conversation with GPT-4o with temperature of 0.7. Doing so ensures that each response is processed independently, much like assigning responses to separate human coders. Moreover, this fully leverages the LLM's ability to process contextual meaning while controlling independence in the coding of each response.

When we uploaded a dataset to a GenAI system with an active code execution environment, our request triggered the GenAI system to rely on the LLM to create a custom Python function for categorizing responses based on a custom dictionary before returning a coded dataset. Specifically, the system inspected the responses, created a Python function for a custom dictionary using 22 keywords culminating with a scaled response between 0 and 9. We could then download the coded dataset, inspect the codes, and save them.

It is important to highlight some key differences between these two approaches and a predetermined dictionary such as LIWC. Consider for example, the LIWC-15 dictionary, which includes 102 terms (e.g., “abstain”, “bad”, “stops” and “trusts”) and provides a standardized but rigid approach to quantifying risk-related language. However, because LIWC relies on a predefined lexicon, it could only detect risk when participants use terms explicitly included in its word list. If participants described risk concerns in non-traditional ways, such as referring to unattractive produce as “weird” or “dirty” rather than “unsafe” or “risky”, LIWC will fail to capture these references. Additionally, dictionaries may fail to adapt to domain-specific language. While “short” implies risk in a financial context, it does not in a fresh produce context.

Dictionaries also fail to capture the context and the relationships among words, which is one of the benefits at the heart of LLM methods (see Section 2). The API-based approach offers a more adaptive interpretation of risk language by processing each response independently while leveraging the LLM’s contextual reasoning and training data (Rathje et al. 2024). This can allow the coding to capture implicit concerns about food safety, even when participants used indirect language to express worry about unattractive produce. However, because LLMs generate responses probabilistically, there will be some response-level variability in scoring. A participant’s risk score will necessarily fluctuate across API calls, depending on settings such as temperature. Moreover, we cannot know *why* responses are given a score.

The GenAI chat-based file-upload approach with code execution takes a different strategy, as doing so triggers the LLM to generate a custom dictionary of risk-related terms before using Python to apply it systematically to the dataset. Examining the Python code produced by the LLM revealed that it identified food safety concerns (e.g., “spoiled,” “rotten,” “expired,” “mold,” “contaminated”), health-related risks (e.g., “illness,” “food poisoning,” "disease," "infection," "toxins"), and general warnings (e.g., "unsafe," "harm," "risk," "concern," "worry," "avoid," "pesticides," "chemicals"). However, because LLMs are probabilistic, the generated dictionary differs across sessions. While convenient and transparent, this approach, being dictionary-based, does not fully leverage the ability of LLMs.

3.4.3 Data analysis and coding best practices.

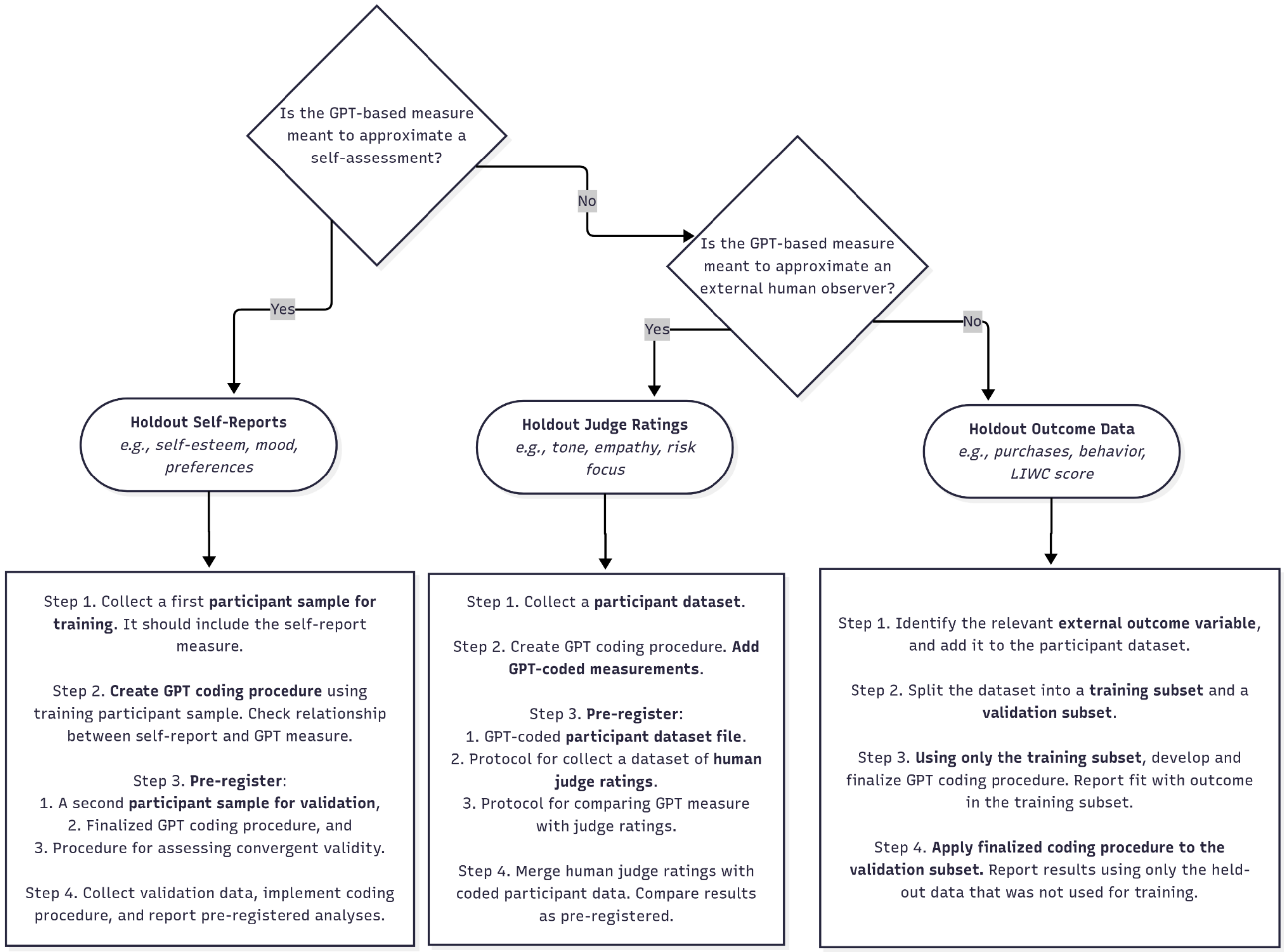
A GenAI system (particularly one with an integrated code environment and the ability to process structured files) can serve as a powerful and accessible tool for exploratory data analysis. Beyond the examples discussed earlier, such systems are especially useful for quickly assessing data integrity, detecting outliers or anomalies, and identifying trends or summary statistics. These are critical steps that are too often overlooked in practice. However, it is important to recognize the limitations of these systems. LLMs are inherently probabilistic, which constrains their ability to produce accurate, transparent, and reproducible outputs. For this reason, we do not recommend conducting confirmatory or inferential analyses directly within GenAI platforms. Instead, researchers should extract any generated code (even when executed successfully within the platform) and re-run the analyses in a dedicated tool such as R, Python, SPSS, or STATA.

When using GenAI to code participant responses, several best practices can improve output quality. One such strategy is the use of structured prompting techniques. For instance, Li et al. (2024) demonstrate that Role-Task-Format (RTF) prompting significantly improves the reliability and reduces variability in LLM-generated classifications. Likewise, few-shot prompting, which provides the model with labeled human examples before coding new responses, can improve alignment between AI and human judgments (Yoo et al. 2025). These approaches emphasize the importance of clear, consistent, and well-structured prompt design.

Yet, as the popularity of using GenAI for open-ended text coding grows, some researchers may assume that a well-written prompt is sufficient to guarantee a valid measure. This assumption is risky. Even if a prompt appears face-valid to a human reader, that does not ensure that the resulting GPT-generated measure faithfully captures the intended construct. Like any other measurement instrument, GenAI-derived measures must be empirically validated—not just judged by whether the prompts used seem reasonable. This concern is particularly acute because GPT-based coding introduces a wide range of researcher degrees of freedom. The same input can yield different outputs depending on how the prompt is phrased, which examples are included, what temperature is set, or how the output is post-processed. Worse, because GenAI systems can re-code entire datasets in seconds, researchers may be tempted (consciously or not) to rerun and tweak until the output aligns with their expectations. This undermines the credibility of any observed effects and makes post hoc validation difficult to evaluate. The central challenge, then, is not only about what the model is capturing but also about procedure: how we ensure that the measure is developed and evaluated without undue degrees of freedom.

To select an appropriate validation strategy, researchers must first clarify what the GenAI-coded measure is intended to reflect. That is, what kind of “ground truth” is the model being asked to approximate? This is not a trivial question. Some GPT-coded measures aim to capture a participant’s internal state, such as their beliefs, emotions, or stated preferences. Others are designed to reflect how an objective, third-party observer might interpret the participant’s text, for example, inferring tone, risk aversion, or expressiveness. Still others are constructed to predict an objective behavioral outcome that exists independently of either the participant or an observer (such as purchase behavior or complaint rates). Next, we outline three validation strategies tailored to the type of construct to validate (see Figure 3 for a guide).

Figure 3 - Validation Approach for GPT Measurement by Construct Type



When the goal is to approximate the participant's internal perspective, we recommend a two-stage strategy using **holdout self-reports**. First, researchers collect a training sample in which participants provide both the data to use for GenAI-coded measurement (e.g., a conversation about purchasing produce) and the relevant self-reports that are meant to be the ground truth (e.g., a self-esteem psychometric scale; a stated preference). Using in the training sample, the researcher can iteratively develop a coding procedure (i.e., refine prompts, select model parameters) and test predictive validity against the self-report. Once the procedure is finalized, we recommend pre-registering the following: 1) the completed coding protocol, 2) the collection of a second independent sample of participants. This second independent sample is then exclusively for confirmatory validation of the predictive validity. For an example, see https://aspredicted.org/yc2y-z27k.pdf

It is important to explain why we do not recommend validating GenAI-coded measures of internal constructs using a strategy in which pre-registration involves a single sample split into training and validation. For any pre-registration to support confirmatory inference, it must include: (1) a dataset of human responses that has not yet been collected, and (2) a finalized analysis procedure. A single-sample design presents a difficult tradeoff: researchers must either pre-register a GenAI coding protocol *before* seeing any data (preventing any opportunity for development) or pre-register the *process* by which they will develop the coding procedure. The latter approach, however, opens the door to researcher degrees of freedom, as the protocol is shaped in response to data that has already been collected.

When the GPT-coded measure is intended to reflect how an independent observer would interpret the participant's response (e.g., the tone of their writing; how risk averse they *seem*), the appropriate validation strategy is to gather **holdout judge ratings**. Here, the GenAI coding procedure is first developed using a dataset of participant responses, and then frozen. Researchers pre-register both the coded dataset (i.e., with GenAI's scores added) and the protocol for a separate group of human judges to independently evaluate the inputs. The comparison between GenAI and human ratings is then conducted exactly as planned, without modifying the coding procedure or the dataset maintaining the GenAI-generated coding. This strategy mirrors best practices in content validation where the coding protocol is finalized before judges' data are collected. See our example at https://aspredicted.org/3mzv-d452.pdf

Finally, when the purpose of the GenAI measure is to predict an outcome that already exists or will soon be available (e.g., the number of visually unattractive produce purchases from scanner panel data or coded receipts), we recommend a validation strategy based on a **holdout outcome**. Researchers begin by identifying the relevant outcome variable and integrating it into the dataset. Specifically, researchers should begin by splitting the sample into training and validation subsets before any outcome data is added. Outcome data can then be merged with the training subset and used to develop the GenAI coding procedure (e.g., refining prompts or selecting model parameters) to improve prediction in the training sample. Once the procedure is finalized, it is applied to the validation subset, whose outcome data can now be added and used. Predictive performance is then evaluated using only this held-out portion of the data.

Because the outcome data in this context often already exists (e.g., from scanner panels or coded receipts), formal pre-registration as required by platforms like AsPredicted.org, which mandate pre-specification before any data collection or analysis, may not be appropriate. However, the approach we recommend follows the same spirit: it minimizes researcher degrees of freedom by clearly separating model development from evaluation, and by ensuring that outcome data for the validation set remains genuinely unseen during development. While it cannot qualify as a pre-registration in the strictest sense, this strategy supports more credible, transparent, and prospectively meaningful validation when working with secondary data.

Finally, it is also important to note that while predictive validity can serve as one component of a validation strategy for such behavioral measures, just because a GenAI-coded measure successfully predicts an outcome does not necessarily mean it is measuring the intended construct. As with traditional measurement development, we must go beyond predictive performance to assess how well the measure represents the theoretical construct of interest.

Whichever strategy they use, we recognize that it is not always practical to collect large-scale participant samples for both the training and validation phases. Similarly, for holdout judge ratings, collecting human judgments at scale can also be resource-intensive, such that it may be sufficient to validate the measure against a randomly selected subset of participant responses against judges' ratings. Similarly, when validating against external outcome data, it is not always possible to obtain outcome measures for the data to be coded. One of the key benefits of using GenAI for measurement is their ability to scale.

4. Principles for Responsible Use of GenAI in the Primary Data Research Process

The successful integration of GenAI into research demands more than just understanding how these systems work (Section 2) or how well they perform at different research tasks (Sections 3.1-3.4). Beyond the summary set of guidelines and recommended best practices we outline in Table 6, we now turn to key general elements to consider as the technology continues to evolve.

4.1 Maintain Full Responsibility for Accuracy and Validity in GenAI-Assisted Research

Scientific integrity depends on two foundational obligations: ensuring the accuracy of information presented and the validity of constructs measured or manipulated. Yet, while GenAI systems can assist with both tasks, they do not relieve researchers of their responsibilities.

First, researchers are fully accountable for the factual accuracy of any literature reviews, summaries, statistical analyses, or descriptive claims generated with GenAI assistance. As discussed throughout this paper, LLM-based systems are not fact-checkers. They generate responses based on statistical likelihoods, not source validation. As a result, they can fabricate citations (Section 3.1.1), misrepresent study findings (Section 3.1.2), or produce incorrect statistical outputs that may appear plausible but are fundamentally flawed (Section 3.4.1). Researchers must therefore verify all outputs, cross-check references, and test statistical code before including any GenAI-assisted content in research reports or publications.

Table 6: Best practices & Guidelines

Literature Review	*Best Practices* - Use LLMs for early-stage ideation, exploratory synthesis, and conceptual mapping - Verify GenAI-generated summaries and citations against original sources - Familiarize yourself with knowledge cutoffs and inability to access paywalled content models you used - Periodically re-upload documents and instructions during long sessions to address context window limits - Use follow-up prompts to clarify summaries or revise incorrect extractions *Guidelines* - Do not upload copyrighted text to systems that reserves the right to use data for model training - Take responsibility for inaccuracies that arise
Research Design	*Best Practices* - Clearly define focal construct in prompts to reduce the risk of confounding; also specify which constructs should remain distinct - Use GenAI systems iteratively to refine tone, specificity, and framing across multiple exchanges - Add to prompt examples of principles or rules for item or manipulation development (e.g., avoiding double-barreled items, holding contextual element constant) - Leverage GenAI's variability in output generation as a path to exploration by asking for diversity - Take advantage of GenAI's multimodal capability to maximize external validity through heightened experimental realism *Guidelines* - Do not assume the use GenAI enables reduce standards for measurement validity and reliability - Document the stimuli generation process, including system used, model version, prompt and outputs.
Study Adminis-tration	*Best Practices* - Pre-test GenAI agents extensively; refine prompts and reinforcement strategies to ensure consistent, role-aligned interactions. - Include post-session survey items to ask participants about technical issues or confusion. - Avoid collecting sensitive or personally identifiable information. - Consider prompt reinforcements to prevent the AI from producing off-topic or inappropriate content. *Guidelines* - Log complete transcripts, system prompts, model versions, and configuration settings for all sessions. - Use only GenAI platforms or deployments that prohibit use of data for model training or improvement - Obtain Institutional Review Board (IRB) approval detailing agent behavior and participant protections.
Data Analysis & Interpretation	*Best Practices* - Use GenAI to generate analysis code but run and verify it in a dedicated analytical package (e.g., R, Python, SPSS). - Match your validation approach to your construct being measurement: use self-reports for introspective constructs, judge ratings for interpretive ones, and behavioral data for predictive use cases. - Pre-register your coding validation strategy, including the evaluation plan, sampling and scoring protocols. *Guidelines* - Do not submit participant responses with sensitive or identifiable information to GenAI systems unless approved by IRB and the provider does not retain data for model training. - Do not code responses by copy-pasting those of multiple participants in a GenAI conversation window. - Do not treat GenAI-coded measures as valid without empirical validation: they require the same scrutiny as any new measure.

Second, researchers are equally responsible for the validity of GenAI-generated measures, manipulations, and coding schemes. GenAI systems operate without conceptual understanding and thus are prone to subtle construct drift. A manipulation intended to test "visual unattractiveness," for instance, might inadvertently confound this with moral judgments or social desirability (Section 3.2.1). A survey item prompted to measure "concern about food safety" might instead cue generalized disgust or risk aversion. And coding responses without clear guidance or validation risks capturing adjacent constructs (Section 3.4.2). In each of these cases, it is the researcher's responsibility to ensure accuracy.

4.2 Limit Researcher Flexibility and Prioritize Reproducibility in GenAI-Assisted Research

One of the central lessons from the broader replicability crisis in science is that researcher degrees of freedom (choices about data cleaning, modeling, and interpretation) can inflate false-positive rates and distort scientific inference (Simmons et al., 2011). The integration of GenAI into research workflows introduces new forms of flexibility that must be addressed with care. Even small changes in prompt wording, model parameters, or sampling settings can yield divergent outputs. The ability to generate or code data rapidly at scale compounds this risk, particularly when researchers can choose from multiple iterations or define success post hoc.

To mitigate these risks, researchers should prioritize transparency and reproducibility in all GenAI-assisted workflows. Prompts, full transcripts of interactions, model version numbers, configuration settings (e.g., temperature, top-p), and any coding or decision rules should be recorded and made available for review–particularly when GenAI is used for study administration or open-ended data collection. Researchers should also consider using pre-registration to reduce researchers' degrees of freedom that GenAI tools can introduce, particularly when these tools are used for data collection or response coding. Chat-based interfaces are not recommended for any task requiring auditability or consistency. These environments lack the structure and traceability needed to ensure reproducibility. Researchers should instead rely on API-based or structured workflows that preserve version control and allow all steps to be documented. To support adoption, we include in our repository not only R scripts for submitting and logging GenAI-coded variables via API, but also an add-on for SPSS that enables researchers to do the same without any coding.

4.3 Safeguard Participant Privacy and Respect Intellectual Property in GenAI Workflows

The use of GenAI in research introduces new ethical responsibilities. Not only in safeguarding participant data, but also in how researchers handle proprietary materials and publish academic content. On the participant side, even seemingly innocuous inputs, such as open-ended responses or chat transcripts, can contain personally identifiable information. Uploading such data into commercial or publicly accessible GenAI systems without institutional oversight may inadvertently violate data protection policies or IRB protocols, particularly if the system retains inputs for model training or logs metadata such as user location or session history. Researchers should be proactive in evaluating the platforms they use and prioritize models or tools that explicitly protect participant privacy and restrict data retention.

Researchers must also consider how they handle academic materials. Many GenAI systems retain uploaded documents, such that submitting proprietary instruments or scholarly articles, especially published ones, can violate publisher agreements or licensing terms. Treating such materials as casual inputs to a GenAI system, whether for summarization, item generation, or translation, risks not only legal exposure but also the erosion of scholarly norms. As a research community, we have a collective obligation to protect the intellectual boundaries of others' works. Not just through citation, but in how we share, and reuse it in the age of GenAI.

5. Conclusion: GenAI as a Research Tool, Responsibility as a Researcher Obligation

Generative AI is rapidly reshaping the landscape of academic research. It accelerates idea generation, streamlines study design, facilitates adaptive data collection, and offers new tools for analyzing both structured and unstructured data. But these capabilities come with trade-offs: between flexibility and control, automation and accountability, innovation and reproducibility.

As this paper has shown, GenAI systems differ widely in architecture, behavior, and underlying assumptions. Whether accessed through APIs, browser interfaces, or local installations, each implementation carries distinct implications for participant protection, data security, construct validity, and research transparency. Even subtle design choices, such as how a prompt is phrased, or how a document is uploaded, can introduce substantial variation. Such variations are not minor technical issues; they can create fundamental scientific concerns. The probabilistic nature of GenAI means that researchers must take active steps to verify accuracy, validate constructs, safeguard participant interactions, and respect intellectual property. That responsibility cannot be delegated to the system: it lies with the researchers.

At the same time, we should not treat GenAI as a threat to rigor but as a prompt for renewed methodological reflection. Used responsibly, these systems can support more adaptive, inclusive, and scalable forms of inquiry. But doing so requires more than prompt engineering: it requires documentation, reflection, and open conversation across disciplines. As GenAI capabilities grow, the academic community must evolve alongside them. We must balance innovation with ethical responsibility, and productivity with scientific care. Ultimately, the future of research with GenAI will be shaped not just by what these tools can do, but by the values and vision of the scholars who choose to use them.

References

Arora, Neeraj, Chakraborty, Indranil, and Nishimura, Yuko (2024), “AI-Human Hybrids for Marketing Research: Leveraging LLMs as Collaborators,” *Journal of Marketing*.

Blythe, P.A., Kulis, C., McGraw, A.P., Haenlein, M., Hewett, K., Yoo, K., Wood, S., Morwitz, V.G. and Huber, J., 2025. Comments on “AI and the advent of the cyborg behavioral scientist”. *Journal of Consumer Psychology*, 35(2), 316-328.

Churchill, Gilbert A. (1979), “A Paradigm for Developing Better Measures of Marketing Constructs,” *Journal of Marketing Research*, 16 (1), 64–73.

de Hooge, Ilona E., van Dulm, Eline, and van Trijp, Hans C. M. (2018), “Cosmetic Specifications in the Food Waste Issue: Supply Chain Considerations and Consumer Preferences,” *Journal of Cleaner Production*, 56, 126–139.

Garvey, Aaron and Blanchard, Simon J. (2025), “Generative AI as a Research Confederate: The LUCID Methodological Framework and Toolkit for Human-AI Interactions Research,” Working paper available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5256150

Grewal, Lauren, Hmurovic, Jillian, Lamberton, Cait, and Reczek, Rebecca W. (2019), “The Self-Perception Connection: Why Consumers Devalue Unattractive Produce,” *Journal of Marketing*, 83 (1), 89–107.

Li, Peiya, Castelo, Noah, Katona, Zsolt, and Sarvary, Miklos (2024), “Frontiers: Determining the validity of large language models for automated perceptual analysis,” *Marketing Science*, 43(2), 254-266.

Rathje, Steve, Mirea, Diana M., Sucholutsky, Iaroslav, Marjieh, Roni, Robertson, Caroline E., and Van Bavel, Jay J. (2024), “GPT Is an Effective Tool for Multilingual Psychological Text Analysis,” *Proceedings of the National Academy of Sciences*, 121 (34), e2308950121.

Simmons, Joseph P., Nelson, Leif D., and Simonsohn, Uri (2011), “False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant,” *Psychological Science*, 22 (11), 1359–1366.

Simonsohn, U., Montealegre, A., & Evangelidis, I. (2024). Stimulus Sampling Reimagined: Designing Experiments with Mix-and-Match, Analyzing Results with Stimulus Plots. *Analyzing Results with Stimulus Plots (February 5, 2024)*.

Sisodia, Anupam, Burnap, Alex, and Kumar, V. (2024), “Generative Interpretable Visual Design: Using Disentanglement for Visual Conjoint Analysis,” *Journal of Marketing Research*.

Tomaino, G., Cooke, A. D., & Hoover, J. (2025). AI and the advent of the cyborg behavioral scientist. Journal of Consumer Psychology.

Tully, Sarah M., Longoni, Chiara, and Appel, Gabi (2025), “Lower Artificial Intelligence Literacy Predicts Greater AI Receptivity,” *Journal of Marketing*.

Yoo, K., Haenlein, M., & Hewett, K. (2025). A whole new world, a new fantastic point of view: Charting unexplored territories in consumer research with generative artificial intelligence. *Journal of the Academy of Marketing Science,* 1-37.

Web Appendix A – Sample generated summary table from Grewal et al. (2019) without document upload and RAG

Study	Independent Variable (IV)	Dependent Variable (DV)	Moderator / Mediator	Sample Source & Context	Main Finding
Study 1	Produce attractiveness (manipulated: 'attractive' vs. 'unattractive' images)	Willingness to purchase, perceived quality	None explicitly tested	Online panel or undergraduate sample (lab setting)	Consumers reported lower willingness to purchase and perceived lower quality for unattractive produce
Study 2	Produce attractiveness (manipulated); measured self-perception concerns	Purchase intention, quality evaluation	Mediator: self-perception concerns (feeling incompetent)	Online participant pool; survey with images	Devaluation of unattractive produce is driven by self-perception concerns about competence
Study 3	(1) Produce attractiveness (manipulated), (2) Usage context (manipulated: high vs. low emphasis on presentation)	Willingness to purchase, product evaluation	Moderator: usage context (impressive dish vs. low-visibility context)	Lab experiment; participants saw varied usage contexts (e.g., blended soup vs. fancy dish)	The negative effect of unattractive produce is attenuated when its appearance is less relevant to the usage context
Study 4	(1) Produce attractiveness (manipulated), (2) Functional advantage of 'ugly' produce (highlighted vs. not)	Purchase intention, quality perceptions	Moderator: functional advantage	Online or lab sample; scenario-based experiment	Emphasizing a functional benefit of unattractive produce mitigates self-perception threat and increases purchase likelihood
Study 5	(1) Produce attractiveness (manipulated), (2) Public vs. private consumption scenario	Purchase or choice, attitudes	Moderator: public vs. private context	Online or student sample; controlled experiment	When consumption is public, self-perception threat is heightened, reducing purchase of unattractive produce
Study 6	(1) Produce attractiveness (manipulated), (2) Individual differences in cooking skill (measured)	Willingness to pay (WTP), likelihood of choosing 'ugly' produce	Moderator: self-reported cooking skill; Mediator: perceived self-image threat	Online or lab-based sample; measured cooking skill followed by choice tasks	Devaluation is strongest among those moderately concerned about competence and weaker among very skilled or unconcerned consumers
Study 7 (Field)	Naturally occurring attractiveness (unattractive vs. regular produce) in-store, plus any in-store signage or discount manipulations	Actual purchase behavior (sales data, observed choice)	Potential moderators (e.g., discount, usage signage)	Real-world grocery or market setting; field experiment	Unattractive produce sells less unless a functional benefit or discount is highlighted