ECONtribute
MARKETS & PUBLIC POLICY

ECONtribute
Discussion Paper No. 416

A Golden Era for Open-Ended Questions? Using LLMs for Text Classification Tasks

Ansgar Hudde

Shannon Taflinger

June 2026

www.econtribute.de

UNIVERSITÄT BONN

UNIVERSITÄT ZU KÖLN

Funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2126/2 – 390838866 – is gratefully acknowledged.

A golden era for open-ended questions? Using LLMs for text classification tasks

Ansgar Hudde* & Shannon Taflinger

University of Cologne, Department of Sociology and Social Psychology

* Corresponding author, hudde@wiso.uni-koeln.de

ABSTRACT

Open-text questions in quantitative surveys can yield rich information from large samples, but analysing and coding these data using qualitative text analysis is resource-intensive. Large Language Models (LLMs) are a promising tool for scaling up such analyses, reducing time and financial costs. In this paper, we compare the coding accuracy of LLMs with that of student assistants, defining accuracy as agreement with a researcher-coded benchmark dataset. We assess performance on a semi-complex coding task: coding approximately 1,400 open-ended text responses from young US Americans about dating across party-political lines. A researcher-designed coding scheme, developed through thematic qualitative text analysis of the open-text responses, was applied by LLMs and student assistants. We evaluate models from OpenAI, Anthropic, and Mistral, with and without access to training data. The most advanced models outperform student assistants, and performance further increases with training data, highlighting LLMs' capability to code open-text responses. Whereas previous research has mainly focused on social media texts, comparatively simple and surface-level coding tasks, and a technically oriented audience, we contribute to the literature by studying a particularly promising use case of open-ended survey responses and by providing practical recommendations to applied social scientists.

Keywords:

Large language models; open-ended questions; text analysis

1. Introduction

The features that make open-ended survey questions uniquely valuable, namely that they can capture unanticipated content in respondents' own words, are precisely those that render them difficult to analyse systematically. Traditionally, researchers have relied on two approaches for analysing such data: They can use qualitative methods, which are flexible and allow for a depth of analysis, yet require costly coding of large amounts of text by hand. Alternatively, they can use computational approaches, which have the advantage of being able to analyse large amounts of data, but require advanced computing skills and may only scratch the surface (Than et al., 2025). However, recent developments in Large Language Models (LLMs) provide a promising avenue for combining qualitative depth with computational efficiency (Abramson et al., forthcoming).

Despite this potential, there is a paucity of research examining whether LLMs can be used to code open-text responses from survey data. Existing research on LLMs for text coding has largely focused on social media texts, particularly Twitter, with largely encouraging results (e.g., Chae & Davidson, 2025; Gilardi et al., 2023; Ivanusch, 2024; Mens & Gallego, 2023; Rathje et al., 2023; Törnberg, 2025). Given that LLMs are trained on social media data, it remains unclear whether their performance extends to survey responses — and whether any biases in coding would lead applied researchers to draw different substantive conclusions.

We consider open-ended questions as a particularly important application because they combine the benefits of survey research — namely the possibility for representative samples, collection of detailed demographic and other self-reported information — with the opportunity for respondents to share their thoughts, narratives, and ideas in their own words, unconstrained by predefined response categories (Fielding et al., 2013; Foddy & Foddy, 1993; Reja et al., 2003). As they are embedded in surveys, this also allows researchers to analyse responses to open-text questions jointly with other information collected (Fielding et al., 2013). In practical terms, open-ended text questions are easy and cost-efficient for social scientists to collect, as they can readily be integrated into existing surveys and data collection infrastructures. In fact, many existing datasets already include responses to open-ended texts, but they often remain unanalysed, because of the high effort and cost of analysis at scale (Bernhard-Harrer & Pfaff, 2025; Fielding et al., 2013; Heyde et al., 2025; Schonlau & Couper, 2016).

In this paper, we test whether LLMs enable large-scale analyses of open-ended text questions by applying a researcher-developed coding scheme of 15 codes across four overarching categories. Specifically, we examine young Americans' narratives about dating across party-political lines from a quota-representative survey of 1,419 US Americans aged 20-33. Respondents were asked to elaborate

on any “reasons, thoughts, feelings, considerations, and reflections” they may have about dating across partisan lines. This dataset provides an ideal test case for LLMs, requiring nuanced coding that captures subtle distinctions in attitudes, reasoning, and emotional responses.

Our methodological approach consists of six steps. (1) We manually developed a coding scheme of responses using thematic qualitative text analysis. (2) Next, we created a hand-coded benchmark dataset of all responses. (3) Two Master-level student assistants and various LLMs were provided with the same coding scheme and coded all responses. We tested different LLMs from OpenAI / GPT, Anthropic / Claude, and Mistral. (4) We compared the models’ performance using only the provided coding scheme versus providing an additional training dataset of 200 randomly selected, hand-coded responses. (5) Further, we tested whether LLM coding accuracy differs by the level of ambiguity of a certain response coding and by individual code-categories. (6) Finally, we tested a use case for the coded responses, examining whether coded text responses predict behaviour in a survey experiment. This allows us to identify whether applied researchers would draw the same substantive conclusions with hand-coded or LLM-coded data.

This paper offers several contributions to the applied methodological literature on the use of LLMs for text analysis. First, we apply LLMs to a use case that remains underexplored despite its great potential for applied social sciences — coding open-text data from a survey. Second, we compare not only the accuracy of coding but also test a realistic and relevant use case for the coded data. Finally, we provide actionable guidelines and suggestions for how applied researchers in the social sciences can use LLMs for coding open-ended answers in large-scale surveys.

While our findings may interest researchers from various disciplines, our target audience is applied social science researchers who may have open-text data or are considering collecting such data but hesitate due to high analysis costs. We particularly aim to provide practical advice and an analytical roadmap for both qualitative researchers wanting to leverage larger samples and quantitative researchers recognizing the value of qualitative insights. Our guidance focuses on approaches that can be run on a standard computer and require no specialized programming skills, supported by replication code that other researchers can readily adapt.

2. Background

New frontier: incorporating LLMs in the research process

Research that uses LLMs for text analysis differs in which parts of the research process incorporates LLMs, and there are lively debates about how much of this process should be carried out by LLMs (for overviews and discussions, see Chatzichristos, 2025; Paulus et al., 2025; Alvero et al., 2026; Abramson et al., forthcoming). Some approaches aim to delegate core analytical and creative tasks typically done by expert researchers, such as identifying themes in texts, developing categories, or providing substantive interpretations (e.g., Abramson et al., forthcoming; Jiang et al., 2025; Kabir et al., 2025; Lee et al., 2024; Wachinger et al., 2025). However, others argue that LLMs lack the depth and nuance to perform such tasks (Morgan, 2023) or propose "hybrid" models where researchers and LLMs perform these analytical and creative tasks in tandem, with the researchers steering and controlling the process (Breznau & Nguyen, 2026; e.g., Hayes, 2025; Krähnke et al., 2025).

Overall, many qualitative researchers see potential in using LLMs and similar artificial intelligence (AI) tools for their work, but "under the condition that interpretive analysis remains in the hands of the researcher" (Cabanillas-García et al., 2025, p. 15; see also Chatzichristos, 2025; Tai et al., 2024; Abramson et al., forthcoming; Breznau & Nguyen, 2026). The workflow we test in this paper is consistent with this position. The central analytical, interpretive, and creative tasks — identifying themes in responses, defining and interpreting categories, and developing a coding scheme — are carried out by researchers using qualitative methods. LLMs are used only to carry out the subsequent standardised coding tasks.

Previous research and gaps

A rapidly growing body of studies examines the capabilities of LLMs to apply codes to text data. Given the breadth of the research, we here focus on studies that compare the performance of LLMs to that of human coders using a predefined coding scheme.[1] As a vanguard study, Gilardi et al. (2023) analyse

[1] Other studies compare the performance of LLMs against alternative computational methods, such as machine learning approaches like BERT-based models. BERT-based models typically require large training datasets and computational resources that exceed regular computers. Such studies tend to find that LLMs perform at least as

several coding tasks applied to newspaper articles and tweets — for example, coding whether a text expresses a positive, negative, or neutral stance toward a given piece of legislation. They find that GPT-3.5 outperforms crowd workers (MTurk). Ornstein et al. (2025) examine coding tasks applied to tweets and political ads, in which the coding task was to identify the "tone of a political advertisement [as] positive, neutral, or negative". When the LLMs (GPT-3 and GPT-4) were provided with a few coded examples, their coding aligned more closely with expert coding than crowd workers aligned with expert coding. Mellon et al. (2024) studied open-ended survey questions in which respondents described what they saw as the "most important issue" facing the country. These responses were then coded into categories, such as "economy", "health", or "immigration", and performed roughly on par with a human coder.

Overall, several studies show that LLMs are able to code texts at levels broadly similar to or better than human coders (Davidson & Karell, 2025; Kravets-Meinke et al., 2025; Törnberg, 2024; see also Chew et al., 2023; Kirsten et al., 2024; Matter et al., 2024; Törnberg, 2025; Tripp, 2025). Moreover, several studies show that more advanced and complex models tend to perform better (Chae & Davidson, 2025; Kirsten et al., 2024; Macanovic & Przepiorka, 2024; Matter et al., 2024; Mellon et al., 2024). For instance, several studies directly compare GPT-3.5 and GPT-4, finding higher performance for GPT-4, the more advanced model (Kirsten et al., 2024; Macanovic & Przepiorka, 2024; Matter et al., 2024). This between-model gap seems particularly large for more complex coding tasks (Kirsten et al., 2024).

Within the body of available research, we identify three interrelated gaps that motivate our study:

First, existing work has rarely studied open-ended survey responses, but rather focused on social media texts, such as Facebook posts (Chae & Davidson, 2025), Reddit posts (Macanovic & Przepiorka, 2024; Rathje et al., 2023), and other forums (Matter et al., 2024; Ziems et al., 2024).[2] A particularly large

---

well as, or better than, these alternatives (Chae & Davidson, 2025; Macanovic & Przepiorka, 2024; Rathje et al., 2023; Törnberg, 2025; Ziems et al., 2024). Meanwhile, other studies assess LLM performance in absolute terms rather than through comparison (Heseltine & Clemm von Hohenberg, 2024). For example, such studies conclude that an LLM works well on a given task because it codes accurately in 90% of cases, without examining whether other computational approaches or human coders would perform better or worse (Burnham, 2024; Heseltine & Clemm von Hohenberg, 2024).

[2] Some studies analyse different types of texts and thereby appear in this overview multiple times.

number of studies have analysed Twitter data (Chae & Davidson, 2025; Chew et al., 2023; Gilardi et al., 2023; Ivanusch, 2024; Macanovic & Przepiorka, 2024; Mens & Gallego, 2023; Rathje et al., 2023; Törnberg, 2025). Further, several studies have analysed newspaper text (Chew et al., 2023; Gilardi et al., 2023; Ivanusch, 2024; Rathje et al., 2023; Tripp, 2025). In contrast, only a few studies have applied LLMs to open-text survey responses. Among the exceptions are studies that coded responses to the question of what respondents consider the most important issue facing the country (Mellon et al., 2024) or classified one-line responses stating why respondents participate in a survey (Heyde et al., 2025). These studies focus on very brief texts, only one line or even just one word, and test a comparatively straightforward, surface-level coding, rather than coding by a more nuanced coding scheme.

With the bulk of research focusing on, in particular, social media texts, it is unclear to what degree such evidence on the performance of LLMs can be transferred to open-ended texts, because these open-ended response data differ from social media texts or newspaper articles in meaningful ways. These differences include that open-text answers are prompted by specific questions, they typically come from recruited respondents, rather than users of a particular platform, and they represent personal and anonymous reasoning rather than public discourse. Finally, LLMs are trained on data such as tweets, Reddit posts, and newspaper articles (Brown et al., 2020; Radford et al., 2019), but likely not, or at least to a lesser degree, on open-ended text answers from anonymous surveys — and LLMs might perform better on the types of texts on which they are trained.

Second, in addition to the focus on social media tasks, many studies of human coders focus on comparatively simple, surface-level tasks (Chae & Davidson, 2025; Gilardi et al., 2023; Heseltine & Clemm von Hohenberg, 2024; Rathje et al., 2023; Törnberg, 2025). This includes the classification of texts into only two or three categories — such as detecting positive, neutral, or negative stances — or coding texts that consist of only a single line or even a single word (Heyde et al., 2025; Mellon et al., 2024). On such tasks, both human coders and LLMs can achieve very high accuracy, because there is little ambiguity or subjectivity involved. In contrast, many coding tasks relevant to social science research require more complex, nuanced, or interpretive categorizations.

Third, most existing studies are designed with a computational or technically oriented audience in mind. What is lacking are studies tailored to applied social science researchers, with evaluation criteria that go beyond overall accuracy to test whether coding errors introduce systematic bias or alter substantive conclusions, and that provide accessible workflows and practical guidance on adoption.

Against this background, we test the performance of LLMs on open-ended survey responses, focusing on a coding task with a higher level of complexity than in many previous tests. Beyond overall accuracy,

we test whether performance varies with response ambiguity and whether coding differences alter substantive conclusions. Finally, we implement and document a workflow that applied social scientists can execute with standard statistical software and on standard computers and address the practical questions researchers face when adopting LLMs for text coding.

3. Methods

All replication files, including data and analysis code, are available here: https://osf.io/3nxrv/overview?view_only=90f6f3b4db9d4291b1550183bd1b66ec.

3.1. Data collection

We conducted a survey among a quota-representative (gender, region) sample of US Americans aged 20-33 about political polarization and online dating. Participants were recruited via the market research company Dynata in February 2024. Ethical approval was received from University of Cologne, Faculty of Management, Economics and Social Sciences. The first half of the survey was a vignette experiment. The second half asked respondents about their interest in and experience with dating members of the other political party. The open-response question we focus on in this study is: "We are interested in learning more about why people are open or hesitant to dating across party lines. Please elaborate on why or why not you would be willing to date a [Republican/Democrat]". The question was coded so that Democrats were shown the word "Republican" and vice versa. Independents and non-voters were randomly assigned to be asked about Democrats or Republicans. The average length of answer was 15 words (standard deviation = 14; median = 11 words).

Of the 3,418 people who clicked on the survey link, 2,037 (59.6%) provided their informed consent to participate, and 1,502 (43.9%) respondents completed the survey. One respondent was removed due to low quality (failed quality check and non-sensical open-text response). 83 participants were shown an alternative question wording and, therefore, removed from the final sample. Thereby, the final analytical sample includes 1,419 respondents.

3.2. Qualitative text analysis

We conducted a thematic qualitative text analysis (Kuckartz, 2014) using an iterative consensual coding procedure that combined inductive and deductive approaches. We first independently coded responses, met to discuss coding decisions and the scheme, applying and refining codes inductively as patterns emerged. This process was repeated, with discrepancies resolved through discussion until we agreed upon both the coding scheme and the coding of the open-text responses.

Our typology and coding scheme contain a total of 15 different coding categories within four overarching categories of interest in dating members of the specified political party: unwilling, situationally dependent, willing, and a residual other category. To maintain analytic clarity and coding consistency, responses could receive multiple codes within the same overarching category but were not permitted to belong to multiple categories.

Here is a brief overview of the categories; for a more detailed description of the qualitative coding procedures and findings, see (Taflinger & Hudde, manuscript in progress).

Unwilling responses (31.9% of all responses) expressed refusal to date across party lines, with subcategories capturing negative stereotypes of the out-party (coded as unwilling: negative; 8.2% of all responses), perceived differences in values or lifestyles (unwilling: different; 15.7%), or anticipated conflict arising from political disagreement (unwilling: conflict; 4.4%). The final subcategory consisted of respondents who reported unwillingness, but their reason was unclear or idiosyncratic (unwilling: other; 3.5%). Some example open-text responses include (all responses are copied verbatim, including typographical errors), “I dont think i would get along with a republican woman. This is because republican are typically uncaring of others and i cannor relate to that.” (unwilling: negative); “They don’t believe in healthcare, gun control” (unwilling: different); “I don't want to get caught in the crossfire of politics.” (unwilling: conflict); “Not my kink!” (unwilling: other).

Situationally-dependent respondents (12.8%) indicated that they would or would not date a member of the political party under particular conditions or expressed ambivalence about dating a member of the political party.

The sub-categories included whether the particular conditions were political, e.g., including the potential partner’s political behaviour or opinions on specific partisan issues (situationally dependent: political; 8.2%), non-political aspects such as shared values or compatibility (situationally dependent: non-political; 3.4%), or contained unspecified or unclear contingencies (situationally dependent: other; 1.2%). Typical cues for this overarching category include formulations such as “it would depend on…”

or “I would, as long as…”. Examples include “Political affiliation doesn’t matter to me but if your entire personality (or even a major part of it) is about politics we probably won’t get along very well” (situationally dependent: political); “I would date a republican I don’t care for political parties as long as they have common sense” – meaning that the willingness is dependent on the person having common sense (situationally dependent: non-political); “It would depend on a lot of things” (situationally dependent: other).

Willing respondents (35.2%) were those who were open to dating members of the specified political party without specifying any contingencies, with some treating politics as irrelevant (willing: irrelevant; 22.3%), others emphasizing tolerance for differing views (willing: tolerant; 8.1%), or even framing cross-party dating positively (willing: positive; 2.9%), or did not clearly fit into any of the identified sub-categories (willing: other; 2.0%). Some examples are “I wouldnt care either way.” (willing: irrelevant); “I think they are entitled to vote however they want even if it disagrees with me. There is more to a relationship than politics.” (willing: tolerant); “The reason why is because the conversation would be amazing and indecisive to see who is right or wrong. Good way to look at both views.” (willing: positive); “I wouldn't mind dating a Democrat” (willing: other).

Finally, responses that did not provide a substantive stance were coded as other (20.1%), which included explicit uncertainty (other: don’t know; 2.9%), unclear or ambiguous responses (other: unclear; 10.4%), refusals to answer (other: no answer; 3.9%), and nonsensical or irrelevant text (other: nonsense; 3.0%). This structure provided an exhaustive and mutually exclusive typology of orientations. Examples include “I’m not sure” (other: don’t know); “It be divided it going to be agreement if say something wrong” (other: unclear); “oh uboub oubno oonobgu” (other: nonsense); “N/a” (other: no answer).

The instructions provided to the student assistants were about two pages long and included a description and some illustrative examples for each category. As an example, here is an excerpt of the manual. It includes the introduction of the “group of unwilling codes” and the instruction for the category of “unwilling different”.

> Group of unwilling-codes: Respondent conveys their desire not to date a member of the aforementioned political party. Their unwillingness may be explicit e.g.: I would not date a Democrat or implicit e.g.: because we wouldn’t get along. As a characteristic of this type of response, respondents do not mention any conditions under which they would date a member of the aforementioned party. They only mention reasons as to why they would not like to date them.
>
> […]

UNWILLING_DIFFERENT: Differing values/morals/beliefs/interests/lifestyles, e.g.: We would have different values; We are just too different; We like different things so it wouldn't work out; We wouldn't see eye to eye on things; I just disagree with what they believe.

The examples provided in the coding instructions included both illustrative examples written by the researchers and a few quotes from respondents. The full coding instruction is found in the Appendix.

3.3. How we use LLMs

Data protection and security

We did not upload any person-identifying information to the LLMs; we only uploaded the responses to one question, along with a newly created random ID variable. Before uploading, we read through all responses to ensure that they did not include any identifying information, such as email addresses. We did not upload any other information about the respondents, such as their gender, age, or region.

LLMs by different providers

We compare large language models from three major providers. First is OpenAI, best known for ChatGPT, the interface through which users access their GPT models. Second is Anthropic with its Claude language model family. While these two are US American companies, the third is Mistral, Europe's leading LLM company, which offers both open-source and proprietary models for commercial and research use. As a French company, Mistral is subject to stricter regulations concerning data privacy and related topics, and Mistral emphasizes data security and data storage in the European Union (Mistral, 2025).

Models with different levels of complexity

From these providers, we use models that differ in terms of their costs and complexity. Analyses were conducted in early 2026, using the latest models available at that time. Table 1 shows the list of all tested models.

Table 1: List of models compared

Provider / Model family	Model	Exact model name (API-code)
OpenAI / GPT	5 nano	gpt-5-nano-2025-08-07
	5 mini	gpt-5-mini-2025-08-07
	5.2	gpt-5.2-2025-12-11
Anthropic / Claude	4.5 Haiku	claude-haiku-4-5-20251001
	4.5 Sonnet	claude-sonnet-4-5-20250929
	4.5 Opus	claude-opus-4-5-20251101
Mistral	Small 4	mistral-small-2603
	Large 3	mistral-large-2512

Notes: For each provider, the models are listed in ascending order of their complexity and pricing.

For Anthropic / Claude, we compare three models, in ascending order of their complexity and pricing: 4.5 Haiku, 4.5 Sonnet, and 4.5 Opus. Coding all responses costs around $0.30 for 4.5 Haiku, compared to $1.00 for 4.5 Sonnet, and $1.70 for 4.5 Opus (in the 'standard' version with batching in groups of 10 and without training data. See sections on "batching" and "training data" below for more information).

For OpenAI / GPT, we compare three models, in ascending order of their complexity and pricing: GPT 5 nano, GPT 5 mini, and GPT 5.2. The cost for coding all responses is $0.22, $0.38, and $0.55, respectively.

For Mistral, we compare two models, in ascending order of their complexity and pricing: Mistral Small 4 and Mistral Large 3. The cost for coding all responses is $0.02 and $0.07, respectively. Unlike the models by OpenAI or Anthropic, both Mistral models are "open weight", meaning that the models themselves are publicly available for anyone to download and use, though running either model locally requires specialized hardware beyond standard research infrastructure. Since this paper focuses on approaches implementable with standard computers and statistical skills, we access both models via API (Application Programming Interface).

The duration for coding all answers was typically around an hour for OpenAI's models, compared to less than 10 minutes for models by Anthropic / Claude and Mistral. Such performance differences are only a snapshot and change considerably depending on how busy the servers are (e.g., by hour and weekday), or by changes in the provider's resources.

Workflow for the highest possible reproducibility

All models are accessed via API using the R programming environment. That means that our R code

accesses the survey responses (stored in an Excel file) and sends them, along with the coding instructions, as automated requests ("API calls") to the servers of the respective model provider (OpenAI, Anthropic, or Mistral). The model processes each response and returns the response together with a coding decision, which R aggregates and exports into an Excel file. R scripts to replicate the complete workflow are available here: https://osf.io/3nxrv/overview?view_only=90f6f3b4db9d4291b1550183bd1b66ec.

LLMs operate probabilistically by default, meaning the same input can produce different outputs across repeated calls, which hinders reproducibility. When accessed via API, some providers allow researchers to eliminate this randomness through the "temperature" parameter: at temperature zero, the model operates deterministically, always returning the same response to the same input. In practice, even at temperature zero, re-running the coding procedure returns the same codes for more than 99% of responses, but falls short of perfect reproducibility. This is because API responses depend on server-side factors beyond user control: when sending API calls in short succession, models can retain the content of previous calls in memory. For instance, when coding responses 10-19, the model still has the responses 1-9 in memory and might be affected by that memory. The amount retained varies with current server load and available computational resources, producing minor output variations even at zero temperature.

Where possible, we set the temperature to zero for all API calls. This is supported by Anthropic / Claude and Mistral, but not by the GPT-5 model family (including GPT-5 nano, GPT-5 mini, and GPT-5.2). For the example of GPT-5.2, we therefore assessed reproducibility by running the coding procedure three times and computing agreement across all three pairwise run comparisons. For GPT-5.2, pairwise consistency ranged from 92 to 94%.

Other parameters

OpenAI's GPT-5 has a "Verbosity Parameter", allowing one to "hint the model to be more or less expansive in its replies" (OpenAI, 2025). Our R script sets the value to "low" for "minimal prose".

Coding instructions for student assistants and LLMs

As in several previous studies, LLMs receive the same coding instructions as the human coders (Misiejuk et al., 2024; Suter & Meckel, 2024; Törnberg, 2023, 2025). The only difference is that the

version for LLMs included the following sentence: “You are an expert coder analyzing responses about dating across party lines.” Full coding instruction can be seen in the Appendix.

“Batching”

When using LLMs via API, responses can be submitted either individually (one response per API call) or grouped into a single request (“batching”), where multiple texts are processed together. Batching reduces both API calls and costs, but larger batches increase prompt complexity and may increase errors.

In our implementation, we varied batch sizes to assess practical trade-offs between efficiency and reliability. While batching generally improves efficiency[3], very large batches occasionally result in incomplete outputs (e.g., fewer coded responses returned than inputs), requiring additional post-processing. For example, sending 50 texts sometimes returned only 47 coded rows, creating missing data that requires manual correction. We generally used batches of 10 for our analyses, as this realised most of the efficiency gains and did not lead to errors for any of the models analysed.

Training data

Previous research has shown that the performance of LLMs can improve by including a sample of training data, namely hand-coded responses, directly in the API call — so-called in-context learning (Agarwal et al., 2024; Chae & Davidson, 2025; Tripp, 2025). We therefore compare two conditions: instructions-only versus including 200 randomly chosen hand-coded responses in each API call. To examine the effect of training data, we compared accuracy across the 1,219 answers that were *not* part of the training data. In practice, this approach requires a coded training dataset, but researchers developing a coding scheme will typically hand-code a subsample of their data during scheme

[3] To illustrate batching's efficiency gains, we use the example of Anthropic's Claude Sonnet 4.5, a model with high accuracy (see Results). With batch size 1—each response sent in a separate API call—coding all responses costs approximately $7.00 and takes circa 60 minutes. With a batch size of 10, costs drop to ca. $1 and time to less than 10 minutes. With a batch size of 25, costs fall further to $0.70 and 6 minutes.

development, so such a dataset arises naturally from the research process. Including training data requires the model to process substantially more text per call, which increases the cost. For Anthropic's Claude Sonnet 4.5, the cost rises from approximately $1.00 to approximately $3.30.

3.4. Analytical strategy

We outline an analytical strategy that compares coding accuracy (both with and without using additional training data), examines how LLM coding accuracy varies with student assistant agreement and benchmark categories, and evaluates whether substantive conclusions differ across benchmark, student-coded, and LLM-generated datasets. Across tests, coding accuracy is defined as full agreement with the benchmark codes defined by researchers. In cases of multiple codes (5.7% of responses in the benchmark data), only the identical list of codes is counted as agreement. Coding accuracy is measured with Cohen's Kappa, which captures agreement beyond chance and is widely used in both qualitative and quantitative studies (Kirsten et al., 2024; Kolesnyk & Khairova, 2022; Matter et al., 2024; Suter & Meckel, 2024).

First, we compare the accuracy of the coding by the various LLMs without training data with the coding by the student assistants.

Second, we compare the coding accuracy of LLMs with training and student assistants. To do so, we compare three conditions: LLM responses generated only using instructions, LLM responses generated with both instructions and including 200 randomly chosen hand-coded responses in each API call, so-called in-context learning (Agarwal et al., 2024; Chae & Davidson, 2025; Tripp, 2025), and the responses of student assistants. To examine the effect of training data, we restrict the sample from the full 1,419 to compared accuracy only across the 1,219 answers that were *not* part of the training data.

Third, we examine how accuracy varies with the degree of ambiguity of a certain text. We operationalise this as student agreement with the benchmark: a case is *clear* if all human coders agree, meaning that both student assistants assign the same code(s) that the two researchers have agreed upon for the benchmark dataset (55.6%). In contrast, a case is *ambiguous* if there is disagreement among the human coders, meaning that at least one of the student assistants assigned different code(s) to a response than the researcher-coded benchmark dataset (44.4%).

Fourth, we examine variation in performance across text codes by evaluating results separately for each of the 15 codes in the benchmark dataset. For these analyses, when the benchmark dataset has multiple

codes, they are grouped into the category “multiple” (5.7%).

Fifth, we explore a use case in which we use the coded data to answer an applied research question, examining whether conclusions would differ depending on whether the benchmark data, the version coded by student assistants, or the LLM codes were used. The open-text questions were gathered within a survey that contained a survey experiment. In the survey experiment, respondents were shown several fictional online dating profiles, some of which included information on the partisanship of the person in the profile. We tested whether reactions to out-partisans — relative to a baseline profile with no partisan information — differ between respondents who explicitly express openness to dating across party lines in their open-ended responses and those who explicitly express unwillingness to do so.

The analyses are restricted to respondents who are partisans, including respondents who lean towards or typically vote for the Democratic or Republican Party. The analytical sample includes 1,060 individuals with 4,240 observations (Taflinger & Hudde, Forthcoming). We estimate regression models with fixed effects at the individual level. The outcome variable is an index of respondents’ romantic interest (mean = 2.20, standard deviation = 1.27, range = 0 to 4), and the central predictor is an indicator of a political match, with two values: political disagreement (value 1), and no political information in the profile (value 0). The regression models include an interaction term between political disagreement and the categorical variable openness to dating a member of another political party, coded as “willing” (as baseline category), “situationally dependent”, “unwilling”, or “other”.

4. Results

4.1. Overall comparison

Figure 1 plots the level of agreement with the benchmark dataset, using Cohen’s Kappa as a metric. For this coding task, the two student assistants achieve Kappa values of 0.65 and 0.66 against the benchmark, indicating substantial agreement (Landis & Koch, 1977). The intercoder agreement between the two student assistants is somewhat lower, with a Kappa of 0.60, indicating moderate to substantial agreement.

Comparing LLM performance across models from the same provider shows a clear pattern: the more advanced the model, the greater the accuracy.

Some of the less advanced models perform considerably worse than the student assistants. This includes OpenAI’s GPT-5 nano and Mistral Small 4. In contrast, several more advanced models significantly and substantially outperform the student assistants. This includes OpenAI’s GPT 5.2 and Claude models 4.5

Sonnet and 4.5 Opus. The estimated accuracy of Mistral's Large 3-model is also higher than that of both student assistants, but the difference is not statistically significant. The best-performing model, Claude 4.5 Opus, reaches a Kappa value of 0.72, followed by Claude 4.5 Sonnet with a Kappa value of 0.71. Hence, models by Anthropic / Claude show the highest performance.

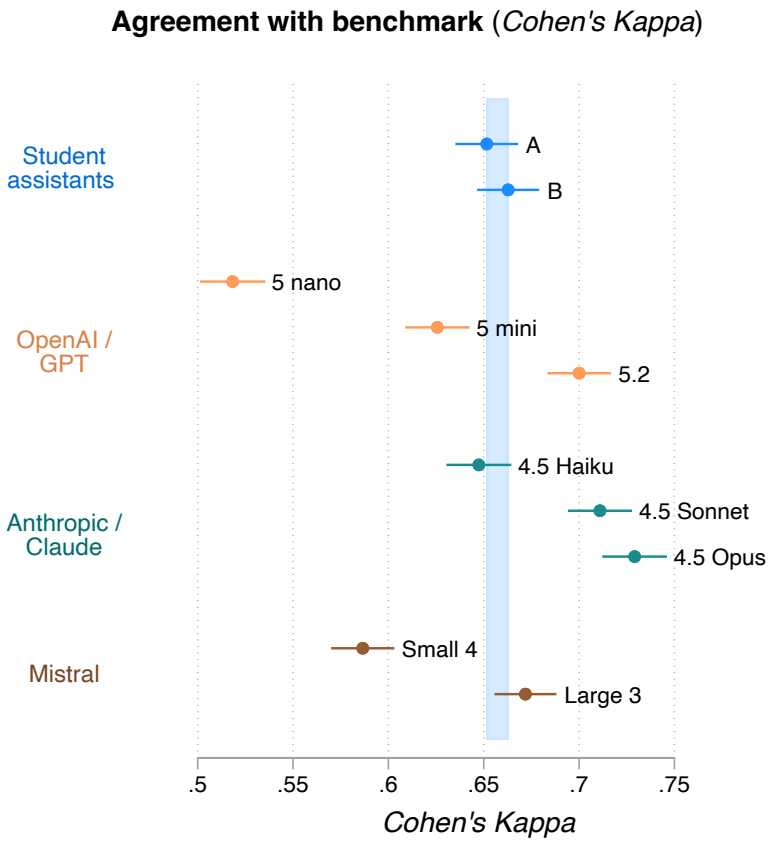

Figure 1: Comparing the coding accuracy of student assistants and LLMs (n=1,419).

4.2. Training data

Figure 2 reports results for LLMs provided with both instructions and training data. All estimates shown in this figure — including the student assistants and the models without training data — compare coding accuracy only for the 1,219 responses that are *not* part of the training data. Results show that all models generally perform better with training data. The increase in accuracy is substantial in several cases, and simpler models with training data partly outperform the more advanced models without training data.

One model does not benefit from training data, namely GPT-5 nano. This is the model with the poorest

performance in the version without training data. This suggests that some models might be too simple for this task, regardless of whether training data is included.

The ranking of models' performance remains unchanged with or without training data, with Claude 4.5 Opus leading, closely followed by 4.5 Sonnet. With training data, Mistral Large 3 also substantially outperforms the reference group of student assistants without training data.

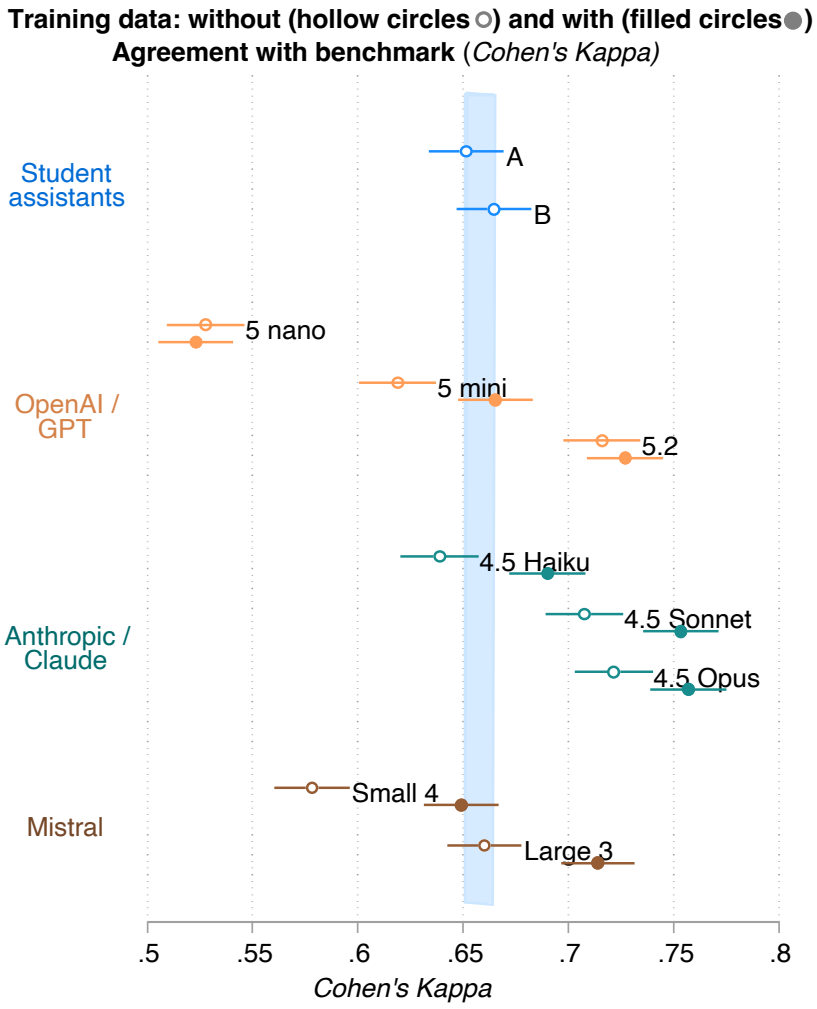

Figure 2: Comparing the performance of LLMs with and without training data (n=1,219).

4.3. Variation by ambiguity of response

Next, we test how accurate LLM-coding is, depending on coding ambiguity. Figure 3 shows coding accuracy of LLMs, separately for cases on which all human coders agreed (55.6%) or not (44.4%)[4], and for versions with and without training data.

Results generally show that for cases where all human coders agree, LLMs generally do so at very high rates as well. The reverse is equally true: where at least one student assistant does not code in agreement with the benchmark coding, LLMs are also less likely to do so. The hierarchy between the models is largely consistent between clear and ambiguous cases: Claude Opus and Claude Sonnet have the highest performance in both groups, while GPT 5 nano and Mistral Small 4 have the lowest performance. The performance improvements from using training data are considerably greater in the group of ambiguous cases.

[4] For 14% of responses, student assistant A coded in agreement with the benchmark dataset, but student assistant B did not; for another 13%, student assistant B coded in agreement with the benchmark dataset but student assistant A did not; and for 17% of responses, neither of the student assistants coded in agreement with the benchmark dataset.

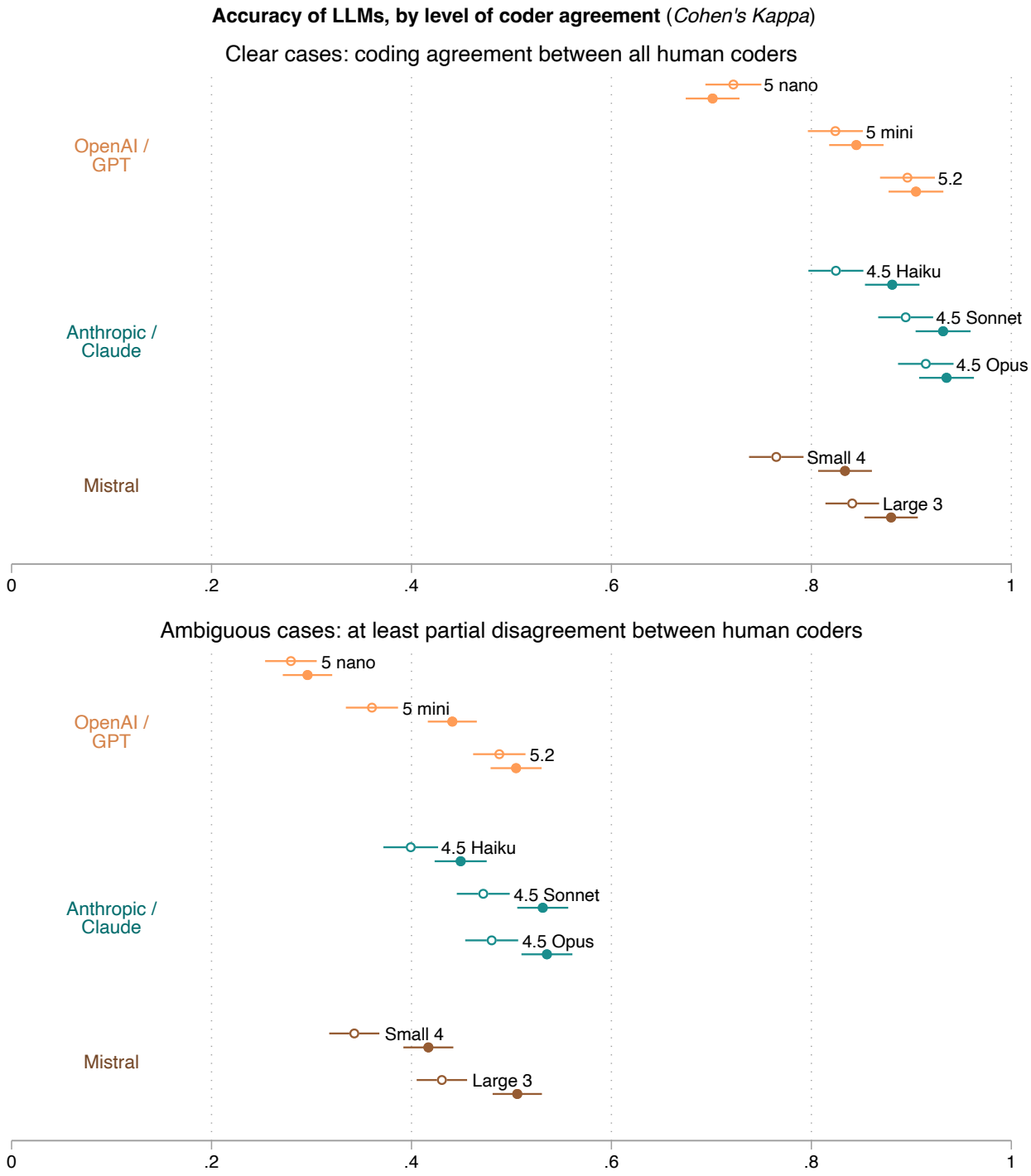

Figure 3: Comparing coding accuracy depending on how clear vs. ambiguous cases are. Hollow circles without training data; filled circles: with training data.

4.4. Code-specific agreement

Overall accuracy alone is insufficient for evaluating coding quality. Even moderate misclassification rates could be problematic if LLMs systematically fail to identify specific categories while performing well on others. Therefore, we examine accuracy separately for each code, comparing models with and without training data. The figure is shown in the Appendix (Figures A1 and A2).

Three patterns emerge. First, codes where both student assistants achieve high accuracy also yield high LLM accuracy. Second, none of the models with the highest performance overall (GPT-5.2, Claude Sonnet 4.5, and Opus 4.5) perform substantially worse than both students on any individual code. For Mistral Large 3, this pattern holds for most codes, with some exceptions where it falls below both students. Third, training data primarily improves performance on codes that initially showed comparatively poor accuracy. This suggests training data functions to stabilise performance across the full range of response categories rather than uniformly boosting all codes. In sum, these code-specific analyses confirm the generally high and consistent performance of advanced models across our multi-dimensional coding scheme.

4.5. Prediction of behaviour in the survey experiment

Beyond coding accuracy, a crucial test is whether different coding methods yield equivalent substantive conclusions when using the coded data in further analyses. We examine whether text-based openness categories predict actual behaviour in our survey experiment, comparing results across all coding methods.

Figure 4 presents coefficient estimates from models predicting romantic interest depending on political disagreement between the respondent and the dating profile, using text responses coded by each method. Based on the benchmark dataset, participants coded as "unwilling" show strong negative reactions to political mismatch, those coded as "situationally dependent" show moderate reactions, and those coded as "willing" or "other" show negligible reactions. While the internally diverse group of “other” might not be particularly interesting substantively, it could be worrisome if LLMs showed vastly different results for that group than for the others.

These estimated reactions are almost identical, regardless of coding method (see Figure A3 in the Appendix for a version with training data, which shows almost unchanged results). While overall coding accuracy clearly improved with LLM model complexity, even the mid-range models performed very well for this predictive task. In sum, even mid-range LLMs produced codes with predictive validity that is equivalent to datasets coded by student assistants or research experts for this substantive research question.

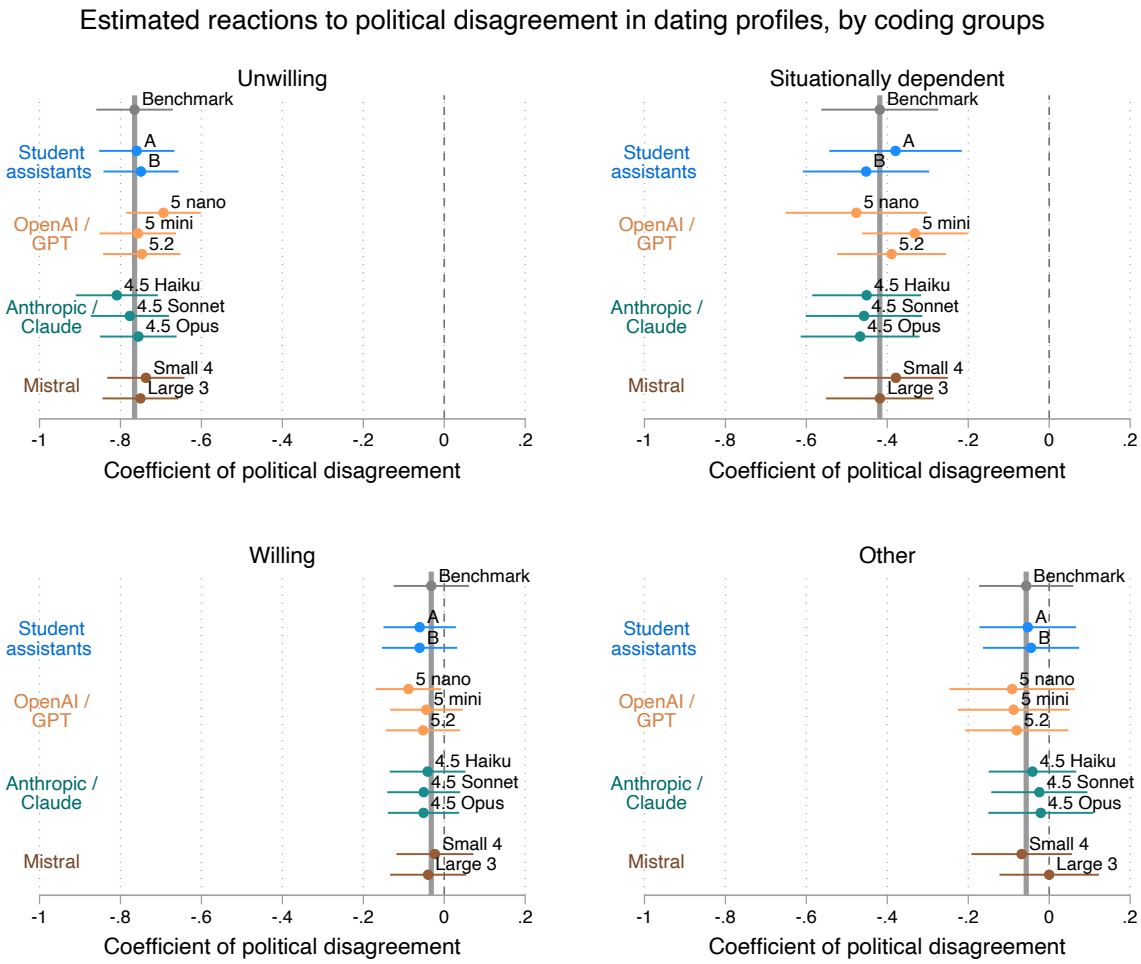

Figure 4: Coefficient estimates from models predicting romantic interest depending on political disagreement between the respondent and the dating profile, using text responses coded by each method.

5. DISCUSSION

In our study of open-response data from 1,419 respondents regarding interest in dating across party lines, we tested how LLM-based coding compares to that of student assistants, where the accuracy of coding was measured as similarity to a researcher-coded benchmark dataset. Our results show that LLMs can classify open-text responses using a predefined scheme developed by researchers, achieving performance comparable to that of human coders. However, model complexity matters: less complex models underperformed compared to student assistants, while the most advanced models from OpenAI

/ GPT, Anthropic / Claude, and Mistral outperformed the student assistants. Further, providing LLMs with training data generally improved their performance. A closer look at *where* LLMs succeed and fail reveals a reassuring pattern of similarity between student assistants and LLMs: Where student assistants reached consensus with the benchmark, LLMs also coded with high accuracy; where student assistants diverged, LLMs were more likely to diverge as well. Finally, applying the LLM codes to answer a substantive research question demonstrated that conclusions did not change whether we used the codes from the researchers, the student assistants, or the LLMs.

While previous research on LLM-assisted coding has mainly focused on social media texts, we consider responses to open-ended survey questions a particularly relevant and promising use case for social scientists. Such questions allow respondents to share their thoughts and narratives in their own words and without relying on predefined response categories (Fielding et al., 2013; Foddy & Foddy, 1993; Reja et al., 2003). Further, answers to open-ended questions can be analysed jointly with other information collected in the survey (Fielding et al., 2013). Finally, open-ended questions are easy to collect as they can be readily integrated into well-established workflows in the social sciences and added to existing studies, such as the European Social Survey, General Social Survey, or German Socio-Economic Panel. With previously available methods, researchers had to code all responses by hand or rely on computational approaches that require advanced computing skills and/or only scratch the surface of the data (Than et al., 2025). LLMs, in contrast, can deal with higher levels of complexity than previous approaches while being easy to use. These findings highlight that LLMs can contribute to social science research by streamlining the research process; analyses that were previously conducted by humans are made more efficient, so that it is possible to scale up research in cases where corpora of text are too large to code by hand (Abramson et al., forthcoming).

More broadly, advances in the coding and analysis of text data contribute to bridging qualitative and quantitative approaches in the social sciences. We first conducted thematic qualitative text analysis, leaving the central analytical, interpretive, and creative tasks with the researchers. Next, the LLM applied the finished scheme at scale. Thereby, LLMs substantially reduce the resources needed to code large amounts of text data. For a task similar to ours, coding even 10,000 responses will cost from under one dollar to a few dozen dollars, depending on the model choice and whether training data are used. Once a typology developed with qualitative rigour is linked to a large representative sample, population prevalences of the categories can be estimated, group differences (e.g., by gender, education, age, time, and space) can be examined, and the categories can be used as variables in inferential models — as predictors, outcomes, or interaction terms — in the same way quantitative sociologists already use latent class memberships or other constructed measures.

Our study has several limitations that also point to directions for future research. As with any study, we had one specific test case: dating members of another party among young US Americans. Performance may differ on different types of texts, from other respondent populations, in other languages, or on coding schemes with a different mix of interpretive demands. Further, we evaluate accuracy against a researcher-created coding scheme and benchmark dataset. Both reflect the analytical judgments of a specific research team, and other researchers might draw different categorical distinctions or assign different codes to borderline cases. All these factors underscore the need for further studies from a variety of research teams and settings to help establish greater confidence in LLM capabilities and clarify the conditions under which these results generalize (Alvero et al., 2026).

Suggestions for applied research

Before offering concrete recommendations, we want to highlight two important considerations. First, the landscape of available LLMs is changing rapidly, and specific recommendations, e.g., on model choice, may change as new releases appear. Second, our recommendations necessarily involve judgment calls about trade-offs that depend on researchers' priorities and cannot be resolved by empirical evidence alone, such as what level of accuracy is sufficient, or how much validation is warranted.

We organise the recommendations as answers to questions that an applied researcher planning a similar workflow is likely to ask. Overall, we aim to be general enough so that they apply to a sufficiently large set of application cases and specific enough to be actionable.

How can one set up a workflow? Concerning the coding scheme and instructions, we generally gave identical instructions to student assistants and LLMs. The exceptions were that LLMs were additionally provided with the sentence "You are an expert coder analyzing responses about dating across party lines." We also used specific formatting for the codes for LLMs, namely, we spelled the codes in all capital letters to distinguish from instruction text and used "_" instead of spaces for easier computational processing. For accessing the LLMs, we recommend accessing through their APIs from R (or a different software, such as Python) rather than through browser chat interfaces. API access is more efficient for repeated tasks, is more reproducible, and makes it easy to switch between providers and models. This setup runs on a standard laptop and requires no specialised technical skill; the replication code we publish can easily be adapted to new cases. Thereby, the workflow we propose is easily accessible to applied social scientists and beyond.

When does it make sense to use LLMs versus code all by hand? While it is not possible to define a precise threshold, efficiency gains from LLM-coding might be relatively low for samples smaller than

1,000 responses, but grow considerably thereafter. Ultimately, it becomes a trade-off between the amount of time it would take to code past saturation and the time it would take to establish the LLM workflow. The biggest cost factor in the analytical process is researchers' labor time for one-time efforts, from identifying the categories and developing the codebook to establishing the R workflow. These efforts are the same, no matter whether the sample includes 1,000, 10,000, or more pieces of text. In contrast, the financial cost of LLM-coding is minimal.

How many responses do researchers need to code before they can finalize the coding scheme and delegate further coding to LLMs? This is, as in qualitative research more broadly, a substantive judgment call rather than a fixed technical threshold. It depends on when researchers have reached sufficient saturation, meaning that they do not come across new cases that would require adjusting the coding scheme in relevant ways, which depends on the degree of variability in the data (Saunders et al., 2018). In a case similar to ours, researchers can likely expect that it will require at least a few hundred cases.

How can applied researchers assess whether LLM-based coding achieves sufficient performance in their specific context? One approach is to hand-code a random sample of responses — say, 200 cases — and calculate Cohen's Kappa against the LLM output. Additionally, researchers can screen the coded output by randomly sorting the output and reading through a sample of responses alongside their assigned codes to identify systematic problems, such as cases where the LLM fails to apply the coding scheme correctly, produces implausible codes, or behaves erratically. If no red flags emerge, researchers can proceed with reasonable confidence that the coding is fit for purpose.

Should researchers provide the LLMs with training data? Our analyses suggest that including a sample of 200 hand-coded responses as training data improved overall accuracy, especially on codes that initially showed comparatively poor accuracy. Given that researchers have to read and analyse a subsample of all responses by hand when developing the coding scheme, including training data is relatively straightforward. While training data increases the models' cost (e.g., coding all responses using Claude 4.5 Sonnet costs around $1 without training data and around $3 with training data), these cost differences will be negligible in many cases. Therefore, we recommend using training data as a default.

Which provider and model should applied researchers use? Our results show that for a coding task like ours, model complexity matters: we generally recommend using each provider's more advanced models for comparable tasks. The choice of provider — OpenAI, Anthropic, or Mistral — is less important, as differences between flagship models were relatively small. That said, Anthropic's Opus and Sonnet

models achieved the best overall accuracy. Further, both Anthropic and Mistral allow for a more reproducible workflow compared to OpenAI (cf. section 3.3). Mistral's flagship model slightly lags behind the other two providers, but still matched student assistant performance without training data and outperformed them with it. Researchers who prefer a European provider, for instance, due to Mistral's being subject to stricter EU regulation, and the provider's emphasis on data security and EU-based data storage, can do so with only comparatively small performance sacrifices. As a further advantage, Mistral's models are remarkably low-cost.

How can researchers ensure data protection and security? The most important aspect is to ensure that no person-identifying information is uploaded to the LLMs. In our case, we checked to be sure that the open-text responses did not contain any person-identifying information and only uploaded the responses to one question together with a newly created, random ID variable. We did not upload any other information on the respondents, such as their gender, age, or region. While all providers covered in this study say that data sent via API is not used for model training, researchers might require options with higher levels of data protection in some circumstances. Such options exist, but involve trade-offs. For many, using Mistral through the API access described above will represent the right compromise between data protection and model quality. Higher data security guarantees may be needed in some cases, e.g., when texts contain sensitive or potentially person-identifying information. Several academic institutions offer access to LLMs that run entirely on their own servers, eliminating any data transfer to the model provider. A notable example is Chat AI, operated by German public academic institutions (GWDG, 2026). Among the models that are offered and run on their local servers is Mistral Large 3, which showed relatively good performance in our study. However, these alternatives have drawbacks: they currently tend to be less user-friendly, lack access to the best-performing models, and might not be available to researchers from all countries and institutions.

Taken together, this study demonstrates the potential of LLMs for qualitative text coding at scale, highlights open-ended survey questions as a particularly promising use case, and offers practical guidance for applied social scientists. Because LLMs can handle higher levels of interpretive complexity than previous computational approaches while remaining accessible to researchers without specialist technical skills, they open up new analytical possibilities for text data more broadly, and may herald a golden era for open-ended questions more specifically.

Acknowledgments

We would like to thank Marita Jacob, Emilia Kmiotek-Meier, and Christian Czymara for their valuable feedback.

Funding information: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2126/2 – 390838866.

Bibliography

Abramson, C. M., Prendergast, T., Li, Z., & Dohan, D. (forthcoming). Qualitative Research in an Era of AI: A Pragmatic Approach to Data Analysis, Workflow, and Computation. *Annual Review of Sociology*, *52*. https://doi.org/10.48550/arXiv.2509.12503

Agarwal, R., Singh, A., Zhang, L., Bohnet, B., Rosias, L., Chan, S., Zhang, B., Anand, A., Abbas, Z., Nova, A., Co-Reyes, J. D., Chu, E., Behbahani, F., Faust, A., & Larochelle, H. (2024). Many-Shot In-Context Learning. *Advances in Neural Information Processing Systems*, *37*, 76930–76966. https://doi.org/10.52202/079017-2447

Alvero, A. J., Stoltz, D. S., Stuhler, O., & Taylor, M. A. (2026). Generative AI in Sociological Research: State of the Discipline. *Sociological Science*, *13*, 45–62. https://doi.org/10.15195/v13.a3

Bernhard-Harrer, J., & Pfaff, K. (2025). Question form Matters: Examining Trust in Government Through Open and Closed Survey Items. *Journal of Survey Statistics and Methodology*, *13*(4), 370–392. https://doi.org/10.1093/jssam/smaf014

Breznau, N., & Nguyen, H. H. V. (2026). *An Introduction to Generative Artificial Intelligence for Academics*. F1000Research. https://doi.org/10.12688/f1000research.166513.2

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Burnham, M. (2024). *What is Sentiment Meant to Mean to Language Models?* (arXiv:2405.02454). arXiv. https://doi.org/10.48550/arXiv.2405.02454

Cabanillas-García, J. L., Sánchez-Gómez, M. C., & Del Brío-Alonso, I. (2025). *Voices of Researchers: Ethics and Artificial Intelligence in Qualitative Inquiry*.

Chae, Y., & Davidson, T. (2025). Large Language Models for Text Classification: From Zero-Shot Learning to Instruction-Tuning. *Sociological Methods & Research*, 00491241251325243. https://doi.org/10.1177/00491241251325243

Chatzichristos, G. (2025). Qualitative Research in the Era of AI: A Return to Positivism or a New Paradigm? *International Journal of Qualitative Methods*, *24*, 16094069251337583. https://doi.org/10.1177/16094069251337583

Chew, R., Bollenbacher, J., Wenger, M., Speer, J., & Kim, A. (2023). *LLM-Assisted Content Analysis: Using Large Language Models to Support Deductive Coding* (arXiv:2306.14924). arXiv. https://doi.org/10.48550/arXiv.2306.14924

Davidson, T., & Karell, D. (2025). Integrating Generative Artificial Intelligence into Social Science Research: Measurement, Prompting, and Simulation. *Sociological Methods & Research*, *54*(3), 775–793. https://doi.org/10.1177/00491241251339184

Fielding, J., Fielding, N., & Hughes, G. (2013). Opening up open-ended survey data using qualitative software. *Quality & Quantity*, *47*(6), 3261–3276. https://doi.org/10.1007/s11135-012-9716-1

Foddy, W., & Foddy, W. H. (1993). *Constructing questions for interviews and questionnaires: Theory and practice in social research*. Cambridge University Press.

Gilardi, F., Alizadeh, M., & Kubli, M. (2023). ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, *120*(30), e2305016120. https://doi.org/10.1073/pnas.2305016120

GWDG. (2026). *Chat AI*. Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen. Documentation for HPC. https://docs.hpc.gwdg.de/services/ai-services/chat-ai/index.html

Hayes, A. S. (2025). "Conversing" With Qualitative Data: Enhancing Qualitative Research Through Large Language Models (LLMs). *International Journal of Qualitative Methods*, *24*, 16094069251322346. https://doi.org/10.1177/16094069251322346

Heseltine, M., & Clemm von Hohenberg, B. (2024). Large language models as a substitute for human experts in annotating political text. *Research & Politics*, *11*(1), 20531680241236239. https://doi.org/10.1177/20531680241236239

Heyde, L. von der, Haensch, A.-C., Weiß, B., & Daikeler, J. (2025). *AIn't Nothing But a Survey? Using Large Language Models for Coding German Open-Ended Survey Responses on Survey Motivation* (arXiv:2506.14634). arXiv. https://doi.org/10.48550/arXiv.2506.14634

Ivanusch, C. (2024). Where do parties talk about what? Party issue salience across communication channels. *West European Politics*, *0*(0), 1–27. https://doi.org/10.1080/01402382.2024.2322234

Jiang, Y., Ko-Wong, L., & Valdovinos Gutierrez, I. (2025). The Feasibility and Comparability of Using Artificial Intelligence for Qualitative Data Analysis in Equity-Focused Research. *Educational*

Researcher, *54*(3), 153–163. https://doi.org/10.3102/0013189X251314821

Kabir, S. M. A., Ali, F., Ahmed, R. L., & Sulaiman-Hill, R. (2025). Exploring the Use of AI in Qualitative Data Analysis: Comparing Manual Processing with Avidnote for Theme Generation. *International Journal of Qualitative Methods*, *24*, 16094069251336810. https://doi.org/10.1177/16094069251336810

Kirsten, E., Buckmann, A., Mhaidli, A., & Becker, S. (2024). *Decoding Complexity: Exploring Human-AI Concordance in Qualitative Coding* (arXiv:2403.06607). arXiv. https://doi.org/10.48550/arXiv.2403.06607

Kolesnyk, A. S., & Khairova, N. F. (2022). Justification for the Use of Cohen's Kappa Statistic in Experimental Studies of NLP and Text Mining. *Cybernetics and Systems Analysis*, *58*(2), 280–288. https://doi.org/10.1007/s10559-022-00460-3

Krähnke, U., Pehl, T., & Dresing, T. (2025). *Hybride Interpretation textbasierter Daten mit dialogisch integrierten LLMs: Zur Nutzung generativer KI in der qualitativen Forschung*. https://www.ssoar.info/ssoar/handle/document/99389

Kravets-Meinke, D., Schmid-Petri, H., Niemann, S., & Schmid, U. (2025). *Generative Large Language Models (gLLMs) in Content Analysis: A Practical Guide for Communication Research* (arXiv:2510.24337). arXiv. https://doi.org/10.48550/arXiv.2510.24337

Kuckartz, U. (2014). *Qualitative text analysis: A guide to methods, practice & using software* (A. McWhertor, Trans.). Sage.

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159–174. https://doi.org/10.2307/2529310

Lee, V. V., Lubbe, S. C. C. van der, Goh, L. H., & Valderas, J. M. (2024). Harnessing ChatGPT for Thematic Analysis: Are We Ready? *Journal of Medical Internet Research*, *26*(1), e54974. https://doi.org/10.2196/54974

Macanovic, A., & Przepiorka, W. (2024). A systematic evaluation of text mining methods for short texts: Mapping individuals' internal states from online posts. *Behavior Research Methods*. https://doi.org/10.3758/s13428-024-02381-9

Matter, D., Schirmer, M., Grinberg, N., & Pfeffer, J. (2024). *Close to Human-Level Agreement:*

Tracing Journeys of Violent Speech in Incel Posts with GPT-4-Enhanced Annotations (arXiv:2401.02001). arXiv. https://doi.org/10.48550/arXiv.2401.02001

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

Mens, G. L., & Gallego, A. (2023). *Scaling Political Texts with ChatGPT* (arXiv:2311.16639). arXiv. https://doi.org/10.48550/arXiv.2311.16639

Misiejuk, K., Kaliisa, R., & Scianna, J. (2024). Augmenting assessment with AI coding of online student discourse: A question of reliability. *Computers and Education: Artificial Intelligence*, *6*, 100216. https://doi.org/10.1016/j.caeai.2024.100216

Mistral. (2025). *Where do you store my data or my Organization's data? | Mistral AI - Help Center*. Mistral. https://help.mistral.ai/en/articles/347629-where-do-you-store-my-data-or-my-organization-s-data

Morgan, D. L. (2023). Exploring the Use of Artificial Intelligence for Qualitative Data Analysis: The Case of ChatGPT. *International Journal of Qualitative Methods*, *22*, 16094069231211248. https://doi.org/10.1177/16094069231211248

OpenAI. (2025). *GPT-5 New Params and Tools*. OpenAI Developers. https://developers.openai.com/cookbook/examples/gpt-5/gpt-5_new_params_and_tools

Ornstein, J. T., Blasingame, E. N., & Truscott, J. S. (2025). How to train your stochastic parrot: Large language models for political texts. *Political Science Research and Methods*, *13*(2), 264–281. https://doi.org/10.1017/psrm.2024.64

Paulus, T., Lester, J. N., & Davis, C. (2025). The construction of the role of AI in qualitative data analysis in the social sciences. *AI & SOCIETY*. https://doi.org/10.1007/s00146-025-02488-3

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Rathje, S., Mirea, D.-M., Sucholutsky, I., Marjieh, R., Robertson, C., & Bavel, J. J. V. (2023). *GPT is an effective tool for multilingual psychological text analysis*. https://doi.org/10.31234/osf.io/sekf5

Reja, U., Manfreda, K. L., Hlebec, V., & Vehovar, V. (2003). Open-ended vs. Close-ended questions

in web questionnaires. *Developments in Applied Statistics*, *19*(1), 159–177.

Saunders, B., Sim, J., Kingstone, T., Baker, S., Waterfield, J., Bartlam, B., Burroughs, H., & Jinks, C. (2018). Saturation in qualitative research: Exploring its conceptualization and operationalization. *Quality & Quantity*, *52*(4), 1893–1907. https://doi.org/10.1007/s11135-017-0574-8

Schonlau, M., & Couper, M. P. (2016). Semi-automated categorization of open-ended questions. *Survey Research Methods*, *10*(2), 143–152. https://doi.org/10.18148/srm/2016.v10i2.6213

Suter, V., & Meckel, M. (2024). Using GPT-4 for Text Analysis: Insights from English and German Language News Classification Tasks. *Proceedings of the International AAAI Conference on Web and Social Media*. https://workshop-proceedings.icwsm.org/pdf/2024_31.pdf

Taflinger, S., & Hudde, A. (Forthcoming). Why do young US Americans avoid cross-partisan dating? A closer look at mediators and variation by gender and party. *European Sociological Review*. https://doi.org/10.1093/esr/jcag020

Tai, R. H., Bentley, L. R., Xia, X., Sitt, J. M., Fankhauser, S. C., Chicas-Mosier, A. M., & Monteith, B. G. (2024). An Examination of the Use of Large Language Models to Aid Analysis of Textual Data. *International Journal of Qualitative Methods*, *23*, 16094069241231168. https://doi.org/10.1177/16094069241231168

Than, N., Fan, L., Law, T., Nelson, L. K., & McCall, L. (2025). Updating "The Future of Coding": Qualitative Coding with Generative Large Language Models. *Sociological Methods & Research*, *54*(3), 849–888. https://doi.org/10.1177/00491241251339188

Törnberg, P. (2023). *ChatGPT-4 Outperforms Experts and Crowd Workers in Annotating Political Twitter Messages with Zero-Shot Learning* (arXiv:2304.06588). arXiv. https://doi.org/10.48550/arXiv.2304.06588

Törnberg, P. (2024). *Best Practices for Text Annotation with Large Language Models* (arXiv:2402.05129). arXiv. https://doi.org/10.48550/arXiv.2402.05129

Törnberg, P. (2025). Large Language Models Outperform Expert Coders and Supervised Classifiers at Annotating Political Social Media Messages. *Social Science Computer Review*, *43*(6), 1181–1195. https://doi.org/10.1177/08944393241286471

Tripp, A. (2025). Benchmarking AI and human text classifications in the context of newspaper frames:

A multi-label LLM classification approach. *Research & Politics*, *12*(2), 20531680251332353. https://doi.org/10.1177/20531680251332353

Wachinger, J., Bärnighausen, K., Schäfer, L. N., Scott, K., & McMahon, S. A. (2025). Prompts, Pearls, Imperfections: Comparing ChatGPT and a Human Researcher in Qualitative Data Analysis. *Qualitative Health Research*, *35*(9), 951–966. https://doi.org/10.1177/10497323241244669

Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can Large Language Models Transform Computational Social Science? *Computational Linguistics*, 1–55. https://doi.org/10.1162/coli_a_00502

Appendix

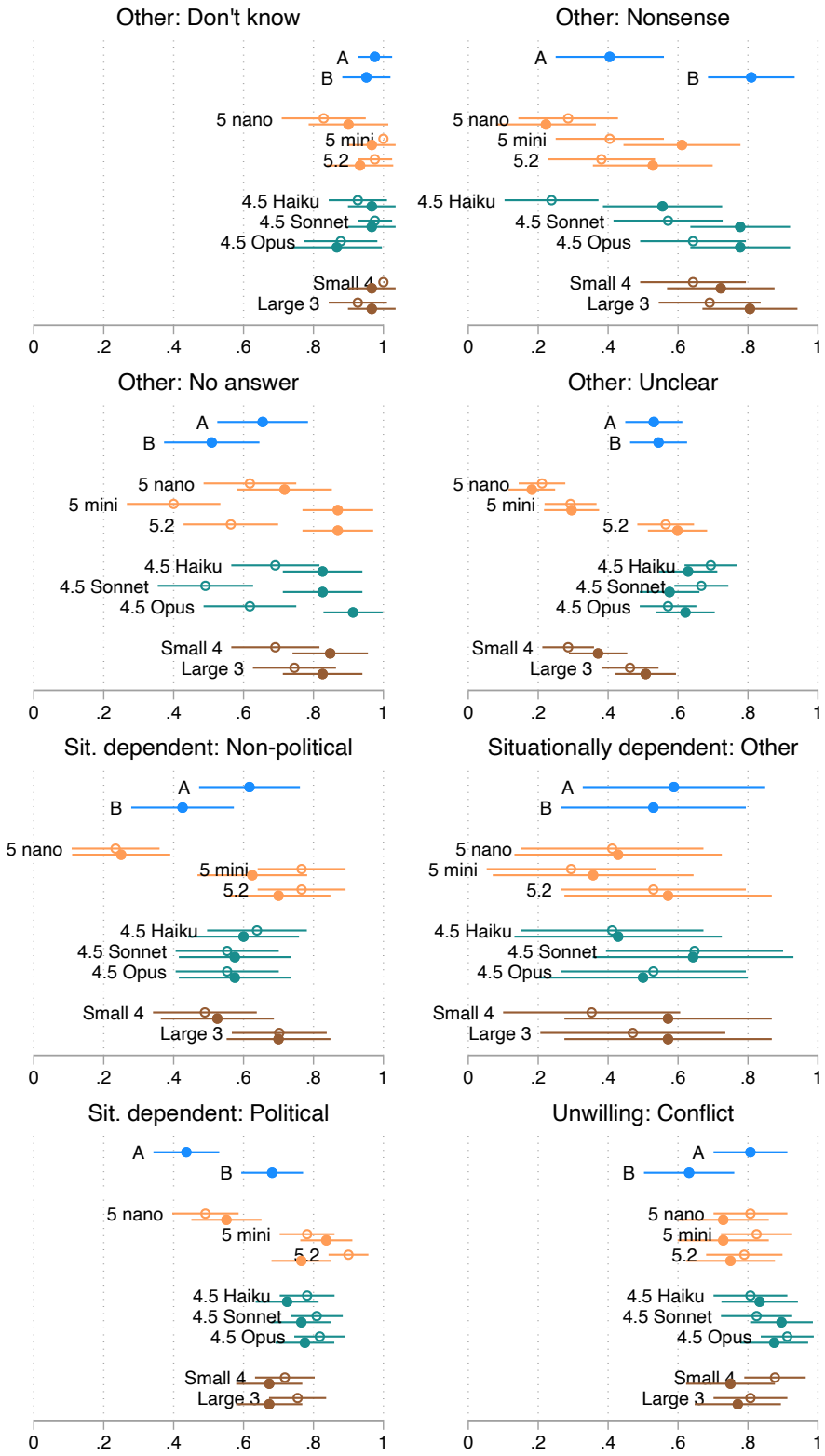

Figure A1: Code-specific agreement, without training data (hollow circles) and with training data (filled circles). Figure A1 shows the first 8 of all 16 code categories; Figure A2 the remaining code categories.

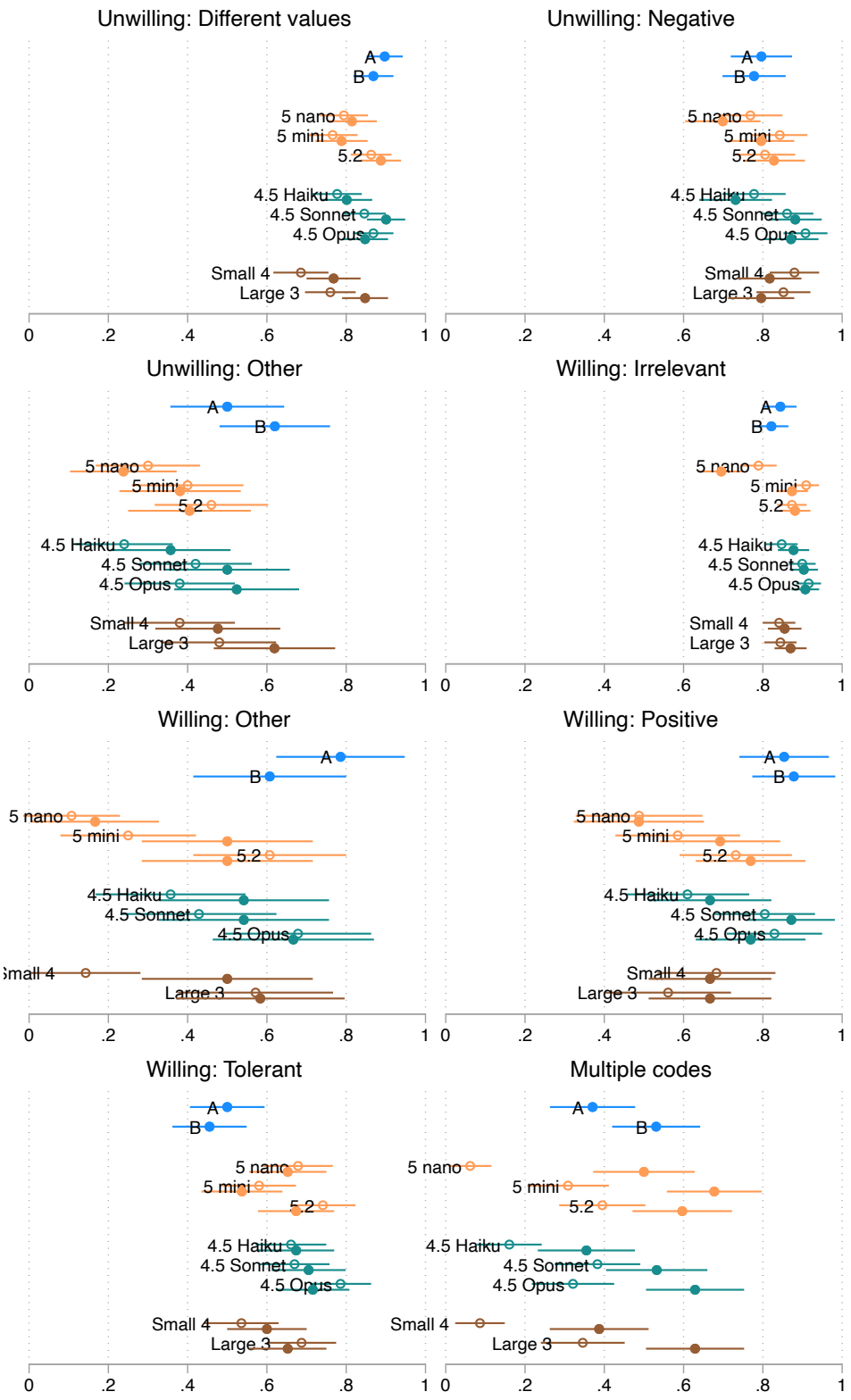

Figure A2: Code-specific agreement, without training data (hollow circles) and with training data (filled circles). Figure A1 shows the first 8 of all 16 code categories; Figure A2 the remaining code categories.

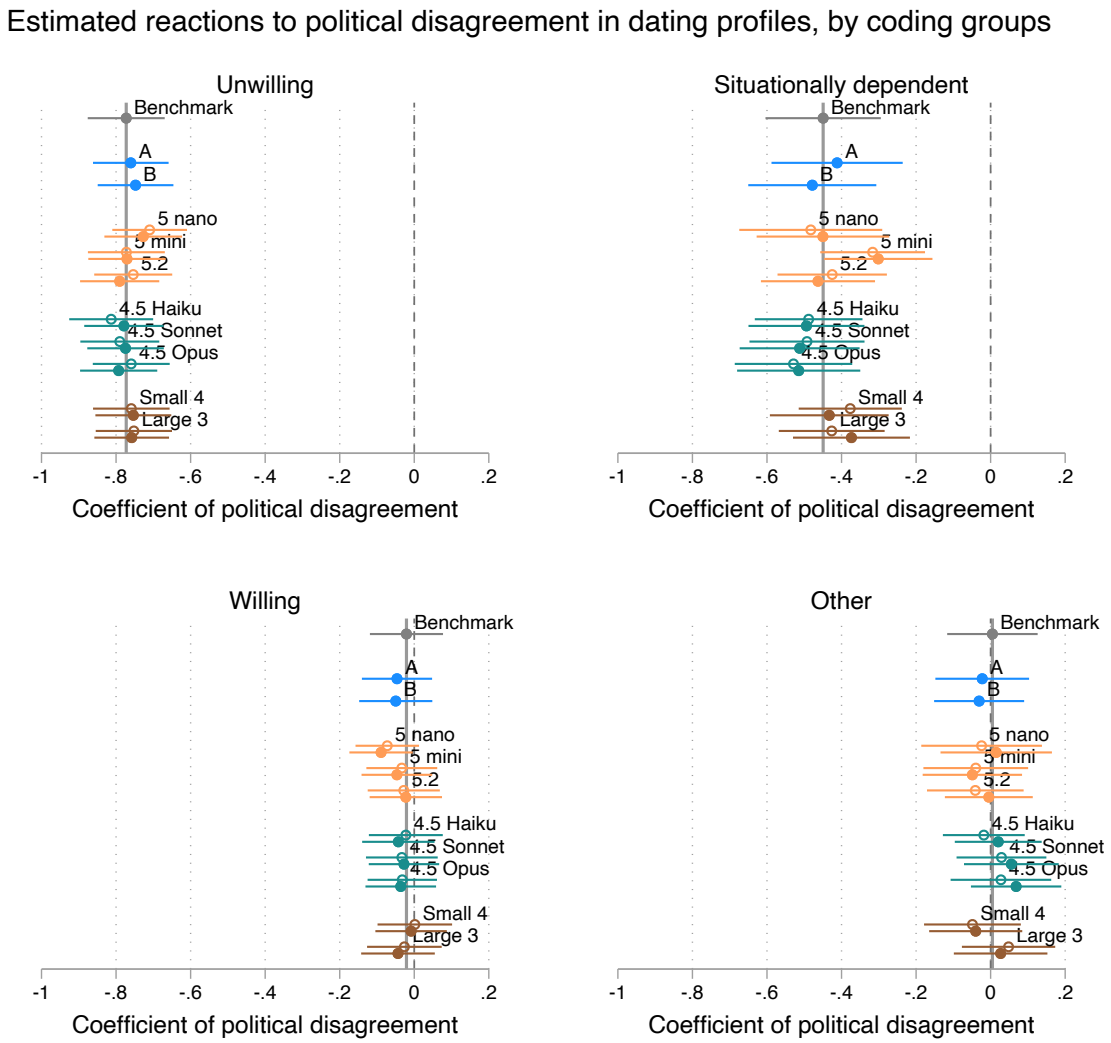

Figure A3: Coefficient estimates from models predicting romantic interest depending on political disagreement between the respondent and the dating profile, using text responses coded by each method and comparing between LLM-coding without training data (hollow circles) and with training data (filled circles).

Coding instructions. All text in regular font was identical for student assistants and LLMs; the text in *italics* was only included / repeated for LLMs.

Act as a conscientious expert coder.

As your coding task, please analyze the following responses about dating across party lines. Provide only the applicable code(s), separated by commas if multiple, and add a numeric value indicating your degree of certainty in the coding task: 0 for no uncertainty, 1 for some uncertainty or 2 quite a lot of uncertainty:

Use these codes:

Group of unwilling-codes: Respondent conveys their desire not to date a member of the aforementioned political party. Their unwillingness may be explicit e.g.: I would not date a Democrat or implicit e.g.: because we wouldn't get along. As a characteristic of this type of response, respondents do not mention any conditions under which they would date a member of the aforementioned party. They only mention reasons as to why they would not like to date them.

UNWILLING_NEGATIVE: Negative perceptions and stereotyping of the other party. These may stand alone, e.g.: I don't like Democrats; Republicans are selfish or may be substantiated by policies that the person is assumed to support e.g.: They are morally corrupt because they support anti-LGBTQ policies; They support policies that harm LGBTQ people; They are bad, dangerous people; They just don't understand.

UNWILLING_DIFFERENT: Differing values/morals/beliefs/interests/lifestyles, e.g.: We would have different values; We are just too different; We like different things so it wouldn't work out; We wouldn't see eye to eye on things; I just disagree with what they believe.

UNWILLING_CONFLICT: Expected conflict. e.g.: Because we would fight all the time; Because we would argue; I don't want to have to debate someone in a relationship. We would have too much disagreement.

UNWILLING_OTHER: Other/unclear/no reason given, e.g.: Because I'm a Democrat.

Group of situationally-dependent codes: Respondent indicates that they would only date the member of the aforementioned party under particular circumstances or they name both positive and negative aspects of dating the person e.g.: Only if we had similar values; It depends on whether they respect my point of view; It could be interesting but may also lead to conflict.

SITUATIONALLY_DEPENDENT_POLITICAL: Depends on degree of political tension, including degree of political involvement; whether the partner can accept and is respectful of opposing viewpoints; and agreement on particular political issues e.g.: It depends on how involved they are politically; I'm ok with disagreement as long as they are pro-life; As long as they aren't extreme aka maga; Extreme political views are a turn off; As long as they don't talk about politics so much.

SITUATIONALLY_DEPENDENT_NON-POLITICAL: Depends on non-political aspects of the person that may be related to politics, including matching values, beliefs, and compatibility e.g.:

Depends on how well we get along; I'm not exactly for it, but if we have common values it could work, as long as they are not racist.

SITUATIONALLY_DEPENDENT_OTHER: Other/unclear/opposing reasons given/what it is dependent on is not specified, e.g.: I'm unsure how the situation would be; Depending on type of encounter (degree of seriousness of relationship), Depends on a lot of things.

Group of willing-codes: Respondent expresses openness to date a member of the other party. Respondents who are willing are often characterized by an openness to dating members of other parties as well as the belief that politics should not play a role in dating e.g.: I don't care which party they vote for; I'm open to it.

WILLING_IRRELEVANT: There is no indication that politics matter, including personally uninterested in politics and viewing politics as irrelevant for dating e.g.: I care more about the personality; Don't care; It does not matter much; Politics does not define my dating; The person is more important than the party.

WILLING_TOLERANT: Acknowledges that there are differences in political beliefs/preferences and accepts them or would be willing to at least try to date despite such differences. Such responses often emphasize tolerance for other viewpoints and perceive selection depending on political party as a form of discrimination or a minor aspect that would not prevent them from dating e.g.: Because I am open-minded; I don't discriminate by party; We can have different beliefs, and it can still work; Yeah, I could get past that; People are people.

WILLING_POSITIVE: Views dating the member of the aforementioned party positively e.g.: I think it would be interesting or inspiring; liberals are hot

WILLING_OTHER: Other/unclear/no reason given e.g.: I've dated a democrat.; wanna meet new people.

Group of other-codes:

OTHER_DONT_KNOW: The respondent indicates that they do not know or are not sure e.g.: Don't know.

OTHER_UNCLEAR: The meaning of the response is unclear e.g.: out of my control, don't like politics; seems to have misunderstood the question e.g.: I don't want to online date; response is in a language other than English; only answer the question for people in general and do not describe their own feelings e.g.: people generally are against it because they don't like the other party.

OTHER_NO_ANSWER: They do not want to respond e.g.: N/A.

OTHER_NONSENSE: The response is nonsense or unrelated to the question e.g.:kjgakjhdfj; This is a great product.

For each response, provide only the applicable code(s), separated by commas if multiple. If you have at least one code that does not end in _OTHER, do not add an _OTHER code, even if it applies i.e.: do

not assign both the codes UNWILLING_NEGATIVE and UNWILLING_OTHER. If necessary, assign multiple codes. BUT: never mix codes from different groups of codes. E.g., NEVER mix dependent-codes with WILLING- or UNWILLING-codes.

Act as a conscientious expert coder. Analyze the following responses about dating across party lines:

[responses listed here]

Provide for each response the applicable code(s), separated by commas if multiple, and add a numeric value indicating your degree of certainty in the coding task. Format your answers as one line per response, in the same order.