

The General Social Survey



Leveraging Large Language Models to Code Open-Ended Responses in the General Social Survey

GSS Methodological Report #150
July 2026

Authors:
Rachel Sparkman
Soubhik Barari
Benjamin Schapiro
René Bautista



SUMMARY

Large language models (LLMs) are transforming survey research by expanding the capacity to process, code, and analyze open-ended data. Automated coding can significantly reduce the time and effort required of human coders, but questions remain about reliability, bias, and implications for data quality. This research addresses these gaps by leveraging the General Social Survey (GSS), which provides nationally representative data and includes a recent innovation: respondents classifying their own open-ended answers. Using this methodology, the GSS can serve as a benchmark for evaluating LLM performance using respondent-provided classifications as well as expert (i.e., human) code as reference standards, also referred to as “ground truth.”

Specifically, this research applies LLM-based coding methods to open-ended responses from the 2024 GSS follow-on questionnaire, focusing on two domains: the most important issues facing America today and respondents’ household composition. These questions differ in complexity. The first elicits nuanced attitudinal responses, whereas the second captures factual information that can be difficult to measure using standard closed-ended question batteries. Respondents were asked to classify their open-ended responses into predefined GSS response categories, enabling direct comparison between the LLM coding versus respondent-provided classification. The dual-input coding design within the GSS follow-on allows us to measure accuracy, consistency, and potential sources of error in automated coding. Moreover, the GSS provides a unique opportunity to evaluate LLM-based coding against traditional human coding approaches while also offering a rich source of contextual information that can inform prompt design.

Overall, this research finds that while human coding and review remain important, LLMs can effectively support the coding of open-ended survey responses. Notably, GPT-5-Mini (which requires substantially fewer computational resources than more advanced reasoning models) achieved coding accuracy comparable to that of expert human coders and the more computationally intensive Claude Sonnet 4.5 model. Both LLMs and human coders achieved higher accuracy when classifying factual responses about household composition than when classifying attitudinal responses concerning the most important issues facing the nation. We find that both the human coder and LLMs approached the attitudinal verbatims similarly, with higher agreement particularly with the GPT models. We also examine differences in coding accuracy across prompting conditions, tradeoffs involving cost and processing time, and implications for data cleaning and survey design.

SIGNIFICANCE OF STUDY

This research contributes to an emerging literature on LLM-assisted coding, grounded in the General Social Survey (GSS), one of the most widely used sources of data on the attitudes and behaviors of the U.S. public. The GSS is a platform that provides demographic information about respondents, their attitudes on a myriad of issues, and decades of attitudinal trends that can enrich prompt engineering. Our research differs from others in that we leverage the GSS at the design stage to collect data that enables a richer evaluation of LLM-assisted coding of open-ended survey responses. Our contribution is threefold.

First, this research evaluates LLM performance within a high-quality, nationally representative survey context rather than on convenience datasets or standalone coding exercises. By leveraging the GSS, the study assesses whether findings from the growing LLM literature generalize to population data collection efforts, while taking advantage of the rich contextual information available within the survey infrastructure.

Second, the study advances methodological operationalizations of “ground truth” in open-ended coding. The 2024 GSS follow-on captures respondents’ open-ended answers, respondents’ own classifications of those answers, and classifications produced by expert coders and LLMs. This design creates a unique opportunity to compare respondent meaning, expert interpretation, and model-based classification within a unified framework. Rather than assuming a single correct classification, this approach allows us to examine how different sources of interpretation converge or diverge when applied to the same respondent-generated text. As a result, the study evaluates not only coding accuracy, but also the broader question of what may constitute valid classification in open-ended survey measurement.

Third, the study contributes to a growing literature on prompt engineering for AI-assisted survey research by evaluating whether survey-specific contextual information improves coding performance. Drawing on respondent demographic characteristics and historical response distributions available within the GSS, we assess whether contextual information commonly used by survey researchers can also enhance LLM-assisted classification. This provides practical evidence on how survey infrastructure and auxiliary data may be incorporated into automated coding workflows.

Taken together, these contributions extend existing research on LLM-assisted coding by evaluating model performance within a nationally representative survey, introducing multiple reference standards for assessing coding quality, and examining the role of contextual information in prompt design. The findings provide practical guidance for integrating open-ended questions, respondent self-coding, expert review, and LLM-assisted coding into future survey workflows and questionnaire design.

BACKGROUND

Large Language Models

Large language models (LLMs) are transforming social science research. Because LLMs are capable of simulating human-like reasoning, judgment, conversation, and behavior, they are well-suited to assist with a range of research tasks, including research design, data collection, data analysis, and scientific reporting (Grossmann et al. 2023; Ziems et al. 2024). LLMs allow for the augmentation of survey research broadly (Rothschild et al. 2025) and, in particular, more efficient coding of open-ended survey responses (Mellon et al. 2024; Buskirk et al. 2025; Rothschild et al. 2026).

Why use LLMs for response coding? LLMs offer several conveniences over alternative machine learning (ML) methods for open-ended response classification: they do not require a large dataset of labelled training data, they are highly inexpensive (in some cases, costing no more than 1¢ to classify a single open-ended response), and they can be conversationally instructed to perform tasks via “prompting”. Although researchers are able to utilize cutting-edge LLMs (e.g., GPT-5.1) via chatbot applications (e.g., ChatGPT) “off the shelf” without the traditional burdens of ML classification, several factors have been shown to impact classification performance and require consideration including: prompting techniques (also called *prompt engineering* or *context engineering*), model selection (particularly the choice of large versus small models), and fine-tuning (whether a model is further trained on task-specific examples of correct outputs). Throughout these decisions, humans remained involved in the coding process and prompt engineering.

What factors influence the performance of LLMs in response coding? Recent research has investigated the downstream impacts of different LLM configurations on the quality of classification outputs. Concerning prompt engineering, two popular approaches are known as *few-shot prompting*, where the LLM is given task instructions, the relevant codebook, and example inputs that belong to each code category (Brown et al. 2020), or *zero-shot prompting*, where the LLM is only provided instructions and the codebook with no guiding examples (Kojima et al. 2022). Increasingly, findings reveal that few-shot prompting improves the performance of open-ended coding relative to zero-shot prompting (Mellon et al. 2024; Von der Heyde et al. 2025).

Other research examines the impacts of the model itself. Von der Heyde et al. (2025) find that, beyond the usage of few-shot prompting, the best success was achieved when using a model fine-tuned on previously coded open-ended survey responses – that is, enhancements to the model led to bigger gains than enhancements to the prompts. Foundational research on “scaling laws” suggests that increasing model size, training data, and compute leads to predictable improvements in performance, typically reflected in lower error and higher accuracy across tasks (Kaplan et al. 2020; Kostina et al. 2025). However, more recent research complicates this narrative, documenting diminishing returns to scale and instances where smaller language models outperform larger so-called “reasoning” models on narrowly defined or simpler classification tasks (Chen et al. 2025; Kumar et al. 2025). This has led to a growing recognition that “bigger is not always better,” and that performance reflects an interaction between model size, data characteristics, and human activities such as prompt design.

Open questions. Despite rapid progress, several important factors in open-ended classification remain underexplored, particularly the inclusion of contextual, task-specific information in prompts. One key gap concerns the incorporation of *respondent-level demographic information* or *question-level metadata* into prompts, mirroring their use as covariates in traditional statistical classification models. Indeed, survey researchers often utilize respondent characteristics, design features, or question context as informative auxiliary variables, whether in imputation, nonresponse adjustment, or models of open-ended text such as topic models (Andridge & Little 2010; Roberts et al. 2014).

A related open question concerns the use of *historical information or past trends in response patterns*, which can be conceptualized as analogous to lagged outcomes in panel or time-series models. In many survey settings, particularly repeated cross-sections such as the General Social Survey (GSS), it is possible to observe stable question wording and track changes in response distributions over time. Prior work in survey research emphasizes the importance of temporal consistency and trend analysis for understanding public opinion and measurement properties (see, for instance, Petty & Krosnick 2014), yet these dynamics are largely absent from current research on LLM-assisted open-ended response classification.

Taken together, these gaps motivate the present study, which leverages a unique design feature to generate high-quality ground-truth data for evaluating classification performance, which we describe next.

The General Social Survey Follow-On Questionnaire

The General Social Survey (GSS) is a nationally representative survey that has been conducted since 1972 to measure the attitudes and opinions of people in the United States (GSS n.d.). Primarily conducted every two years, the GSS added a web-based follow-on questionnaire (also called GSS Next) in 2022 to ask respondents additional questions and field survey question experiments. The follow-on questionnaire of the 2024 GSS asked respondents two open-ended questions: the respondents' household composition (HHTYPE) and the most important issues facing America today (TOPPROB). The GSS provides a unique and novel opportunity to compare traditionally implemented open-ended coding efforts (i.e., manual coding with human labor) with LLMs and has expansive data to test prompts with additional contexts, such as by including demographic information about the respondents who provided the open-ended response, and any past trends of data.

The variable HHTYPE (household type) is a commonly constructed variable from household rostering or from other available questions asking about household composition in the household module of the baseline questionnaire. In the 2024 GSS baseline, the variable was constructed using traditional methods, but the follow-on to the 2024 baseline presented a new set of questions. In the 2024 follow-on questionnaire, respondents were first asked an open-ended question to describe their household: "Please describe your household- how many people live in your household and what is each of their relationship with you and each other." On the next screen of the web instrument, respondents were asked, "Given what you said, which of these matches your household?" and a list of the HHTYPE response categories was provided. Exhibit 1 displays the first screen respondents saw on the left, and the next screen they were shown on the right.

Unlike HHTYPE, the variable TOPPROB (the most important issues facing America) has only been asked of GSS respondents in 2010 and 2021. However, in the 2024 follow-on questionnaire, and similar to HHTYPE, respondents were first asked an open-ended question: "What do you think is the most important issue for America today?" On the next screen of the web instrument, respondents were asked, "Which of the following issues matches your answer to the previous question?" and a list of the TOPPROB response categories was shown. Exhibit 2 displays the first screen respondents saw on the left and the next screen they were shown on the right.

The follow-up questions to both HHTYPE and TOPPROB open-ended questions are a way to validate the respondents' verbatim text by asking them to self-code their open-ended response with the follow-up question in each instance. In this research, we use the follow-up questions HHTYPE1_NEXT and TOPPROB1_NEXT to measure the accuracy of how the human coder and LLMs code the open-ended response that was asked in the first question. For purposes of this research, we refer to these responses as the "ground truth" moving forward.

Exhibit 1. Example of HHTYPE respondent-facing screens from the 2024 GSS follow-on web questionnaire.

HHTYPE	HHTYPE1_NEXT
SAVE & EXIT <i>Please describe your household- how many people live in your household and what is each of their relationship with you and each other. For example, "I live with my spouse and our child" or "I live with my parents and children."</i> BACK NEXT If you experience technical issues, please call [redacted] or email [redacted] for assistance. The General Social Survey 	SAVE & EXIT <i>Given what you said, which of these matches your household:</i> ☐ Married couple, no children in the household ☐ Single parent ☐ Other family relationship (e.g. cousins, grandparents, aunts & uncles), no children in the household ☐ Single adult ☐ Cohabiting couple, no children in the household ☐ Non-family relationship (e.g. friends, coworkers), no children in the household ☐ Married couple, children in the household ☐ Cohabiting couple, children in the household ☐ Other family relationship, children in the household ☐ Non-family relationship, children in the household ☐ None of these BACK NEXT If you experience technical issues, please call [redacted] or email [redacted] for assistance. The General Social Survey 

Note: Variable names HHTYPE and HHTYPE1_NEXT were not shown to respondents. They are included here for ease of interpretation. HHTYPE1_NEXT was shown separately to respondents.

Exhibit 2. Example of TOPPROB respondent-facing screens from the 2024 GSS follow-on web questionnaire.

TOPPROB	TOPPROB1_NEXT
SAVE & EXIT <i>What do you think is the most important issue for America today?</i> BACK NEXT If you experience technical issues, please call [redacted] or email [redacted] for assistance. The General Social Survey 	SAVE & EXIT <i>Which of the following issues matches your answer to the previous question?</i> ☐ Health care ☐ Education ☐ Crime ☐ The environment ☐ Immigration ☐ The economy ☐ Terrorism ☐ Poverty ☐ None of these BACK NEXT If you experience technical issues, please call [redacted] or email [redacted] for assistance. The General Social Survey 

Note: Variable names TOPPROB and TOPPROB1_NEXT were not shown to respondents. They are included here for ease of interpretation. TOPPROB1_NEXT was shown separately to respondents.

METHODS

To investigate the utilization of LLMs to code verbatim respondent answers, we use open-ended HHTYPE (n=920) and TOPPROB (n=923) responses from the 2024 GSS Next follow-on questionnaire, self-administered via the web. Motivated by open questions around the impacts of model type and prompting on LLM coding performance, this study poses the following research questions:

RQ1. Does the performance of response coding differ across different LLM *models*?

RQ1a. Does including *contextual information* in prompts improve response coding performance?

RQ2. Does LLM response coding performance differ between *factual* (household composition) and *opinion* (societal issues) questions?

RQ3. How does the performance of LLM response coding compare to *expert human* response coding?

Analytic Plan

Models. To address our research questions, we evaluate¹ coding outcomes across three specific LLMs of various capacities from small and fast, to larger “reasoning” models: GPT-5-mini, GPT-5, Claude-Sonnet-4.5. These models were chosen for three key reasons. First, we wanted to analyze outcomes across both large and small models. Second, we were limited to models within the NORC secure compute environment, which, as we began the project, included multiple varieties of ChatGPT, Claude, and Microsoft Copilot. We eliminated Copilot from contention based on advice from colleagues as well as the absence of a convenient programmatic pipeline to it. We selected more contemporary models out of concern for the discontinuation of older models, such as ChatGPT4 models, which had public access issues during the planning phase. Ultimately, we selected what we viewed as the standard large and small ChatGPT models (GPT-5 and GPT-5-mini, respectively), as well as the best non-ChatGPT model we had secure access to (Claude-Sonnet-4.5).

Outcomes. To evaluate the quality of the LLM-generated codes, we compare LLM coding results with respondent self-coded data, examining the degree of agreement, and identifying systematic category-level differences. The manual coding is completed by a researcher with over 4 years of experience coding GSS open-ended data, whose coding results we herein refer to as “expert code” in the results below. Additionally, we analyze the accuracy of the coding with respect to the respondent’s own coding, which we refer to as the “self code” and treat as the “ground truth” for our primary analysis.² As a secondary analysis (see Appendix C) we elicit the LLM for a confidence score for each of its predicted response codes.

This research has two sources of “ground truth” to test coding accuracy. First, this research uses the respondent’s own coding, which is captured in GSS variables HHTYPE1_NEXT and TOPPROB1_NEXT, and follows their paired open-ended questions, as “ground truth” and the primary measure to test how accurate the LLMs and expert coder are. Second, the expert code can also serve as a reference point for coding open-ended responses, which we use as “ground truth” in supplemental analyses of the TOPPROB coding.³ This design mimics two different scenarios for coding survey data – relying on respondent self-report, and utilizing expert coding instead of respondent self-report, and allows us to test against both scenarios with limited survey questions. Having multiple forms of “ground

¹ To standardize our evaluation, this analysis employs a programmatic API-based approach. All model prompting workflows and data analysis was conducted in R.

² Though we operationalize this data as ground truth for the purposes of this study, this data is also subject to measurement error (Chew et al. 2026). Previous work has shown that when the self-coding task is presented in the format of a yes/no question, codes suffer from acquiescence bias or an inflation of false positives (Barari et al. 2025).

³ We also conducted this analysis for HHTYPE and have omitted the results (which largely align with TOPPROB) for brevity.

truth” allows us to test the coding accuracy of the expert code and LLMs against the respondent, as well as the coding accuracy of the LLMs against the expert code. In either case, at least one human remains in the loop as the gold standard for each test: the respondent themselves, or the expert reviewer. We note, however, that both sources may still experience different forms of measurement error.

Prompt Conditions. Across each model and question type, we evaluate how prompt design influences LLM response codes by determining whether a second prompt with more contextual information improves the accuracy.

For each set of verbatim responses (HHTYPE and TOPPROB), we compare coding results across two zero-shot prompting strategies: base prompts and contextual prompts. Base prompts consist of concise, direct instructions guiding the LLM to assign the most appropriate category label to a respondent’s open-ended answer (e.g., “Your task is to assign the most appropriate category label from the list below to a respondent’s open-ended answer”). In contrast, contextual prompts provide additional contextual information tailored to each variable to assess whether expanded guidance affects coding decisions. For HHTYPE, we include respondent demographic characteristics such as sex, age, nativity, marital status, household generational composition, and number of children as supplemental information⁴ in the second prompt. For TOPPROB, we provide previous TOPPROB trends from the 2010 and 2021 GSS as supplemental information in the second prompt. Both sets of prompts for HHTYPE and TOPPROB can be found in Appendix A and Appendix B, respectively.

Code Preparation, Prompt Engineering, and Significance Testing

Code Preparation. To prepare the verbatim responses for LLM coding, we first exported the verbatim responses for both HHTYPE and TOPPROB from the raw 2024 GSS follow-on data file. Next, we removed any personal identifying information from the verbatim data sets and kept the GSS respondent ID assigned to each verbatim response. This resulted in 920 HHTYPE and 923 TOPPROB verbatim cases to be coded.

Each set of code for HHTYPE and TOPPROB, aside from the differences in the second prompt design detailed above in the Analytic Plan, has similar coding designs tailored to each LLM. Each set of code relied on the same pipeline to communicate with the various models – importing the open-ends from an Excel file into R, and stripping out any columns that we did not want to allow the LLMs to access; reading in a system prompt; a batch processing script which automatically imposed model, maximum spend, delimiters, batch sizes, and all inputs and outputs. Following model output, pipelines diverged to address each model-specific, batch-specific error, cleaning up poorly formatted output, misaligned columns, and other errors that were localized on a batch level.

We specified relatively large batches for these prompts, relative to other open-ended coding approaches we have encountered – up to 25 cases per batch, with an automatic provision in our R code to re-send a batch if output did not match minimum parameters in terms of output columns. However, this did not prevent models from providing misaligned or other erroneous output. For example, when addressing the other specified codes in TOPPROB, Claude-Sonnet-4.5 would occasionally misalign a batch and output each additional code as its own column, rather than keeping them all in a single column separated by commas. Some model batches produced unspecified numeric codes outside of the bounds of the code frames. However, we allowed these codes to persist, as they represented coding failures that we are interested in accounting for. We believe that smaller batch sizes are likely to reduce these errors in the future, but with the tradeoff of idiosyncratic batch failures occurring more often and requiring more post-processing cleanup.

These workflows and issues both demonstrate the necessity of human oversight for LLM coding of open-ended responses. While we tested several different iterations of the workflow, standardization process, and error

⁴ Specifically, we use GSS variables SEX, AGE, USCITZN, MARITAL, FAMGEN, and CHILDS to represent demographic characteristics for each respondent in this prompt.

correction process, the one that we ultimately deployed involved the most human labor to produce outcomes. Other workflows with fewer standardization steps tended to produce higher batch failure rates, or LLMs failing to code all batches.

Prompt Engineering and Output. Our prompts utilized a standard structure: a single line identifying the LLM's persona for the coding activity, followed by a task specification. Next, we included the verbatim question text and code frame for the LLM to use. Then, we included any additional information – a default instruction to think carefully before assigning numeric responses, as well as other information for Prompt #2. Finally, we ended on a set of classification rules, specifying formatting and expected output independent from other instructions. Please see Appendix A and B for the prompts, which include classification rules.

Each prompt was iterated 3-5 times before producing the final results analyzed in this report, identifying missing rules relating to output and formatting, as well as specifying codes. We utilized a persona-based approach for prompting, instructing each LLM, *"You are an expert survey research coder trained in coding open-ended responses with high precision,"* to set the initial conditions for open-ended coding, but did not include any other persona-based information. Initial prompts did not include any instructions for the standard GSS missing value codes (primarily .s for Skipped), while the final version of the prompts includes instructions for handling these values. These iterations further emphasize the need for human review in the process of working with LLMs – a naive prompting strategy with the same outline of steps (persona, task, verbatim question text, classification rules, and formatting) would produce less accurate and reliable results than our iterated prompts.

Results were generated quickly once prompts were finalized. All three LLMs processed the data in less than 30 minutes per prompt by set of open-ended verbatims. Claude-Sonnet-4.5 was significantly faster than the two ChatGPT models, resolving coding within about 10 minutes for each prompt, while GPT-5-Mini was the slowest, taking more than 25 minutes per prompt. However, Claude-Sonnet-4.5 also required the most error correction in post-processing. Individual batches (25 verbatims) of Claude-Sonnet-4.5 prompts would fail at a rate of 2-4 per prompt, while both ChatGPT models saw fewer than one failure per prompt (1 overall for GPT-5, none for GPT-5-Mini).

Statistical Tests. To test for statistically significant differences in accuracy (i) between pairs of coders and (ii) between prompt conditions, we use the two-sample test for equality of proportions with Yates' continuity correction applied⁵ where the respondent self-coded "ground truth" was treated as "correct" coding response. We conducted these tests both overall per question as well as for individual categories within each question. Given the volume of statistical tests, we adjust p-values for multiple hypothesis testing via the Benjamini-Hochberg (BHq) procedure (Benjamini & Hochberg, 1995). In general, as we show in great detail in the results that follow, we did not surface many statistically significant differences between models and the expert coder or between prompting conditions.

In the next section, we provide specific code frame details and results per HHTYPE and TOPPROB analyses.

⁵ Tests use R's `prop.test(correct = TRUE)`, a continuity-corrected test for differences in independent proportions. Because models were evaluated on the same cases, McNemar's paired test would ordinarily be preferable. However, model-specific failures mean each paired comparison would use a different subset of valid cases, so paired-test accuracies may not match the reported marginal accuracies or be comparable across pairs. We therefore use marginal accuracy comparisons, treating failures as missing at random.

RESULTS

Household Type (HHTYPE) Coding and Results

First, we investigate the coding accuracy of open-ended responses for respondents to describe their household, “Please describe your household- how many people live in your household and what is each of their relationship with you and each other.” We provided the HHTYPE code frame seen in Exhibit 3 in the LLM coding prompts to guide the LLMs on how to assign verbatim responses to household categories. These are the same categories presented in the second household composition question shown to respondents (HHTYPE1_NEXT, see Exhibit 1).

Exhibit 3. HHTYPE Code Frame.

Value Label	Response Category
1	Married couple, no children in the household
2	Single parent
3	Other family relationship (e.g., cousins, grandparents, aunts & uncles), no children in the household
4	Single adult
5	Cohabiting couple, no children in the household
6	Non-family relationship (e.g., friends, coworkers), no children in the household
7	Married couple, children in the household
8	Cohabiting couple, children in the household
9	Other family relationship, children in the household
10	Non-family relationship, children in the household
11	None of these

HHTYPE Coding Results Across Models and Prompts

We use responses to the second question in the HHTYPE series, HHTYPE1_NEXT, as the source of truth for the accurate code as it is the respondents’ self-coded answer to the open-ended question that was asked first. Exhibit 4 reports the overall coding accuracy of expert and LLM code, compared to the respondents’ answers in HHTYPE1_NEXT across Prompt #1 (direct instructions) and Prompt #2 (including demographic background). Please see Appendix A for the detailed language used.

Overall, the human coder produced the highest coding accuracy (81%), though differences in performance with the LLMs in either prompting condition were not statistically significant (either before or after adjustment for multiple comparisons).

Prompt accuracy varies by LLM, but only by a small amount. In general, the smallest model, GPT-5-mini, performs nominally better than the larger models GPT-5 (not significantly) and Claude-Sonnet-4.5 ($p < 0.1$ across both prompt conditions).

Exhibit 4. HHTYPE Overall Accuracy.

Prompt Strategy	Expert Code	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Prompt #1	81%	79%	79%	77%
Prompt #2 (+Change)		80% (+1%)	78% (-1%)	76% (-1%)

Note: Accuracy is determined according to the respondent's self-categorization HHTYPE1_NEXT. The highest accuracy by prompt is indicated in **bold**.

Exhibit 4 also reports on any overall coding improvement from Prompt #1 (direct instructions) to Prompt #2 (including demographic background). Of the LLMs, including demographic information (Prompt #2) marginally improved the accuracy of GPT-5-mini (<1%), but not the larger models GPT-5 and Claude-Sonnet-4.5, which appear to classify less accurately with the inclusion of demographic information of the respondents. Changes in model performance between prompting conditions are not statistically significant.

HHTYPE Coding Results Across Household Categories

Coding accuracy among household categories is highest for open-ended responses describing married couples and single adults across both Prompt #1 (direct instructions) and Prompt #2 (including demographic background) (Exhibit 5 and Exhibit 6). Because the human coder was not influenced or guided by additional information, such as the LLMs, the expert code accuracy results are shown in both Exhibit 5 and Exhibit 6 as a reference, but the prompt manipulation is not relevant to the expert code accuracy results. We also report all errors produced by the LLMs in the last rows of Exhibit 5 and Exhibit 6.

Exhibit 5 reports on the coding accuracy of the expert code and the LLMs across household categories by using only direct instructions in Prompt #1. Households where the respondent is married (accuracy ranging from 83% to 96%), living as a single adult (88% to 93%), or describe their household composition as including non-family members with children present (81% to 90%) have the highest coding accuracy from the expert code and the LLMs. The household categories with the lowest coding accuracy, and the most difficult for the expert code and LLMs, include cohabiting (accuracy ranging from 42% to 47%), other family (12% to 41%), and non-family (33% to 66%) arrangements where children are present, and the “None of these” category (2% to 9%).

Exhibit 5. HHTYPE Prompt #1 Accuracy.

Household Category	Expert Code	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Married, no children	96%	95%	94%	94%
Married, with children	87%	89%	88%	83%
Single parent	70%	59%	64%	62%
Single adult	93%	89%	88%	89%
Cohabiting, no children	69%	72%	68%	68%
Cohabiting, with children	43%	47%	47%	42%
Other Family, no children	67%	58%	50%	50%
Other Family, with children	13%	35%	41%	41%
Non-family, no children	90%	82%	82%	82%
Non-family, with children	33%	33%	67%	67%
None of these	3%	7%	9%	7%
Uncodeable/Error	N/A	0%	0%	0%

Note: Accuracy results are compared to the respondent's self-categorization HHTYPE1_NEXT. The highest accuracy by household category is indicated in **bold**. The “uncodeable/error” row represents the amount of error produced by the LLM.

Exhibit 6 reports on the coding accuracy of the expert code and the LLMs across household categories by using the demographic information of respondents in Prompt #2. Specifically, we include respondent demographic characteristics such as sex, age, nativity, marital status, household generational composition, and number of children as supplemental information in the second prompt to determine if additional demographic contexts improve the coding accuracy and data quality of the simple directive output (Prompt #1).

Exhibit 6. HHTYPE Prompt #2 Accuracy.

Household Category	Expert Code	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Married, no children	96%	96%	94%	91%
Married, with children	87%	89%	87%	81%
Single parent	70%	60%	63%	63%
Single adult	93%	92%	88%	84%
Cohabiting, no children	69%	68%	68%	68%
Cohabiting, with children	43%	47%	45%	41%
Other Family, no children	67%	50%	58%	50%
Other Family, with children	13%	35%	29%	41%
Non-family, no children	90%	82%	82%	82%
Non-family, with children	33%	67%	67%	67%
None of these	3%	7%	9%	12%
Uncodeable/Error	N/A	3%	0%	2%

Note: Accuracy results are compared to the respondent's self-categorization HHTYPE1_NEXT. The highest accuracy by household category is indicated in **bold**. The "uncodeable/error" row represents the amount of error produced by the LLM.

Including demographic information greatly improves the coding of non-family households with children (by 33%) and marginally improves married and single household categories (1% to 2% improvement) in the GPT-5-mini model (Exhibit 7). Prompt #2 slightly improves coding of respondents who describe "other family" households with no children (7%) present in the GPT-5 model, and there is a small improvement for coding of single parent and "none of these" categories (1% to 5%) in the Claude-Sonnet-4.5 model. However, including the additional demographic context lowered the coding accuracy in some categories (e.g., other family households with children coded by GPT-5), as well as produced more coding errors. Differences in the accuracy of any particular category between Prompt #1 and Prompt #2 are not statistically significant across any model.

Exhibit 7. HHTYPE Prompt Improvement from Prompt #1 to Prompt #2.

Household Category	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Married, no children	0.9%	-0.4%	-2.7%
Married, with children	0.0%	-1.0%	-2.0%
Single parent	1.3%	-1.3%	1.3%
Single adult	2.2%	-0.4%	-4.8%
Cohabiting, no children	-4.0%	0.0%	0.0%
Cohabiting, with children	0.0%	-2.6%	-1.6%
Other Family, no children	-7.7%	7.7%	0.0%
Other Family, with children	0.1%	-11.8%	-0.1%
Non-family, no children	0.0%	0.0%	0.0%
Non-family, with children	33.4%	0.0%	0.0%
None of these	0.0%	0.0%	4.7%
Uncodeable/Error	2.8%	0.0%	2.3%

Note: Improvement accuracy results are compared to the respondent's self-categorization HHTYPE1_NEXT. The best improvement (i.e., higher than 0%) by household category is indicated in **bold**. The "uncodeable/error" row represents the amount of error produced by the LLM.

In sum, the expert code and LLM code accuracy perform similarly across household categories in terms of household structures that are representative of married couples with and without children, single adults, and non-

family households with no children (e.g., roommates). The expert code and LLMs have more difficulty parsing out single parent, cohabiting, other family, and non-family households with children present. Overall, the LLM that performs best with coding household composition open-ended responses is GPT-5-Mini, with or without having the demographic characteristics of the respondents. Moreover, GPT-5-Mini offered the best improved coding of non-family households with children, after being given the additional demographic context.

America's Top Problems (TOPPROB) Coding and Results

Next, we use the same framework to investigate the coding accuracy of open-ended responses to the attitudinal question, "What do you think is the most important issue for America today?" We provided the TOPPROB code frame seen in Exhibit 8 in the LLM coding prompts to guide the LLMs on how to assign verbatim responses to their appropriate categories. These are the same categories presented in the second question shown to respondents (TOPPROB1_NEXT, see Exhibit 2).

Exhibit 8. TOPPROB Code Frame.

Value Label	Response Category
1	Health care
2	Education
3	Crime
4	The environment
5	Immigration
6	The economy
7	Terrorism
8	Poverty
9	None of these

TOPPROB Coding Results Across Models and Prompts

We use the second question in the TOPPROB series, TOPPROB1_NEXT, as the source of truth for an accurate code as it is the respondents' self-coded answer to the open-ended question that was asked first. Exhibit 9 reports the overall coding accuracy of expert and LLM code, compared to the respondents' answers in TOPPROB1_NEXT across Prompt #1 (direct instructions) and Prompt #2 (including previous TOPPROB trends). As described in the Methods section, the previous TOPPROB trends are from the 2010 and 2021 GSS. Please see Appendix B for the detailed language used.

Overall, the coding accuracy of the TOPPROB open-ended questions is lower than that of household composition discussed in the previous sections. In TOPPROB coding, LLM GPT-5 produced a slightly better coding accuracy (64% to 65%) across both prompt strategies, compared to the other LLMs and the expert code. The expert code falls mid-range at 63% accuracy of the TOPPROB open-ended responses.

Exhibit 9. TOPPROB Overall Accuracy.

Prompt Strategy	Expert Code	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Prompt #1	63%	64%	65%	61%
Prompt #2 (+Change)		65% (+1%)	65% (+0%)	58% (-3%)

Note: Accuracy results are compared to the respondent's self-categorization TOPPROB1_NEXT. The highest accuracy by prompt is indicated in **bold**.

Prompt accuracy slightly varies by LLM. In general, GPT-5 performs only nominally (though not statistically significantly) better than its smaller counterpart, GPT-5-Mini, and better (statistically significantly so) than the large LLM Claude-Sonnet-4.5 in both the Prompt #1 ($p = 0.03$) and Prompt #2 ($p = 0.03$) conditions.

Exhibit 9 also reports on any overall coding improvement from Prompt #1 (direct instructions) to Prompt #2 (including previous TOPPROB trends). Of the LLMs, including previous trend information (Prompt #2) marginally improved the coding accuracy of GPT-5 and GPT-5-Mini ($<1\%$), but not Claude-Sonnet-4.5 (and none in a statistically significant way). Prompt #1 is not significantly different from Prompt #2 across any model's performance.

TOPPROB Coding Results Across Categories

Coding accuracy among American issues categories is highest for open-ended responses that do not fit in the code frame topics (last category "None of these"), health care, and the economy across both Prompt #1 (direct instructions) and Prompt #2 (including TOPPROB trend history) (Exhibit 10 and Exhibit 11). Because the human coder was not influenced or guided by additional information, such as the LLMs, the expert code accuracy results are shown in both Exhibit 10 and Exhibit 11, but the prompt manipulation is not relevant to the expert code accuracy results. We also report all errors produced by the LLMs in the last rows of Exhibit 10 and Exhibit 11.

Exhibit 10 reports on the coding accuracy of the expert code and the LLMs across American issue categories by using only direct instructions in Prompt #1. American issues described by respondents that do not fit in the substantive code frame topics, thus were coded in the "None of these" category, have a high coding accuracy by the expert code (92%) and the LLMs (ranging from 88% to 93%). Next, the economy (59% to 68%), immigration (51% to 52%), and health care (49% to 56%) categories fall into an intermediate level of coding accuracy. American issue categories with the lowest coding accuracy, and the most difficult for the expert code and LLMs, include education (26% to 31%) and terrorism (14% to 18%). Differences in accuracy between each of the GPT models and Claude-Sonnet-4.5 were statistically significant for the "None of these" category ($p < 0.01$). Otherwise, differences in performance across classifiers are not statistically significant, suggesting that large differences may be artifacts of the small incidence of the corresponding categories.

Exhibit 10. TOPPROB Prompt #1 Accuracy.

Problem Category	Expert Code	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Health care	56%	52%	53%	49%
Education	26%	31%	27%	27%
Crime	46%	47%	48%	49%
The environment	52%	45%	48%	43%
Immigration	51%	52%	52%	52%
The economy	60%	68%	68%	59%
Terrorism	14%	14%	14%	18%
Poverty	41%	36%	33%	43%
None of these	92%	92%	93%	88%
Uncodeable/Error	N/A	1%	0%	3%

Note: Accuracy results are compared to the respondent's self-categorization TOPPROB1_NEXT. The highest accuracy by category is indicated in **bold**, though the improvement is not necessarily statistically significant (we note where it is in the main text). The "uncodeable/error" row represents the amount of error produced by the LLM.

Exhibit 11 reports on the coding accuracy of the expert code and the LLMs across American issue categories by using the previous TOPPROB trends in Prompt #2. Specifically, we include 2010 and 2021 TOPPROB trends by

category as supplemental information in the second prompt to determine if additional trend information improves the coding accuracy and data quality of the simple directive output (Prompt #1).

Exhibit 11. TOPPROB Prompt #2 Accuracy.

Problem Category	Expert Code	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Health care	56%	53%	53%	45%
Education	26%	29%	27%	26%
Crime	46%	48%	51%	43%
The environment	52%	45%	48%	45%
Immigration	51%	52%	51%	51%
The economy	60%	67%	69%	57%
Terrorism	14%	14%	18%	19%
Poverty	41%	38%	36%	45%
None of these	92%	93%	93%	83%
Uncodeable/Error	N/A	0%	0%	4%

Note: Accuracy results are compared to the respondent's self-categorization TOPPROB1_NEXT. The highest accuracy by category is indicated in **bold**, though the improvement is not necessarily statistically significant (we note where it is in the main text). The "uncodeable/error" row represents the amount of error produced by the LLM.

Including TOPPROB trend information 2010 and 2021 in Prompt #2 improves some categories marginally (<4% improvement) and lowers the accuracy of others (Exhibit 12). By American issue category, the environment sees a slight improvement by GPT-5-Mini (<1%) and Claude-Sonnet-4.5 (2.6%) models; terrorism sees a small improvement by GPT-5 (3.5%) and Claude-Sonnet-4.5 (<1%); and the poverty category sees improvement across all models (ranging 2% to 3.5%). By LLM as a whole, GPT-5 sees the most improvement across categories, followed by GPT-5-Mini, then Claude-Sonnet-4.5. Differences in the accuracy of any particular category between Prompt #1 and Prompt #2 are not statistically significant across any model.

Exhibit 12. TOPPROB Prompt Improvement from Prompt #1 to Prompt #2.

Problem Category	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Health care	1.0%	0.0%	-4.0%
Education	-2.0%	0.0%	-0.9%
Crime	1.6%	2.4%	-6.6%
The environment	0.1%	0.0%	2.6%
Immigration	0.0%	-0.7%	-0.9%
The economy	-1.3%	0.4%	-2.2%
Terrorism	0.0%	3.5%	0.9%
Poverty	1.8%	3.5%	2.2%
None of these	1.5%	0.3%	-4.6%
Uncodeable/Error	0.5%	0.0%	0.9%

Note: Accuracy results are compared to the respondent's self-categorization TOPPROB1_NEXT. The best improvement (i.e., higher than 0%) by category is indicated in **bold**, though the improvement is not necessarily statistically significant (we note where it is in the main text). The "uncodeable/error" row represents the amount of error produced by the LLM.

In sum, as with the household coding, the expert code and LLM code accuracy perform similarly across American issue categories, with GPT-5 performing slightly better than the other LLMs and the expert code. Moreover, GPT-

5 saw improvement among some categories by including TOPPROB trend information in Prompt #2, and did not produce any coding errors in Prompt #1 or Prompt #2 results.

Overall, the coding accuracy of the TOPPROB open-ended questions is much lower than that of the household composition discussed in the previous sections. Though GPT-5 had the highest accuracy (64% with Prompt #1, 65% with Prompt #2), and the expert code was close in accuracy (63%), these numbers are still low compared to the accuracy provided in the household composition coding (81% accuracy by the expert code, 79% to 80% accuracy with GPT-5-Mini). This is primarily due to the nature of the question being asked. Accuracy among categories is highest when respondents are clear in their response; otherwise, responses are difficult to code. For example, if a respondent provided the verbatim response, “economic stability,” and then chose the category “Economy” for the second response, both human coder and LLMs are more likely to code this response accurately. On the other hand, if a respondent only provides “the divisiveness” as their verbatim response and then chooses the category “Economy” for the second response, neither human coder nor the LLMs will provide the “correct” coding.

In a certain respect, the respondent-driven codes come from a different context than the expert or LLM codes – they may represent two separate questions, rather than an attempt to assign a close-ended code to an open-ended response. From this perspective, both the expert and LLM coders are at a disadvantage, having been presented with a different task than the human “ground truth” measure.

TOPPROB Open-Ends: How Does LLM Coding Compare to Human Coding?

Due to the nature of having a lot of variance in the verbatim responses from the question asking about the “most important issue for America today,” and the low accuracy of both expert code and LLM code with the respondents’ “ground truth” measure, we provide an additional analysis of using the expert code as “truth” and how accurate the LLM coding is with a human coder.⁶ This additional analysis was conducted for household type as part of our standard assessments of the role of human oversight, but not shown here as the expert code and LLM code were relatively high (~80% accurate, see Exhibit 4).

Without using the respondents’ self-coded answer recorded in TOPPROB1_NEXT as the ground truth, the LLMs have a much higher accuracy compared to the expert code (e.g., the human coder) (Exhibit 13). These results indicate that both human coder and LLMs approached the attitudinal verbatims similarly. Using the expert code as “truth” in this scenario, GPT-5 has the highest accuracy across both prompts (92%), followed by GPT-5-Mini (91%), and Claude-Sonnet-4.5 last (81% to 85%). Differences in accuracy between each of the GPT models and Claude-Sonnet-4.5 are statistically significant ($p < 0.01$), though not with each other and not across prompt conditions.

Exhibit 13. TOPPROB LLM Overall Accuracy Against Expert Code.

Prompt Strategy	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Prompt #1	91%	92%	85%
Prompt #2 (+Change)	91% (+0%)	92% (+0%)	81% (-4%)

Note: Accuracy results are compared to the expert code of TOPPROB open-ended responses. The highest accuracy by prompt is indicated in **bold**.

⁶ This design matches the standard way that open-ended codes are usually processed in the General Social Survey (such as occupation, industry, and religion), with the expert reviewer providing the “ground truth” and overruling respondent complications in the process.

Using the expert code as the “ground truth” instead of the respondent’s open-ended answers, LLM coding accuracy among American issues categories is generally high across both Prompt #1 (direct instructions) and Prompt #2 (including TOPPROB trend history) (Exhibit 14 and Exhibit 15).

Exhibit 14 reports on the coding accuracy of the LLMs across American issue categories by using only direct instructions in Prompt #1. Interestingly, LLM coding has the highest accuracy with expert code in the category of terrorism (100% across all three LLMs), which had one of the lowest accuracy performances when compared to the respondents’ self-coded categorization. Categories with the lowest coding accuracy compared to the expert code include education (71% to 83%) and poverty (62% to 68%). As before, differences between the GPT models and Claude-Sonnet-4.5 are statistically significant for selected categories (“The economy”, $p < 0.01$, “None of these”, $p < 0.01$), however, no other differences in accuracy between models are statistically significant for any category.

Exhibit 14. TOPPROB Prompt #1 Accuracy Against Expert Code.

Problem Category	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Health care	91%	93%	87%
Education	83%	78%	71%
Crime	98%	98%	86%
The environment	85%	90%	88%
Immigration	94%	94%	84%
The economy	95%	96%	90%
Terrorism	100%	100%	100%
Poverty	62%	62%	68%
None of these	92%	93%	85%
Uncodeable/Error	1%	0%	3%

Note: Accuracy results are compared to the expert code of TOPPROB open-ended responses. The highest accuracy by category is indicated in **bold**. The “uncodeable/error” row represents the amount of error produced by the LLM.

Exhibit 15 reports on the coding accuracy of the LLMs compared to the expert code across American issue categories by using the previous TOPPROB trends in Prompt #2. Including 2010 and 2021 TOPPROB trends to guide Prompt #2 improved some categories and lowered the accuracy in others (Exhibit 16).

Exhibit 15. TOPPROB Prompt #2 Accuracy Against Expert Code.

Problem Category	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Health care	93%	93%	74%
Education	78%	78%	77%
Crime	93%	95%	79%
The environment	85%	90%	81%
Immigration	92%	93%	91%
The economy	94%	96%	81%
Terrorism	75%	100%	67%
Poverty	71%	67%	68%
None of these	92%	92%	82%
Uncodeable/Error	0%	0%	4%

Note: Accuracy results are compared to the expert code of TOPPROB open-ended responses. The highest accuracy by category is indicated in **bold**. The “uncodeable/error” row represents the amount of error produced by the LLM.

Exhibit 16. TOPPROB Prompt Improvement Against Expert Code.

Problem Category	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Health care	1.5%	0.0%	-13.3%
Education	-5.6%	-0.1%	6.3%
Crime	-4.6%	-2.3%	-7.1%
The environment	0.0%	0.0%	-7.0%
Immigration	-2.1%	-0.2%	6.8%
The economy	-0.6%	0.0%	-8.9%
Terrorism	-25.0%	0.0%	-33.3%
Poverty	9.5%	4.8%	0.1%
None of these	0.7%	-0.6%	-2.9%
Uncodeable/Error	-0.5%	0.0%	0.9%

Note: Accuracy results are compared to the expert code of TOPPROB open-ended responses. The best improvement (i.e., higher than 0%) by category is indicated in **bold**. The “uncodeable/error” row represents the amount of error produced by the LLM.

By LLM, there is variation across models with categories that see some, little, or no improvement. While both GPT-5-Mini and Claude-Sonnet-4.5 have categories with better and lower improved accuracy, GPT-5 shows very little change after including previous TOPPROB trends to guide Prompt #2. The same statistical differences in category performance between GPT and Claude-Sonnet-4.5 in Prompt #1 appear for Prompt #2.

In sum, while the coding accuracy of both expert code and LLMs compared to the respondents’ self-coding (TOPPROB1_NEXT, the respondents’ “ground truth”) was decent (ranging from 58% to 65% accuracy, see Exhibit 10), the LLMs’ accuracy was much closer to the expert code (81% to 92%) across both prompting strategies. GPT-5 had the highest accuracy to the expert code at 92% accurate overall and performed well across most categories except poverty and education. As before, GPT-5 saw improvement by including TOPPROB trend information in Prompt #2, particularly in the poverty category, and did not produce any coding errors in Prompt #1 or Prompt #2 results. These results indicate that the human coder and LLMs approached coding the attitudinal verbatims similarly. GPT-5 continued to perform the best for this set of verbatim responses, compared to the other models.

DISCUSSION

This research evaluates LLM-assisted response coding on a subset of GSS respondents (i.e., the GSS follow-on study), and the findings offer insights that can inform future large-scale applications, help address gaps and build on LLM-assisted coding literature, and provide empirical data to advance the future of AI-driven survey methodology. To investigate the utilization of LLMs to code verbatim respondent answers, we use open-ended responses about household composition (HHTYPE) and attitudes about American issues (TOPPROB) from the 2024 GSS Next follow-on questionnaire. The GSS provides a unique and novel opportunity to compare traditionally used open-ended coding efforts (i.e., manual coding with human labor) with LLMs and has a breadth of data to test prompts with additional contexts.

Summary of Findings

In general, we find that accuracy is comparable between LLMs and expert coders, regardless of prompt type or model. However, taking the accuracy performances in our sample at face value, the expert coder had the highest accuracy in coding household composition and was on par with LLMs for coding American issues. Similar to other

research (e.g., Chen et al. 2025; Kumar et al. 2025), we find that the smaller model performed well, while larger models performed with less accuracy, depending on the coding frame. Specifically, the smaller LLM (GPT-5-Mini) performed better regarding coding accuracy of household composition, compared to the two larger LLMs used (GPT-5 and Claude-Sonnet 4.5). Regarding the coding of American issues, however, GPT-5 performed the best, and only marginally better than GPT-5-Mini. Interestingly, the most “powerful” LLM we tested, Claude-Sonnet-4.5, performed the worst, *which was among the most consistent and statistically significant findings*. This could be due to factors associated with the size of Claude-Sonnet-4.5, as discussed in Limitations below.

LLMs demonstrate a much higher accuracy in classifying responses to factual questions, such as household composition, than responses to attitudinal questions, such as asking respondents to describe what they think is the top issue in America today. Even though there are complex household arrangements, in general, respondents were clear about their household composition, and in turn, would categorize their household in the follow-up question (e.g., HHTYPE open-ended question followed by HHTYPE1_NEXT described in this report).

By contrast, the attitudinal question was more difficult for the human coder and LLMs to code. This may be due to the nature of the question being asked. In the open-ended question asked first, respondents provided a wide array of responses describing top issues in America, but had difficulty fitting their responses into the categories presented in the follow-up question (e.g., the TOPPROB open-ended question followed by TOPPROB1_NEXT described in this report). Because of the high variance in these responses, we provide an additional analysis using the expert code as “truth” and assessing how accurate the LLM coding is compared to that of the expert coder. **Interestingly, we find that both the expert coder and LLMs approached the attitudinal verbatims similarly, with higher agreement particularly with the GPT models, raising questions around the meaning and measurement of “ground truth.”** This finding points to the distinction between respondent meaning, expert classification, and model classification.

Timing and Cost Comparison

As discussed so far, the accuracy of coding methods across prompting strategies has varied; the expert coder does a good job coding both sets of verbatim responses, with GPT-5-Mini performing the best of the LLMs for coding household composition, and GPT-5 performing the best of the LLMs for coding attitudinal responses about American issues. However, we also want to know if using LLMs is more time and cost-efficient than solely relying on human coding.

In general, base prompting is faster for LLMs compared to contextual prompts (e.g., including demographic characteristics of respondents or previous data trends), though the costs remain relatively low. It is important to note that these high-level takeaways do not include the time or human labor costs associated with writing prompts, writing and troubleshooting R scripts, preparing secondary data for the contextual prompts, or doing quality checks of the coding output. Instead, timing and cost represent the metrics produced based on the LLM usage. Nor do our estimates for expert coding include previous time working with the GSS data, prior rounds of open-ended coding addressing similar questions, and the educational background and training that have gone into the general knowledge of open-ended coding approaches. No fair comparison may exist between these approaches.

Generally speaking, no individual LLM’s value is solely evident in the time it takes to run and the cost of computing. These both underestimate the human tweaking and error correction necessary to produce analyzable results. More time was spent troubleshooting R scripts and developing quality checks than on each prompt to run and produce results. However, the pipeline established by this work may prove both beneficial and efficient for enabling future rounds of research, much more directly than additional expert coder experiences.

Considering Human-in-the-loop Processes Using LLMs for Data Processing and Survey Design

While this study shows that LLMs can do a decent job of coding open-ended responses, human intervention may still be required to make them more accurate, efficient, and effective. This research has shown that LLMs make useful tools for coding open-ended responses; however, human review is still necessary to check accuracy, apply judgment, and ensure the design and output are appropriate for the research. Having a “human-in-the-loop” process maintains high data quality while also pushing forward AI development in survey research.

Finally, this research also considers broader implications for survey design. If LLM-assisted coding proves reliable, it could inform discussions about shorter questionnaires and reduce respondent burden by replacing lengthy question batteries with open-ended questions coded by AI. These potential changes may improve respondent engagement and data richness, lower survey costs, and address important questions about the transparency and reproducibility of automated workflows.

Limitations

There are limitations to this study. First, we do not account for temperature settings across LLMs, which can control the randomness and range of a model’s application (Li et al. 2025). Although we do provide all three LLMs directives through prompting, more attention in this area could alter the coding results, particularly among the largest model Claude-Sonnet-4.5.

There are, of course, limitations to using LLMs to code open-ended responses, as this area is still relatively new in survey research. Troubleshooting can be difficult because model outputs may vary across prompts and contexts, and problems such as inconsistency and uneven reasoning are not always easy to diagnose. Type and size of model also matter, as well as settings such as temperature, since these choices can affect how stable, precise, or creative the output becomes. Last, the quick evolution of LLMs can also present challenges to reproducibility and will require ongoing oversight.

Data for Download

For transparency, we provide researchers with the data mentioned in this report, including the coding output of the expert code, LLMs, and LLM confidence scores for both HHTYPE and TOPPROB variables. Please visit [gss.norc.org/get-the-data](https://gss.norc.umd.edu/get-the-data) to download the data. The dataset does not include the verbatim responses. Researchers interested in this restricted data can email gss@norc.org or learn how to prepare a GSS Sensitive Data Application [here](#).

ACKNOWLEDGEMENT

The authors wish to acknowledge our reviewers for their contributions to this report: Michael Davern, Susan Paddock, Micah Sjoblom, Brian M. Wells, and Ting Yan. This report was prepared with support from the National Science Foundation grant award [NSF grant #2049169].

The recommended citation for this report is:

Sparkman, R., Barari, S., Schapiro, B., & Bautista, R. (2026). *Leveraging Large Language Models to Code Open-Ended Responses in the General Social Survey* (GSS Methodological Report No. 150). NORC at the University. <https://gss.norc.org/content/dam/gss/get-documentation/pdf/reports/methodological-reports/GSS%20MR150%20Leveraging%20Large%20Language%20Models.pdf>

REFERENCES

- Andridge, R. R., & Little, R. J. (2010). A Review of Hot Deck Imputation for Survey Non-Response. *International Statistical Review*, 78(1), 40-64.
- Atreja, S., Ashkinaze, J., Li, L., Mendelsohn, J., & Hemphill, L. (2025, June). What's in a Prompt?: A Large-Scale Experiment to Assess the Impact of Prompt Design on the Compliance and Accuracy of LLM-Generated Text Annotations. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 19, pp. 122-145). <https://doi.org/10.1609/icwsm.v19i1.35807>
- Barari, S., Angbazo, J., Wang, N., Christian, L. M., Dean, E., Slowinski, Z., & Sepulvado, B. (2026). AI-Assisted Conversational Interviewing: Effects on Data Quality and Respondent Experience. Forthcoming in *Survey Research Methods*.
- Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1), 289-300.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901.
- Buskirk, T. D., Keusch, F., von der Heyde, L., & Eck, A. (2025). More Parameters Than Populations: A Systematic Literature Review of Large Language Models within Survey Research. *arXiv preprint arXiv:2509.03391*. <https://doi.org/10.48550/arXiv.2509.03391>
- Chen, Z., Wang, S., Xiao, T., Wang, Y., Chen, S., Cai, X., ... & Wang, J. (2025). Revisiting scaling laws for language models: The role of data quality and training strategies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 23881-23899).
- Chew, R., Eckman, S., Kern, C., & Kreuter, F. (2026). From Ground Truth to Measurement: A Statistical Framework for Human Labeling. *arXiv preprint arXiv:2604.07591*. <https://arxiv.org/abs/2604.07591>
- General Social Survey. (n.d.). About the GSS. NORC at the University of Chicago. <https://gss.norc.uchicago.edu/About-The-GSS>
- Grossmann, I., Feinberg, M., Parker, D. C., Christakis, N. A., Tetlock, P. E., & Cunningham, W. A. (2023). AI and the transformation of social science research. *Science*, 380(6650), 1108-1109. DOI: 10.1126/science.adi1778
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35, 22199-22213.
- Kostina, A., Dikaiakos, M. D., Stefanidis, D., & Pallis, G. (2025). Large language models for text classification: Case study and comprehensive review. *arXiv preprint arXiv:2501.08457*.
- Kumar, C. V., Urlana, A., Kanumolu, G., Garlapati, B. M., & Mishra, P. (2025). No LLM is free from bias: A comprehensive study of bias evaluation in large language models. *arXiv preprint arXiv:2503.11985*.

- Li, L., Sleem, L., Gentile, N., Nichil, G., & State, R. (2025). Exploring the Impact of Temperature on Large Language Models: Hot or Cold? *Procedia Computer Science, International Neural Network Society Workshop on Deep Learning Innovations and Applications*, 264, 242–251. <https://doi.org/10.1016/j.procs.2025.07.135>
- Mellon, J., Bailey, J., Scott, R., Breckwoldt, J., Miori, M., & Schmedeman, P. (2024). Do AIs know what the most important issue is? Using language models to code open-text social survey responses at scale. *Research & Politics*, 11(1), 20531680241231468. <https://doi.org/10.1177/20531680241231468>
- Petty, R. E., & Krosnick, J. A. (2014). *Attitude Strength: Antecedents and Consequences*. Psychology Press.
- Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., ... & Rand, D. G. (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4), 1064-1082.
- Rothschild, D. M., Buskirk, T. D., Eckman, S., Hillygus, D. S., Kreuter, F., & Lazer, D. (2025). Successfully Navigating the Disruption AI will Bring to Survey Research. *The Survey Statistician*, 92(1), 30–44. https://isi-iass.org/home/wp-content/uploads/Survey_Statistician_2025_July_N92_04.pdf
- Rothschild, D. M., Marlar, J., Pew, A. A., Barari, S., Krupenkin, M., Lee, S., ... & Webb, B (2026). Responsible AI Integration in Survey Research. *AAPOR Taskforce Reports*. <https://www.aapor.org/wp-content/uploads/2026/05/Responsible-AI-Integration-In-Survey-Research.pdf>
- Rytting, C. M., Sorensen, T., Argyle, L., Busby, E., Fulda, N., Gubler, J., & Wingate, D. (2023). Towards Coding Social Science Datasets with Language Models (No. arXiv:2306.02177). *arXiv*. <http://arxiv.org/abs/2306.02177>
- Singh, Tushar and Himangshu Kumar. (2025). AI in qualitative research: Using large language models to code survey responses in native languages. *IFPRI Blog: Research Post, Natural Resources and Resilience*. <https://www.ifpri.org/blog/ai-in-qualitative-research-using-large-language-models-to-code-survey-responses-in-native-languages>
- Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., ... & Manning, C. D. (2023, December). Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 5433-5442).
- Von Der Heyde, L., Haensch, A. C., Weiß, B., & Daikeler, J. (2025). Using Large Language Models for Coding German Open-Ended Survey Responses on Survey Motivation. *Survey Research Methods* (Vol. 19, No. 4, pp. 355-370). <https://doi.org/10.48550/arXiv.2506.14634>
- Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., & Hooi, B. (2024, May). Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations* (Vol. 2024, pp. 23650-23678).
- Yang, D., Tsai, Y. H. H., & Yamada, M. (2024). On verbalized confidence scores for LLMs. *arXiv preprint arXiv:2412.14737*.
- Zhang, S., Xu, J., & Alvero, A. J. (2025). Generative AI meets open-ended survey responses: Research participant use of AI and homogenization. *Sociological Methods & Research*, 54(3), 1197-1242. <https://doi.org/10.1177/00491241251327130>

Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can large language models transform computational social science?. *Computational Linguistics*, 50(1), 237-291.
https://doi.org/10.1162/coli_a_00502

APPENDIX A: HHTYPE PROMPTS

For transparency, we included the prompt language used in Prompt #1 and Prompt #2 to guide LLM coding of HHTYPE verbatim responses in this research. Differences between the two prompts are highlighted in the body of Prompt #2.

Prompt #1

You are an expert survey research coder trained in coding open-ended responses with high precision.
Your task is to assign the most appropriate category label from the list below to a respondent's open-ended answer.
The responses correspond to this question: Please describe your household - how many people live in your household and what is each of their relationships with you and each other

Assign the responses to this question into the following categories:

1. Married couple, no children in the household
2. Single parent
3. Other family relationship (e.g. cousins, grandparents, aunts & uncles), no children in the household
4. Single adult
5. Cohabiting couple, no children in the household
6. Non-family relationship (e.g. friends, coworkers), no children in the household
7. Married couple, children in the household
8. Cohabiting couple, children in the household
9. Other family relationship, children in the household
10. Non-family relationship, children in the household
11. None of these

Think carefully about each option before you assign a final numeric code to each open-ended response.

Classification rules

- Assign the most applicable numeric label from the list, or .S if the response contains only .s, for skipped
- Return the ID of the case, the numeric code that best fits the response, and a confidence in the accuracy of the assigned code.
- Output must be in the following format:
ID:assigned code:confidence
- Confidence must be a number from 0 to 1 (two decimal places).
- Do NOT output explanations, reasoning, or additional text.
- Use only the values listed for the categories above.
- Base coding on meaning, not mere word overlap.
- Do not infer beyond the response content.

Prompt #2

You are an expert survey research coder trained in coding open-ended responses with high precision.
Your task is to assign the most appropriate category label from the list below to a respondent's open-ended answer.
The responses correspond to this question: Please describe your household - how many people live in your household and what is each of their relationships with you and each other

Assign the responses to this question into the following categories:

1. Married couple, no children in the household
2. Single parent
3. Other family relationship (e.g. cousins, grandparents, aunts & uncles), no children in the household
4. Single adult
5. Cohabiting couple, no children in the household
6. Non-family relationship (e.g. friends, coworkers), no children in the household
7. Married couple, children in the household
8. Cohabiting couple, children in the household
9. Other family relationship, children in the household
10. Non-family relationship, children in the household
11. None of these

To help you assign responses, review the other demographics information included with each case:
marital_status: The respondent's marital status.

num_child: Number of children the respondent has ever had.
age: The respondent's age.
sex: The respondent's sex.
us_born: If the respondent was born in the US or not.

Think carefully about each option before you assign a final numeric code to each open-ended response.

Classification rules

- Assign the most applicable numeric label from the list, or .S if the response contains only .s, for skipped
- Return the ID of the case, the numeric code that best fits the response, and a confidence in the accuracy of the assigned code.
- Output must be in the following format:
ID:assigned code:confidence
- Confidence must be a number from 0 to 1 (two decimal places).
- Do NOT output explanations, reasoning, or additional text.
- Use only the values listed for the categories above.
- Base coding on meaning, not mere word overlap.
- Do not infer beyond the response content.

APPENDIX B: TOPPROB PROMPTS

For transparency, we included the prompt language used in Prompt #1 and Prompt #2 to guide LLM coding of TOPPROB verbatim responses in this research. Differences between the two prompts are highlighted in the body of Prompt #2.

Prompt #1

You are an expert survey research coder trained in coding open-ended responses with high precision. Your task is to assign the most appropriate category labels from the list below to a respondent's open-ended answer. The responses correspond to this question: What do you think is the most important issue for America today?

Categorize the responses to this question into one of the following topics:

1. Healthcare
2. Education
3. Crime
4. The environment
5. Immigration
6. The economy
7. Terrorism
8. Poverty
9. Something else

Think carefully about each option before you assign a final numeric code to each open-ended response. Respondents were also asked if any other codes apply. After you select the most applicable code, please assign additional codes which correspond to the other categories, if possible. Do not reassign the original code a second time.

Classification rules

- Assign the single most relevant numeric code from the list, or .S if the response contains only .s, for skipped
- Include any additional numeric codes as necessary, each separated by ","
- Return the ID of the case, the numeric code or codes that best fits the response, and a confidence in the accuracy of the assigned code.
- Output must be in the following format:
ID:assigned code:additional codes:confidence
- Confidence must be a number from 0 to 1 (two decimal places).
- Do NOT output explanations, reasoning, or additional text.
- Use only the values listed for the categories above.
- Base coding on meaning, not mere word overlap.
- Do not infer beyond the response content.

Prompt #2

You are an expert survey research coder trained in coding open-ended responses with high precision. Your task is to assign the most appropriate category labels from the list below to a respondent's open-ended answer. The responses correspond to this question: What do you think is the most important issue for America today?

Categorize the responses to this question into one of the following topics:

1. Healthcare
2. Education
3. Crime
4. The environment
5. Immigration
6. The economy
7. Terrorism
8. Poverty
9. Something else

Think carefully about each option before you assign a final numeric code to each open-ended response. Respondents were also asked if any other codes apply. After you select the most applicable code, please assign additional codes which correspond to the other categories, if possible. Do not reassign the original code a second time.

You can also consider the historical trend for TOPPROB when assigning initial codes. The existing trend for the TOPPROB data are as follows:

In 2010, 22% of people felt that health care was the most important issue for America and that increased to 32.2% in 2021.
 In 2010, 18.8% of people felt that education was the most important issue for America and it decreased to 12.7% in 2021.
 In 2010, 1.7% of people felt that crime was the most important issue for America and it increased to 3.8% in 2021.
 In 2010, 4% of people felt that the environment was the most important issue for America and it increased to 11.8% in 2021.
 In 2010, 4.4% of people felt that immigration was the most important issue for America and it decreased to 4.0% in 2021.
 In 2010, 37.7% of people felt that the economy was the most important issue for America and it decreased to 21.8% in 2021.
 In 2010, 7.8% of people felt that terrorism was the most important issue for America and it decreased to 2.4% in 2021.
 In 2010, 2.7% of people felt that poverty was the most important issue for America and it increased to 7.5% in 2021.
 In 2010, 1% of people felt that none of the response options was the most important issue for America and it increased to 3.8% in 2021.

Classification rules

- Assign the single most relevant numeric code from the list, or .S if the response contains only .s, for skipped
- Include any additional numeric codes as necessary, each separated by ","
- Return the ID of the case, the numeric code or codes that best fits the response, and a confidence in the accuracy of the assigned code.
- Output must be in the following format:
 ID:assigned code:additional codes:confidence
- Confidence must be a number from 0 to 1 (two decimal places).
- Do NOT output explanations, reasoning, or additional text.
- Use only the values listed for the categories above.
- Base coding on meaning, not mere word overlap.
- Do not infer beyond the response content.

APPENDIX C: CONFIDENCE SCORES

In addition to analyzing the accuracy of the coding with respect to the respondent's own coding, we elicited each LLM for a confidence score for each of its predicted response codes. LLM self-reported confidence scores may have limited utility. Some researchers (Tian et al. 2023) have found that confidence scores can be well-calibrated, while others (Yang et al. 2024; Xiong et al. 2024) show evidence that LLMs report confidence similar to how humans report it, tending towards over-confidence regardless of actual accuracy.

Exhibits C1 and C2 display the results previously shown in Exhibits 4 and 9, respectively, with the average of confidence scores for predictions from each of the three models. Interestingly, the LLM-reported confidence score reports in the opposite direction with GPT-5-mini reporting a lower average confidence compared to GPT-5 and Claude-Sonnet-4.5, though still performing with a higher coding accuracy. In general, we do not surface convincing evidence that confidence scores are strong predictors of performance.

Exhibit C1. HHTYPE Model Accuracy and Confidence.

Prompt Strategy	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Prompt #1	79%	79%	77%
Confidence	86%	90%	93%
Prompt #2 (+Change)	80% (+1%)	78% (-1%)	76% (-1%)
Confidence	86%	90%	91%

Note: Accuracy is determined according to the respondent's self-categorization HHTYPE1_NEXT. The highest accuracy by prompt is indicated in **bold**.

Exhibit C2. TOPPROB Model Accuracy and Confidence.

Prompt Strategy	GPT-5-Mini	GPT-5	Claude-Sonnet-4.5
Prompt #1	64%	65%	61%
Confidence	88%	87%	86%
Prompt #2 (+Change)	65% (+1%)	65% (+0%)	58% (-3%)
Confidence	86%	86%	85%

Note: Accuracy results are compared to the respondent's self-categorization TOPPROB1_NEXT. The highest accuracy by prompt is indicated in **bold**.