Data Annotation with Large Language Models: Lessons from a Large Empirical Evaluation

Eddie Yang[†] Zoey Wang[‡] Carl Zhou[‡] Yaosheng Xu[§]

October 20, 2025

Abstract

Large Language Models (LLMs) are increasingly used in social science research to annotate unstructured data, often replacing research assistants and experts. However, using these predicted annotations in downstream statistical analyses can yield biased estimates – a problem compounded by the black-box and stochastic nature of LLMs. This study evaluates the consequences of LLM annotation for empirical political science research. We conduct a systematic replication and reanalysis of 14 recently published papers from leading political science journals, re-annotating data originally coded by humans or supervised models with 15 different open-weight and proprietary LLMs. Our analysis of over 300 million annotations reveals that LLM annotations have low intercoder reliability with the original annotations and moderate reliability among the LLMs themselves. Smaller models and reasoning models are particularly sensitive to minor variations in artifacts such as prompt format. As a result, downstream estimates derived from different sets of annotations show significant variation, often altering the statistical and substantive conclusions of the original studies. Mitigation strategies, such as in-context learning and bias correction techniques, are useful but have limitations. Based on these findings, we propose best practices for using LLMs for annotation and provide an open-source `R` package, `localLLM`, to support their implementation.

[†]Assistant Professor, Department of Political Science, Purdue University. Email: eddieyang@purdue.edu
[‡]Undergraduate student, College of Communication, University of Amsterdam.
[§]PhD student, Department of Political Science, Purdue University.

1. Introduction

Operationalizing theoretical concepts is a core component of political science research and often requires annotating or classifying unstructured data. Traditionally, this annotation relies on human coders or supervised machine learning models. For example, the widely used conflict and protest dataset ACLED relies on expert coding of textual data from newspapers, reports, and social media (Raleigh et al., 2010). Similarly, the Wesleyan Media Project uses a team of researchers to hand-code American political advertisements for their content and tone (Fowler et al., 2025). These annotation approaches are time-consuming and expensive. Human annotation is slow and difficult to scale, while supervised machine learning requires the creation of a large, hand-labeled training dataset. Furthermore, both methods are subject to researcher influence and manipulation, as they require training either the human coders or the supervised models themselves.

Over the past few years, the development of large language models (LLMs),[1] such as ChatGPT, has offered a new approach to data annotation. Researchers can now simply pass coding rules and unstructured data as prompts to LLMs, which then generate the desired annotations. This approach allows for generating annotations quickly and scaling at a low cost for a wide range of tasks. Additionally, as LLMs can perform "zero-shot" annotation without training, they hold the promise of reducing researcher degree of freedom and minimizing manipulation. Recent studies have also found that LLMs tend to be more accurate than crowdsourced annotations (Gilardi, Alizadeh and Kubli, 2023). For these reasons, LLMs are increasingly used in political science research to generate key variables of interest.[2]

Despite these promises, the rapid development of LLMs also raises many questions for social science research. Given that many annotation tasks are subjective in nature, what

[1] We distinguish LLMs from supervised machine learning models by their ability to annotate without training ("fine-tuning") on the specific annotation task. Our definition differs from some existing studies. In particular, we consider models like BERT to be supervised models rather than LLMs.

[2] See e.g., Mellon et al. (2024); Breuer et al. (2025); Le Mens and Gallego (2025); Lin (2025*a*).

subjective biases do different LLMs encode and how are they different from those of human coders and supervised models? In other words, when an annotation task is given to different LLMs, how much do different LLM annotations agree with each other and with human coders and supervised models? When these annotations are then used in downstream statistical analysis, how much variation in coefficient estimates do we observe as a result of the choice of LLM and model size? As researchers increasingly incorporate LLMs in their research, especially for data annotation, these questions demand careful consideration.

Recent studies have also pointed out several problems with using LLMs for data annotation. For one, measurement errors in the LLM annotations can lead to substantial bias and invalid confidence intervals in downstream statistical analyses (Egami et al., 2024). Additionally, the proliferation of different LLMs generates a hidden researcher degree of freedom as the choice of LLMs can potentially influence the annotations as well as the result of the downstream analyses (Baumann et al., 2025). Even for the same LLM, its annotations can change when queried at different times due to the stochastic nature of LLM generation and the fact that LLM may be updated without notice to the users (Barrie, Palmer and Spirling, 2024). How prevalent and serious these problems are in the context of empirical political science research is a question that remains understudied.

In this paper, we provide a comprehensive evaluation of the impact of using LLM for data annotation in empirical political science research. We use 15 different proprietary and open-weight[3] LLMs to re-annotate datasets from 14 studies published in leading political science journals, which were originally coded by humans or supervised models. We assess annotation quality by measuring intercoder reliability between LLM and original annotations, as well as among the LLMs themselves. We then replicate the original statistical analyses with these new annotations to assess the effect on the published findings. Furthermore, we test how annotation quality is affected by in-context learning (i.e., including examples in the prompt) and by variations in prompt format. Finally, we evaluate the effectiveness of bias-correction

[3]We define an LLM as "open-weight" if its trained parameters (weights) are publicly available for anyone to download and use. In contrast, proprietary models are those for which the weights are not publicly available.

techniques aimed at mitigating systematic measurement error from the LLM annotation process.

Analyzing more than 300 million annotations, we identify four sets of empirical regularities. First, while LLMs have high simple agreement rates with other annotators, they show low intercoder reliability with the original annotations and moderate reliability among the LLMs themselves. This reliability varies significantly depending on the study and the specific LLM. Second, this lack of reliability has downstream consequences: using annotations from different LLMs can lead to different statistical results, sometimes altering a study's substantive conclusions. However, we identify a useful linear relationship: when LLMs agree more with each other, they also tend to agree more with humans and supervised models. This insight allows us to propose a typology of tasks to help researchers determine when LLMs are a suitable choice. Third, in-context learning can improve intercoder reliability, but its benefits quickly plateau. In contrast, prompt formatting has a relatively minor effect on annotation quality and consistency. Fourth, methods for correcting LLM bias are useful but introduce a significant bias-variance trade-off. These corrections can reduce bias but widen the confidence intervals of downstream estimates, and narrowing them requires a large ground-truth dataset. Based on these findings, we offer a set of best practices for researchers using LLMs for annotation.

To our knowledge, this paper presents the first large-scale empirical evaluation of using LLMs for data annotation in political science. We provide a comprehensive comparison of different LLMs across a wide range of data annotation tasks. The exercise contributes to an emerging literature on the evaluation of the use of LLMs for social science research (Ziems et al., 2024; Barrie, Palaiologou and TÃķrnberg, 2024; Bisbee and Spirling, 2025; Timoneda and Vera, 2025). Our work builds on recent assessments of LLMs as data annotators (Barrie, Palmer and Spirling, 2024; Burnham, 2024; Baumann et al., 2025; Halterman and Keith, 2025) and extends this literature in several ways. First, we broaden the coverage of existing evaluations by analyzing a diverse set of proprietary and open-weight LLMs, various political

science annotation tasks, multiple prompt formats, and common concerns and mitigation techniques. Doing so enables us to uncover new insights – for example, on the correlation between human and LLM judgements as well as the most suitable type of LLMs for data annotations. Our hope is that this can provide a set of empirical regularities that future researchers can reference when choosing the best annotation approach. Second, we shift from the conventional perspective of treating human annotations as ground truth to what we think is a more defensible perspective that assumes all annotators are subject to measurement error. This approach guides the majority of our study design and yields results that are robust to concerns about the quality of the original annotations. Finally, we provide a new `R` software package, `localLLM`, to support the implementation of several proposed recommendations in this paper.

2. LLM for data annotation: promises and pitfalls

Large language models (LLMs) are deep neural networks trained on vast amounts of text data[4] to understand and generate human-like text. Prominent models like OpenAI's ChatGPT and Meta's Llama have demonstrated a wide range of capabilities, from sentiment analysis to translation and summarization. LLMs process free-form text as input and can generate either free-form or structured output based on the user's prompt.[5] Here, "structured" means the output is constrained to a limited set of choices (e.g., "positive" and "negative"). This capacity to transform unstructured text into structured data makes LLMs a powerful tool for annotation. For instance, Le Mens and Gallego (2025) used an LLM to code the policy and ideological positions of political texts, while Mellon et al. (2024) applied one to identify the most important issues in open-ended survey responses. In this section, we highlight the key advantages and drawbacks of using LLMs for data annotation, which are summarized in Table 1.

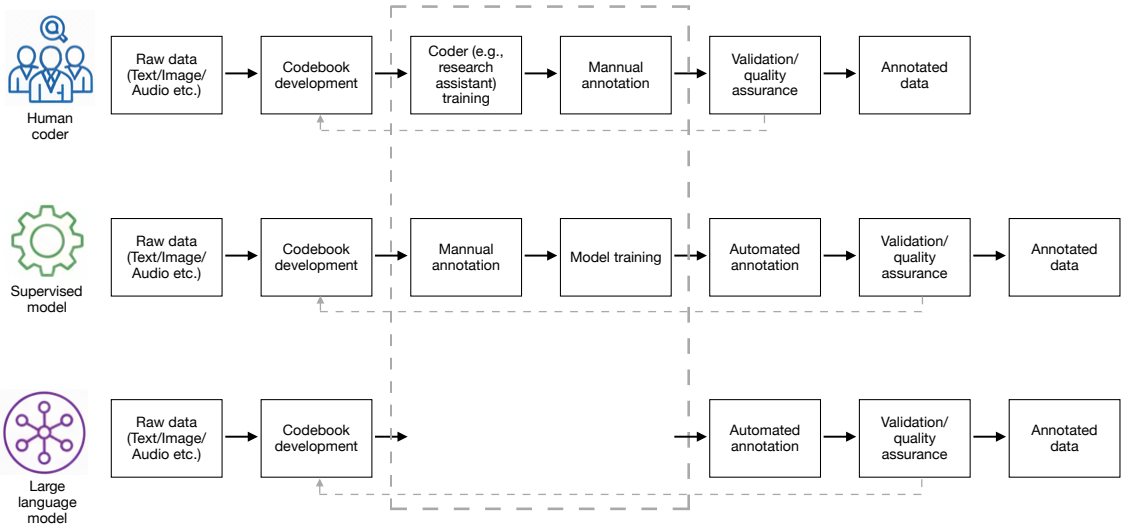
[4]Recent LLMs are trained on tens of trillions of tokens, where a token is the smallest unit (typically a sub-word) that LLMs use to represent text.

[5]In addition to textual data, some LLMs can also take other media, such as audio, image, and video data, as input. In this paper, we focus on LLMs' textual abilities.

Compared to the two prevailing data annotation approaches: annotation by human coders and supervised models, LLMs have several advantages. Perhaps the biggest advantage of LLMs is that there is no need for manual coding. As the annotation processes in Figure 1 show, annotation by human coders requires manual coding of the entire dataset. For supervised models, there is still a need to manually code a subset of the dataset to serve as the training data for the supervised model. On the other hand, annotation by LLMs simply skips this step and automates the annotation of the entire dataset once the codebook is developed. Without the need for manual coding, LLMs can typically annotate data at a much lower cost compared to the other two approaches. This makes them highly scalable, as they can handle very large datasets with minimal additional human effort or cost. For example, the price for OpenAI's GPT-5 is $2.50 per 1 million tokens. In the 14 studies we analyzed in this paper, the median dataset has 62,226 samples and a median token count of 2.02 million tokens. Annotating such a dataset with GPT-4o would only cost about $5. While this only accounts for the cost of input into the LLM, the token count for structured output is generally much smaller and thus cheaper. In contrast, hiring a research assistant or an expert coder to go through thousands of samples would incur a cost that is orders of magnitude higher.

While LLMs do not require traditional, large-scale training data, they can still leverage annotations through a technique called in-context learning or few-shot learning. The learning is "in context" because examples of text-annotation pairs are included directly within the prompt given to the LLM (see e.g., Figure 2). Similar to fine-tuning for supervised models, in-context learning enables an LLM to adapt quickly to a specific annotation task, but it is often more efficient as it requires only a handful of examples. This approach is particularly useful when researchers have high-quality annotations available or wish to guide the model's output by providing specific demonstrations. Studies have shown that LLMs through in-context learning can match or surpass the performance of state-of-the-art fine-tuned models (Brown et al., 2020).

FIGURE 1. DATA ANNOTATION PROCESSES



Notes: Different workflows for data annotation: one performed by human coder, one using a supervised model, and one using a large language model. Each workflow is represented by a horizontal row of boxes and arrows, showing the sequence of steps from raw data to annotated data. The dashed grey box in the center of the diagram encloses steps for manual coding that is required for human coders and supervised models but not for LLMs.

Despite their promise, LLMs also present several notable drawbacks when used for data annotation. First, measurement errors in LLM annotations are more difficult to anticipate, as researchers often lack a clear framework for predicting where such errors will arise or in what direction they will bias results. With human annotators, by contrast, a body of theory and empirical evidence helps identify potential sources and directions of bias. For example, studies show that human coders may rely on partisan cues when coding political texts (Ennser-Jedenastik and Meyer, 2018) or that their demographic background can shape their decisions (Al Kuwatly, Wich and Groh, 2020; Sap et al., 2021). The biases and behaviors of LLMs, however, remain far less understood, in part due to their "black-box" nature and their recent emergence as a technology. This issue is compounded by the fact that the LLM annotation workflow requires less human intervention and input, potentially further increasing the opacity of measurement errors.

A serious consequence of LLM measurement errors is that downstream statistical analyses using the LLM annotations can produce biased results. While measurement errors are

ubiquitous for all annotation approaches, not just for LLMs, the opacity of LLM annotations makes it more challenging to predict, diagnose, and correct for systematic errors in their annotations. For example, without the guidance of existing theories and empirical findings, it is much less efficient to look for specific annotations where measurement errors may occur. In light of this problem, several methods have been proposed to correct the bias resulting from measurement errors (Angelopoulos et al., 2023; Egami et al., 2023, 2024) but their applicability and usefulness have not been systematically tested in political science research.

A related issue is that the proliferation of LLMs allows researchers to choose from a large pool of models. Even when their overall performance is similar, each model may have different biases and measurement errors. This gives rise to a new form of "researcher degree of freedom" where a researcher can cherry-pick an LLM to get their preferred result. This problem is empirically demonstrated by Baumann et al. (2025), who show that, by using different LLMs and prompts, a researcher can arrive at almost every kind of conclusions (null, negative significant, positive significant) with the same data. Similarly, by re-annotating the data in Bor and Petersen (2022) with different LLMs, Barrie, Palmer and Spirling (2024) show that different LLMs can be comparable in overall accuracy but yield very different coefficient estimates. Furthermore, even for LLMs in the same model family and developed by the same company (e.g., GPT-3.5 and GPT-4), there is no guarantee that they will produce annotations that yield similar coefficient estimates (Barrie, Palmer and Spirling, 2024).

In addition, LLMs may also be sensitive to seemingly minor artifacts in the annotation workflow. For example, minor changes to the prompt can result in different LLM annotations (Barrie, Palaiologou and TÃķrnberg, 2024; Baumann et al., 2025). Furthermore, because of the stochastic nature of LLMs, a different seed for the random number generator can potentially yield different annotations. For proprietary models, since the weights are not publicly available, they may be updated by their developers without notice. As a result, querying the same model at different times can generate different annotations (Barrie, Palmer

Table 1. Advantages and Drawbacks of Using LLMs for Data Annotation

Advantages	Drawbacks
• **No Manual Coding:** Automates the annotation process, saving significant time and effort.	• **Opaque Errors:** The black-box nature of LLMs makes it difficult to anticipate or understand the sources and direction of annotation errors and biases.
• **Low Cost & Scalability:** Inexpensive to run on large datasets, making it highly scalable.	• **Biased Downstream Analysis:** Unpredictable measurement errors can bias results in statistical analyses, and these errors are challenging to diagnose and correct.
• **Efficient Adaptation:** Adapts quickly to new tasks with few-shot learning (in-context examples).	• **Researcher Degrees of Freedom:** The proliferation of models allows researchers to potentially "cherry-pick" an LLM that produces their preferred results
• **High Performance:** Can match or surpass the accuracy of fine-tuned supervised models.	• **(Non-)reproducibility & (In-)stability:** Annotations can vary due to minor prompt changes, model updates, or stochasticity.

and Spirling, 2024). However, how prevalent and serious these problems are in political science research requires systematic evaluation.

3. Evaluating LLMs as data annotators

We assess how issues with LLM annotation affect empirical political science research. Our assessment is guided by common questions researchers face when deciding whether and which LLM to use for annotation. Specifically, we ask:

1. How well do LLM annotations align with those from humans/supervised models, and how consistent are they across different LLMs?
2. Given LLMs may generate different annotations, to what extent does the choice of LLM influence downstream coefficient estimates?
3. How sensitive are LLMs to small changes in prompt design?
4. How much do bias-correction methods help reduce concerns with LLM annotation reliability and sensitivity, and what are the trade-offs?

5. Are certain LLMs (e.g., larger or proprietary models) better suited for annotation than others?

First, we establish the extent of annotation disagreement between humans/supervised models and LLMs and among LLMs themselves (Question 1). We then quantify how this disagreement affects the results of downstream statistical analyses (Question 2). Next, we evaluate how researcher decisions in prompt construction (e.g., choice of prompt format or inclusion of annotated examples)influence annotation consistency (Question 3). Given these potential issues, we examine whether bias-correction methods can be used to address them, assuming a sample of ground-truth annotations is available (Question 4). Finally, we consider if model characteristics like model size are correlated with annotation quality and consistency (Question 5).

Our hope is that, in answering these questions, we can provide comprehensive and empirically-grounded evidence for researchers seeking to responsibly leverage LLMs for data annotation.

4. Data and research design

To answer our research questions, we reanalyze 14 recent studies from five political science journals for which some variables of interest are the result of annotations of text data. Our design proceeds in four main stages. First, for each study, we use its original codebook to construct prompts and instruct each of the 15 LLMs to re-annotate the original text data. Second, we evaluate the quality of these annotations by calculating standard inter-coder reliability metrics (e.g., Krippendorff's alpha) to measure the agreement between each LLM's annotations and the original annotations, as well as the agreement among the LLMs themselves. Third, to assess the impact on downstream statistical inferences, we substitute the original annotated variable(s) in the authors' replication code with the newly generated variables and re-estimate the original models. This allows us to quantify the variation in coefficient estimates, standard errors, and substantive conclusions that arises from the choice

of annotator. Finally, we conduct a series of tests to explore the effects of in-context learning, prompt format variations, and the effectiveness of bias-correction methods. Below we detail our criteria for selecting the studies and LLMs, as well as the annotation and analysis procedure used.

4.1. Study selection

We focus on studies published between 2018 and 2025 in five political science journals: *APSR*, *AJPS*, *JOP*, *BJPS*, and *PSRM*. We consider studies that meet the following three criteria: 1) they involve annotations of text data by either humans or supervised models (e.g., random forest, BERT); 2) the annotations are discrete (categorical) rather than continuous; and 3) the annotations are used in downstream statistical inferences (e.g., a regression). We exclude studies with continuous annotations (e.g., probabilities) because LLMs tend to have poor confidence calibration (Guo et al., 2017). Our selection thus represents a "less-problematic" setting for the use of LLMs. Our search yielded approximately 35 studies that meet these criteria. Of those, 9 studies included both the text and annotations in their public replication data and had results we could successfully replicate. After contacting authors directly, we obtained the necessary data for an additional 5 studies. Our reanalysis is thus based on 14 studies, which are summarized in Table 2. In total, the studies include more than three million annotations and span a variety of textual data sources, ranging from elite communication, such as judicial opinions and legislative debates, to user-generated content like social media posts and corporate financial transcripts.

4.2. LLM selection

We base our LLM selection on both existing studies in political science that have used LLMs as well as our knowledge about the field of large language models. We survey recent studies that used LLMs as well as Hugging Face, the largest LLM repository, for commonly used LLMs. In total, we select 15 different LLMs with variations in model size, type,

Table 2. Summary of Studies

Study (year)	Journal	Variable description	Variable type	Original annotation method	Sample size
Choi, Harris and Shen-Bayh (2022)	APSR	Judicial decisions	Binary	Human	9,545
Fowler et al. (2021)	APSR	Type of political tv ad	Categorical	Human + Supervised model	14,452
Gohdes (2020)	AJPS	Type of killing	Categorical	Human + Supervised model	65,274
Gohdes and Steinert-Threlkeld (2025)	AJPS	Tweet sentiment	Categorical	Human + Supervised model	34,412
Hulme (2025)	APSR	Stance on military intervention	Binary	Human	27,811
Hunter (2025)	JOP	Type of responsibility attribution	Categorical	Human	5,943
Li (2023)	BJPS	Analyst sentiment	Binary	Supervised model	578,411
Lin (2025*b*)	JOP	Political nature of firm Q&As	Binary	Human + Supervised model	418,480
Milliff (2024)	APSR	Level of control & predictability	Categorical	Human + Supervised model	3,115
Müller and Fujimura (2025)	PSRM	Policy domain	Categorical	Human + Supervised model	59,619
Müller and Proksch (2024)	BJPS	Nostalgic rhetoric	Binary	Human + Supervised model	1,192,675
Pan and Chen (2018)	APSR	Government wrongdoing	Categorical	Human	1,412
Rozenas and Stukal (2019)	JOP	Event responsibility attribution	Categorical	Human	4,317
Widmann (2025)	JOP	Emotional appeal	Categorical	Supervised model	627,102

Notes: A more detailed description of each article's annotation procedure is included in Section D of the Supplementary Materials (SM). The sample size is based on the number of samples in each article's replication package.

developer, and reasoning capability, with a preference for popular and more recent models developed by well-known companies. Table 3 provides a summary of the selected LLMs. The selected LLMs include models like gpt-4o mini, llama 8b, and llama 70b that have been often used in existing studies, as well as newer models such as gpt-5, gpt-oss 120b, and gemma-3 27b. The models also show a wide variety of sizes, ranging from 4 billion parameters to 120 billion parameters.

Another dimension in which LLMs differ is their "reasoning" capability. Reasoning models are more recent LLMs that were trained through reinforcement learning to reason before completing a task. In contrast to non-reasoning LLMs that directly generate the final output, reasoning LLMs will often "think" by generating a "chain of thought" - sequence of text that attempts to break down a problem into steps and process each one logically in a "thinking out loud" fashion – before arriving at the final output. Because of their ability to process tasks incrementally, reasoning models often achieve better performance for more complex tasks like solving Olympiad-level math problems and passing professional exams.[6] On the

[6]For example, reasoning models from Google and OpenAI achieved gold medal-level performance at the 2025 International Mathematical Olympiad (IMO). See https://www.axios.com/2025/07/21/openai-deepmind-math-olympiad-ai.

Table 3. Summary of LLMs

Model name	Size	Type	Reasoning	Developer	Release date
Qwen3-4B-Instruct-2507	4 billion	Open-weight	No	Alibaba (China)	August, 2025
Apertus-8B-Instruct-2509	8 billion	Open-weight	No	Swiss AI (Switzerland)	September, 2025
Llama-3.1-8B-Instruct	8 billion	Open-weight	No	Meta (U.S.)	July, 2024
DeepSeek-R1-0528-Qwen3-8B	8 billion	Open-weight	Yes	DeepSeek (China)	May, 2025
gemma-3-12b-it	12 billion	Open-weight	No	Google (U.S.)	March, 2025
gpt-oss-20b	20 billion	Open-weight	Yes	OpenAI (U.S.)	August, 2025
Mistral-Small-3.1-24B-Instruct-2503	24 billion	Open-weight	No	Mistral AI (France)	March, 2025
gemma-3-27b-it	27 billion	Open-weight	No	Google (U.S.)	March, 2025
Qwen3-32B	32 billion	Open-weight	Yes	Alibaba (China)	April, 2025
Llama-3.3-70B-Instruct	70 billion	Open-weight	No	Meta (U.S.)	November, 2024
Qwen2.5-72B-Instruct	72 billion	Open-weight	No	Alibaba (China)	September, 2024
gpt-oss-120b	120 billion	Open-weight	Yes	OpenAI (U.S.)	August, 2025
GPT-4o mini	-	Proprietary	No	OpenAI (U.S.)	July, 2024
GPT-4.1 mini	-	Proprietary	No	OpenAI (U.S.)	April, 2025
GPT-5	-	Proprietary	Yes	OpenAI (U.S.)	August, 2025

Notes: Model size for proprietary LLMs is not included because there is no publicly available information.

other hand, the chain of thought results in a much longer output token count, making the reasoning models more expensive to run.

4.3. Annotation procedure

We use the 15 selected LLMs to re-annotate text data from the 14 studies. Using LLMs as annotators requires carefully constructed prompts. For consistency, we adopt a standardized prompt design, as shown in Figure 2, across all annotations. Each prompt consists of four sections: annotation task, coding rules, target text, and output format. For tests involving in-context learning, we additionally include an "Examples" section. Because prompts are central to the LLM annotation workflow, we take great care in constructing the prompt template for each study. When studies provide detailed codebooks, we adapt them to fit the prompt structure while preserving their substance as closely as possible. When codebooks are unavailable, we infer annotation tasks and coding rules through close reading of the studies. Each prompt template is pilot tested on a small sample of target texts to ensure that LLMs demonstrate correct understanding of the annotation task and are able to generate valid

labels.

FIGURE 2. PROMPT FORMAT

```
## Annotation Task
Your task is to classify the target text, a sentence from a political party manifesto, based on whether it contains
nostalgic rhetoric.

## Coding Rules
Nostalgia is defined as a predominantly positive emotion associated with recalling important or momentous past
events, usually experienced collectively with close others or as part of a national identity. Return 1 if the text is
nostalgic, and 0 if it is not.

* Not Nostalgic (Code 0): The text does not contain a positive, emotional reference to the past. This includes
statements that are future-oriented, purely factual descriptions of the past, or negative critiques of past events or
governments.
* Nostalgic (Code 1): The text positively and emotionally references a collective past. The text should express a
longing for or a proud recollection of a nation's history, heritage, traditions, or a bygone era.

## Examples
Better protection of data transferred to the US warden
Answer: 0

We build the Estonian National Museum and the Tallinn Music High School, new buildings.
Answer: 1

## Target Text
{{target-text}}

## Output Format
{"answer": "Your choice here"}

Remember to replace the placeholder text in the "answer" field with your actual annotation. Your annotation must
be one of the numerical codes (1 or 0). Respond only with a valid JSON object and nothing else.
```

Notes: The figure illustrates an example of the prompt used in the annotation procedure. The prompt is organized into sections, each introduced by a header beginning with "##". It includes the task definition, coding rules, illustrative examples, the target text placeholder, and the required output format. The prompt is also an example of "2-shot learning", where two annotated examples are provided as demonstrations before the target text.

To examine the effect of in-context learning, we repeat the annotation process with prompts that include the "Examples" section. Specifically, we test 2-shot, 5-shot, and 10-shot learning, meaning that the prompt contains two, five, or ten annotated examples, respectively. These examples are sampled randomly from the original annotations. For comparison, we refer to annotations made without examples as "0-shot" in the following sections.

To assess the effect of minor changes in prompt design, we also conduct the annotation process using an alternative format in which markdown symbols (e.g., ##, *) are replaced

with XML tags (<> and </>). An example of this alternative design is provided in Section A of the Supplementary Materials (SM). We further test in-context learning under the alternative format and compare results across the two designs.

Each run over all studies using one LLM requires roughly 3.04 million annotations. For each prompt design, we conduct four runs (0-, 2-, 5-, 10-shot). Therefore, iterating over all 14 studies, 15 LLMs,[7] four in-context learning designs, and two prompt designs, our total annotation count is roughly 300 million. We use Nvidia GPUs and a performant LLM inference engine, `vllm`,[8] for annotation. Importantly, the `vllm` engine supports reproducible annotation, meaning each annotation can be exactly reproduced given the same input, software, and hardware. We verify this is indeed the case and adopt this option for all our annotations. Additional details about the annotation procedure and its implementation, including the procedure for reproducible annotation, are provided in SM B.

4.4. Reanalysis and additional procedures

Given the original and LLM annotations, we design a series of evaluations to study the viability and implications of using LLMs as a data annotator.

First, we assess the LLMs' instruction-following capability, arguably the most basic requirement for LLMs to be reliable annotators. We define instruction-following as the ability to produce valid annotations in the format specified in the prompt. In our case, each study's set of annotation choices is defined both in the "Coding Rules" section and by the final instruction. The expected output format (JSON) is also defined in the prompt. An LLM's failure to produce valid annotations in the specified format renders its output unusable for subsequent analysis. Therefore, before analyzing the content of the annotations, we first evaluate the rate at which each model produces structurally and semantically valid outputs, establishing a baseline for its viability as a data annotator.

Next, to answer our first research question, we examine the agreement between the orig-

[7] Given the cost of querying proprietary LLMs, we only use them for 0-shot annotations.
[8] https://docs.vllm.ai/.

inal and LLM annotations, as well as among LLMs. Here, we deviate from several existing studies (e.g., Egami et al. 2023, 2024; Baumann et al. 2025) and *do not* adopt the perspective that annotations by humans or supervised models are the ground truths. We instead treat all annotators as entities with their distinct subjective biases and all subject to measurement errors. Accordingly, we use intercoder reliability to quantify agreement across annotators. A key advantage of intercoder reliability over other metrics such as accuracy or simple agreement rate is that it accounts for categorical imbalance within the annotation dataset. Specifically, we use two measures – Krippendorff's alpha and Cohen's kappa – to quantify intercoder reliability among different annotators.

To evaluate the downstream consequences of annotator disagreements (second research question), we replicate the studies' analyses using the LLM annotations. We focus on analyses for which the annotated variables are either the main independent or dependent variables. Because each study may have multiple annotated variables as well as model specifications, there may be more than one estimate that we replicate for a given study. In total, we replicate 63 coefficient estimates from the 14 studies. For each study, we first replicate the reported estimates using the original annotations and document any deviations in SM C. Overall, we are able to exactly replicate most estimates and all deviations are minimal and do not change the original conclusions. We then repeat the analyses with LLM annotations. We compare the LLM estimates with the original estimates to document the extent to which the choice of LLM affects the conclusions drawn. We also quantify the variation in coefficient estimates among the LLMs.

To answer our third research question, we examine the effects of in-context learning and changes in prompt format on annotation consistency and downstream analysis. For in-context learning, we compare 2-shot, 5-shot, and 10-shot annotation results to 0-shot annotations as the baseline. Our analysis first examines whether providing a few annotated examples increases intercoder reliability. We hypothesize that as the number of shots increases, the agreement between LLM annotations and the original annotations (as measured

by Krippendorff's alpha) will improve, and the agreement among the LLMs themselves will also increase. Second, we assess the impact on downstream statistical results by measuring the variation in the 63 replicated coefficient estimates across the 12 open-weight LLMs for each few-shot setting. The goal is to determine if in-context learning can reduce the sensitivity of research conclusions to the choice of a specific LLM. This design allows us to investigate not only whether few-shot learning helps, but also whether there are diminishing returns as the number of examples increases (e.g., from 5 to 10 shots). We also explore whether the benefits of in-context learning are uniform across all models or if it disproportionately helps smaller or less capable models align with the annotation task. Because of the high cost of running millions of annotations with proprietary LLMs, we only use them for 0-shot annotations and restrict our in-context learning analysis to open-weight models.

We also analyze the impact of changes in prompt formatting. Specifically, we compare the annotations and downstream estimates generated using our primary markdown-based prompt format with those from an alternative format that uses XML tags. This comparison is conducted across all 12 open-weight LLMs and all in-context learning settings (0, 2, 5, and 10 shots). The goal is to determine whether seemingly superficial changes in prompt structure - which researchers might make arbitrarily - can introduce systematic variation, thereby affecting the stability and replicability of LLM-based annotation.

In a final set of analyses, we address our fourth research question by evaluating the effectiveness of bias-correction methods. We test two recently proposed methods – design-based supervised learning (DSL) (Egami et al., 2024) and prediction-error robust inference (PRISA). Both methods adjust the coefficient estimates post-hoc using a sample of ground-truth annotations that are assumed to be available. By necessity, we shift our perspective here and treat the original annotations as the ground truths. For each study, we randomly sample a set of original annotations of a given size and use them as the ground truth dataset. We then compare the bias-corrected LLM estimates with the naive LLM estimates. Specifically, we are interested in three quantities: the amount of bias reduction, the trade-off

between bias and variance, and how these two quantities are a function of the size of the ground truth dataset.

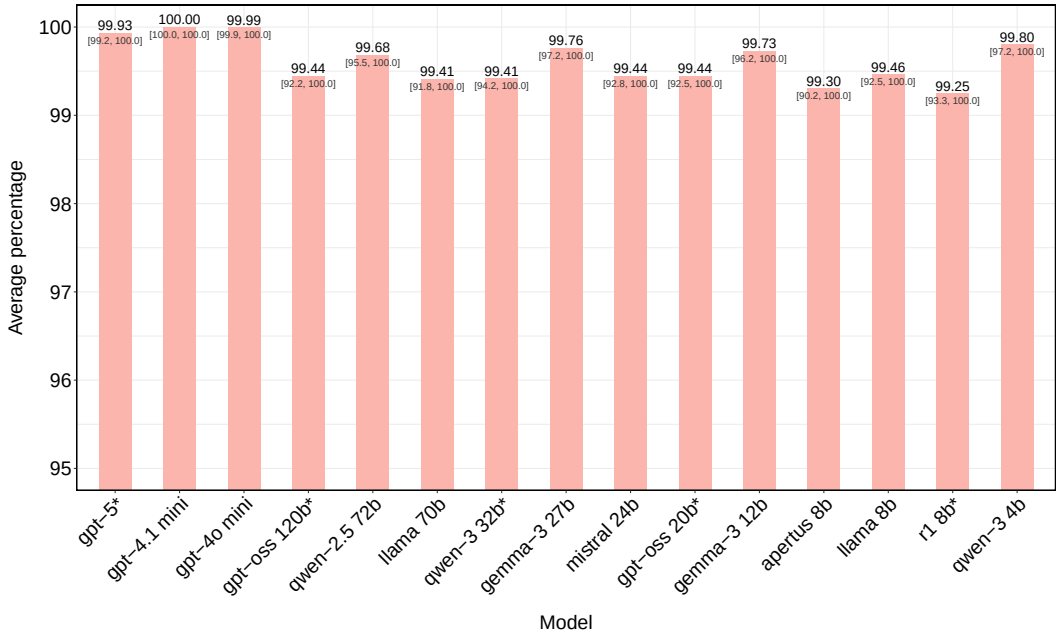
5. Results

We perform the annotation and reanalysis procedures described above for all 14 studies in our sample. This section offers a summary of our findings, with complete results for each study available in the SM.

5.1. LLMs have good instruction following capability

We first evaluate the ability of LLMs to produce valid annotations in the format specified in the prompt. Figure 3 presents the results for 0-shot annotations, showing the average, minimum, and maximum percentage of valid labels for each LLM across the 14 studies.

FIGURE 3. AVERAGE PERCENTAGE OF VALID LABELS BY MODEL (0-SHOT)



Notes: The figure presents the average percentage of valid labels by LLM for 0-shot annotations. The number on top of each bar indicates the average percentage for the corresponding LLM and the numbers in brackets indicate the minimum and maximum percentages across the 14 studies. Reasoning models are suffixed with an asterisk (*).

As Figure 3 shows, LLMs demonstrate good instruction-following capability, with average

validity rates above 99%. Proprietary models show consistently high validity percentages, with gpt-4.1 mini achieving a perfect 100% across all studies. While open-weight models also achieve high average scores, they exhibit less consistency. For example, the minimum scores for llama 70b and apertus 8b drop to 91.8% and 90.2% on certain studies, showing a wider performance variance compared to their proprietary counterparts. However, we observe no notable difference in performance between reasoning and non-reasoning models. If anything, reasoning models demonstrate slightly weaker instruction-following ability. Results for 2-, 5-, and 10-shot annotations are included in SM E, with similar findings.

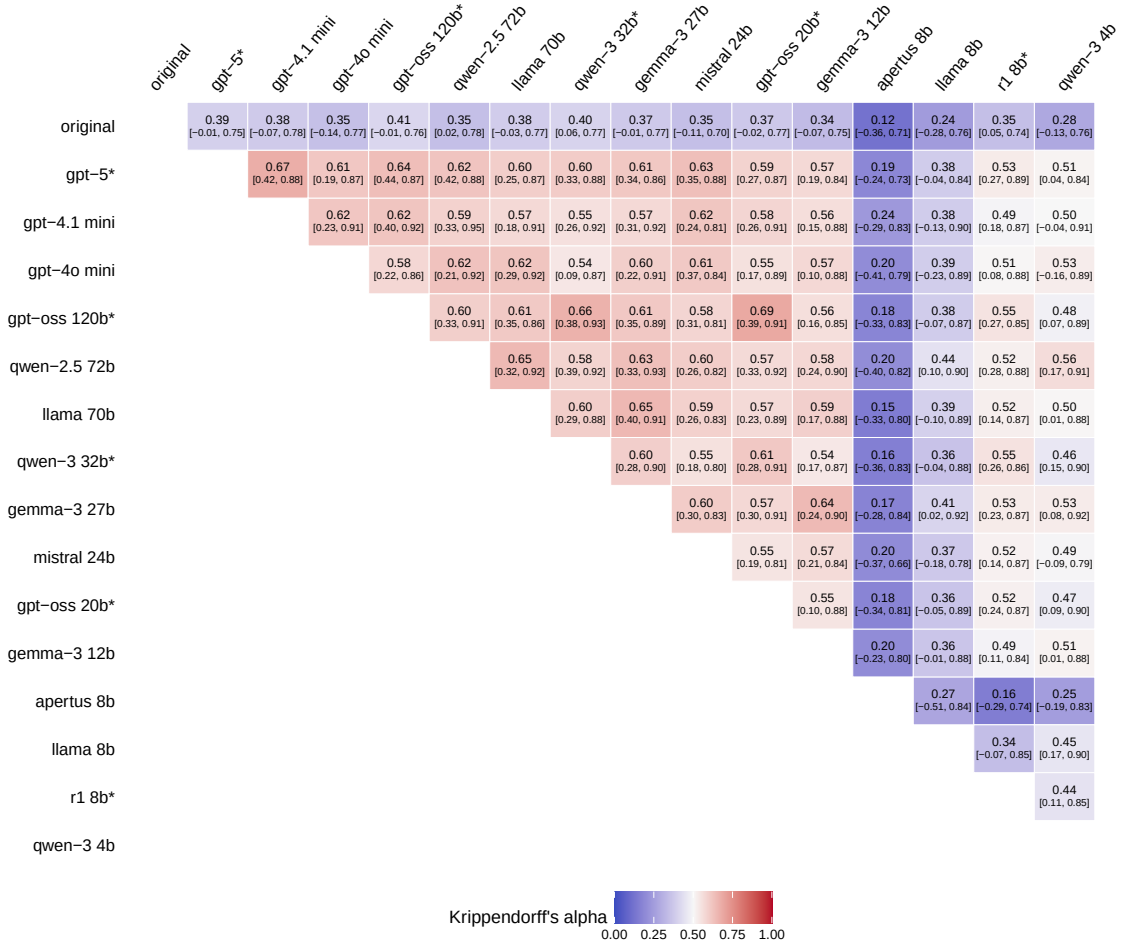
5.2. LLMs produce different annotations

We next assess annotation agreement among LLMs, human coders, and supervised models. We use pairwise intercoder reliability to quantify the level of agreement between any pair of annotators. We report results using Krippendorf's alpha as a measure of intercoder reliability in the main text and include results using Cohen's kappa in SM G.

Our analysis reveals a clear divergence between annotations generated by LLMs and those from humans or supervised models. As shown in the heatmap of pairwise intercoder reliability (Figure 4), the agreement between LLMs and the original annotators is low, with average Krippendorff's alpha scores ranging from 0.12 to 0.41 across the 15 LLMs. These values fall below Krippendorff's (2018) recommendation that studies should "rely only on variables with reliabilities $\alpha \geq 0.8$" and "consider variables with reliabilities between $\alpha = 0.667$ and $\alpha = 0.8$ only for drawing tentative conclusions." As a benchmark from published work, we found 20 studies in the past decade that reported at least one Krippendorff's alpha in the same five political science journals, and the average Krippendorff's alpha is 0.73. We emphasize that the low intercoder reliability does not imply that one annotator is objectively better than the other. Rather, it indicates that LLMs and the original annotators disagree on coding decisions beyond what would be expected by chance. Adjudicating which annotator is more suitable likely requires systematic validation, a point we return to in the discussion

section.

FIGURE 4. HEATMAP OF PAIRWISE INTERCODER RELIABILITY (0-SHOT)



Notes: The figure presents Krippendorff's alphas for all pairs of annotators, averaged across the 14 studies. The numbers in brackets indicate the minimum and maximum alphas for each pair. "Original" indicates the original annotator(s) of the 14 studies. Reasoning models are suffixed with an asterisk (*).

Perhaps surprisingly, we also find only moderate agreement among the LLMs themselves, with pairwise alphas ranging from 0.16 to 0.69. This is notable given that all models received identical prompts, suggesting that different LLMs may have distinct subjective biases in social science annotation tasks. We also observe that agreement is correlated with model size: larger models (>12b parameters) tend to agree with one another ($\alpha > 0.5$), while smaller models (<8b parameters) show lower reliability with all other annotators (including larger models, human annotators, and other small models). Among the largest models (>70b parameters), however, we find little difference between proprietary and open-weight models,

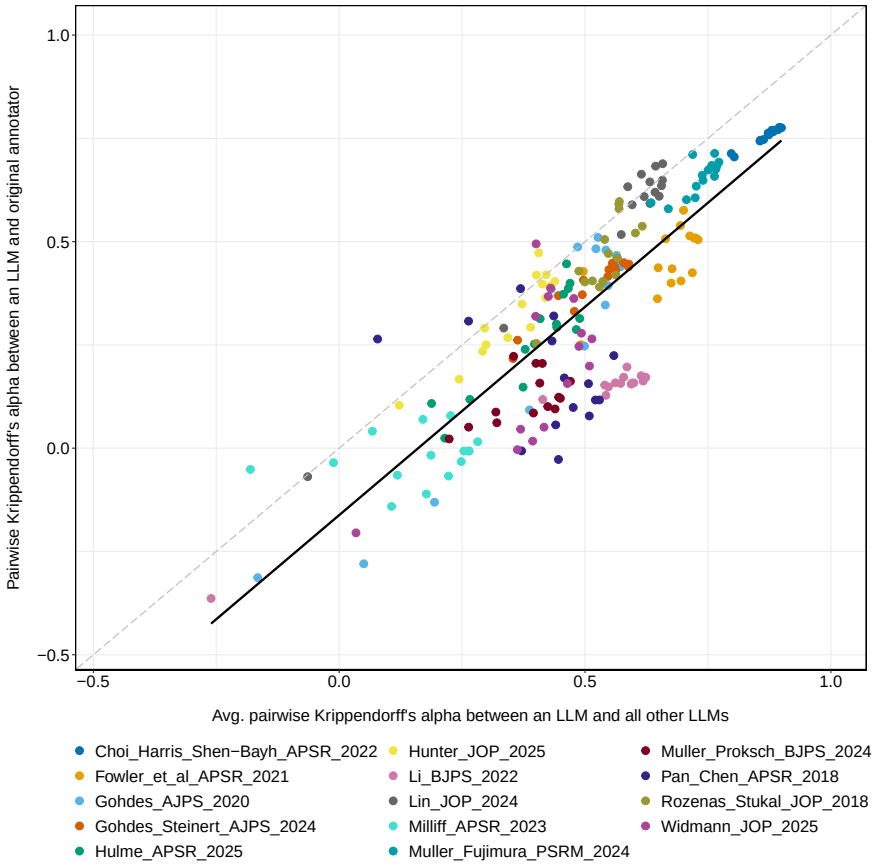
or between those with reasoning capability and those without.

While these reliability scores are low, the simple agreement rate (proportion of agreement over all annotations; SM F) is often high. For instance, larger models agree with the original annotators 71–76% of the time and 81–88% among themselves. This discrepancy between the low intercoder reliability and the high simple agreement rates is explained by the imbalanced categories common in these datasets (see SM F). Krippendorff's alpha, by accounting for chance agreement, corrects for this and reveals the underlying systematic disagreement, which, as we show in the next section, has important consequences for downstream estimates.

The variation in intercoder reliability across studies offers further insight. Figure 5 presents a scatter plot of Krippendorff's alphas, comparing LLM-LLM agreement with LLM-original annotator agreement. The figure reveals two notable findings. First, a large majority of the points fall below the 45-degree line. This shows that for any given study, an LLM's annotations are, on average, more similar to those of other LLMs than to the annotations from the original human coder or supervised model. This result reinforces the conclusion that LLMs as a group produce annotations that are distinct from those of human coders and supervised models. Second, the figure shows a strong positive correlation between LLM-LLM agreement and LLM-original annotator agreement. This implies that there may be some underlying structure to social science annotation tasks, where tasks that elicit high agreement among LLMs also tend to elicit high agreement between LLMs and the original annotators.

Through a qualitative review of the 14 annotation tasks, we find that high-agreement tasks often involve identifying concepts grounded in explicit textual evidence. This means that the target text often includes words or phrases that can be used as strong evidence for an annotation decision. For example, in Choi, Harris and Shen-Bayh (2022), which has some of the highest alpha values in Figure 5, the task is to determine the outcome of judicial appeals. These outcomes are often stated explicitly in the written decisions, making them easy to identify. In contrast, tasks with lower agreement typically involve more complex

Figure 5. Scatter plot of LLM-LLM and LLM-original intercoder reliability



Notes: The figure presents a scatter plot of LLM-LLM and LLM-original annotator intercoder reliability. "LLM-LLM" is defined as the intercoder reliability between an LLM and all other LLMs for a given study. Each dot represents an LLM-study pair. Dots on the 45-degree dotted line indicate equal LLM-LLM and LLM-original annotator intercoder reliability. Dots below the 45-degree line indicate that the LLM-original annotator intercoder reliability is lower than that for LLM-LLM. A best-fit line (black) is plotted to facilitate interpretation.

or ambiguous concepts that lack clear textual signals and may require external contextual knowledge. For instance, in Milliff (2024), the task is to identify the speaker's appraisal of control and predictability based on a sentence extracted from oral histories of Indian Sikhs. This task likely generates disagreement because it lacks clear signals in the text and requires LLMs to make inferences based on their internal knowledge of the historical context.

The linear relationship between LLM-LLM agreement and LLM-original annotator agreement also implies that we may be able to predict the difficulty and agreement level of an annotation task before all annotations are completed. We implement this in the `localLLM`

R package, where a user can use multiple LLMs to annotate a sample of the texts to assess the level of agreement.

5.3. Different annotators yield highly variable estimates

Given that different annotators produce substantially different annotations, we next assess the impact of this variability on downstream statistical analyses. We compare the original coefficient estimates from the 14 studies with estimates derived from LLM annotations. We are also interested in how variable the downstream coefficient estimates can be as a result of the choice of LLM.

Table 4 summarizes this comparison, showing the percentage of LLM-derived estimates that align with the original estimates in terms of both sign and statistical significance. Statistical significance is calculated at the 0.05 level. For now, we focus on the result for the main prompt format with 0-shot learning (first row of Table 4).

Table 4. Comparison of LLM and original estimates

Prompt format	**In-context learning**	**Same Sign**		**Different Sign**	
		Same Sig. (%)	**Diff. Sig. (%)**	**Same Sig. (%)**	**Diff. Sig. (%)**
Main	0-shot	62.6	17.3	12.2	8.00
	2-shot	60.2	19.4	13.8	6.61
	5-shot	64.9	19.6	10.6	4.89
	10-shot	63.2	20.8	9.39	6.61
Alternative	0-shot	62.0	20.2	11.7	6.11
	2-shot	60.2	21.6	11.9	6.35
	5-shot	63.4	18.3	11.4	7.01
	10-shot	63.6	20.1	10.2	6.08

Notes: The table shows the percentage agreement in sign and statistical significance between original and LLM-derived estimates, broken down by prompt format and in-context learning setting. "Same Sign" indicates that both estimates are positive or both are negative. "Same Sig. (%)" indicates that both estimates have the same significance status (i.e., both are significant or both are not), while "Diff" indicates a mismatch.

We find some degree of congruence: in 62.6% of cases, the LLM estimates match the

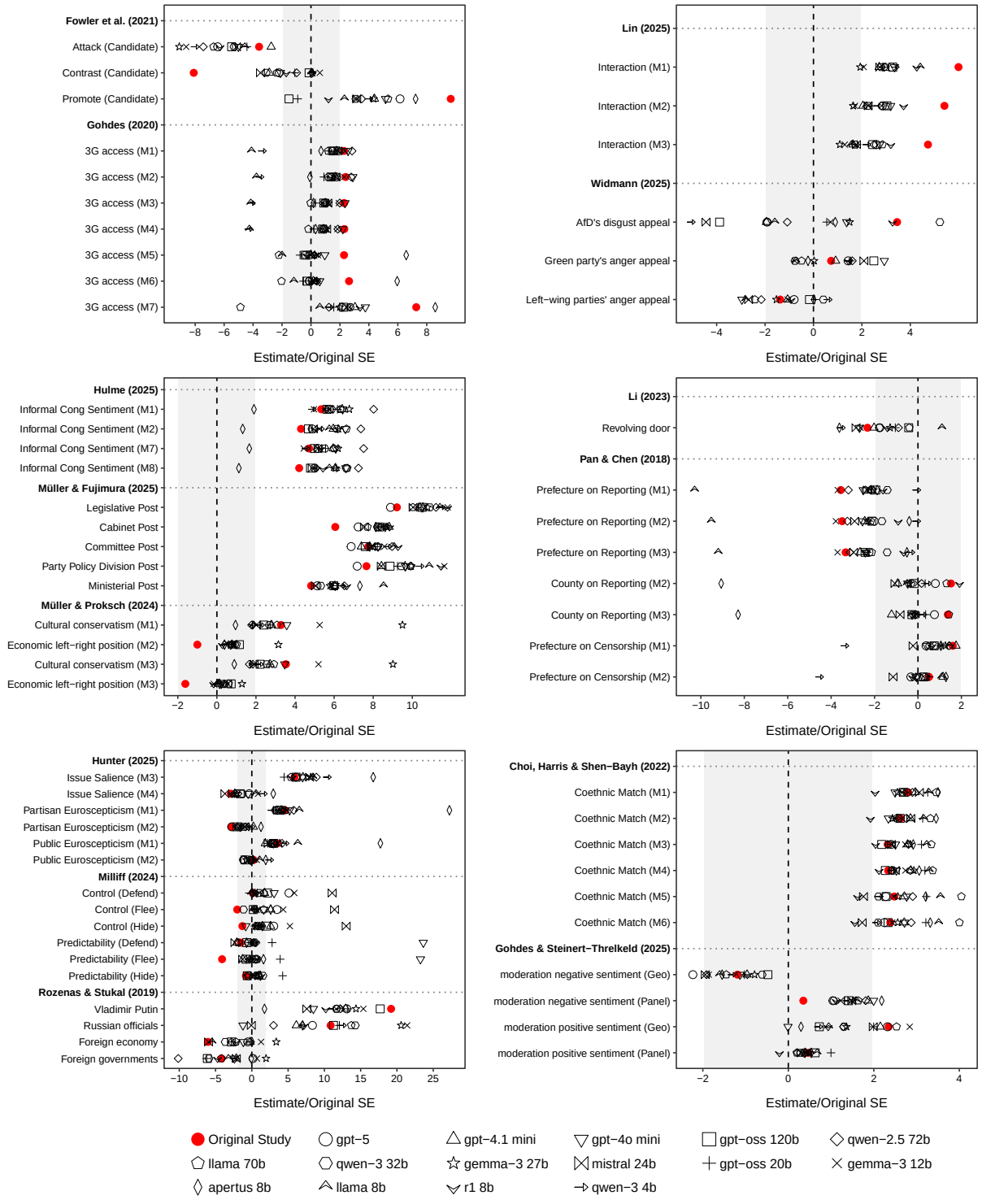
original estimates in both sign and statistical significance. Ignoring the magnitude of the estimates for now, this means that we would reach the same conclusion whether we use an LLM or the original annotator. However, there are also substantial discrepancies. 25.3% of LLM estimates (17.3% + 8.00%) yield a different statistical conclusion than the original estimates. More concerningly, about 20.2% of LLM estimates (12.2% + 8.00%) point in the opposite direction as the original coefficients.

While Table 4 summarizes agreement in sign and significance, it provides limited information on the magnitude of the estimates. To investigate this, Figure 6 visualizes the coefficients for each study, normalized by the original standard errors. As Figure 6 shows, in nearly every case, the estimates show a wide spread, with the difference sometimes reaching an order of magnitude of the original standard error. For example, in the "AfD's disgust appeal" analysis from Widmann (2025), estimates range from strongly negative and significant (-5.55 of the original SE) to strongly positive and significant (5.22 of the original SE), illustrating that the choice of LLM can lead to diametrically different conclusions. In SM I, we document that estimate variability is negatively correlated with intercoder reliability. Furthermore, while some models (e.g., apertus 8b) appear more prone to generating outliers, no single LLM consistently replicates the original estimates. In fact, there seems to be no discernible pattern suggesting that any particular model will systematically produce larger or smaller coefficients than the original.

5.4. In-context learning helps, but not by a lot

To mitigate the high variance in annotation and downstream statistical inference, a common strategy is in-context learning. It works by conditioning the LLM generation on annotated examples that are included in the prompt. To assess its effectiveness, we compare the change in intercoder reliability, aggregated across studies and LLMs, between different settings of in-context learning (0-, 2-, 5-, and 10-shot). We also report the congruence in sign and statistical significance in Table 4.

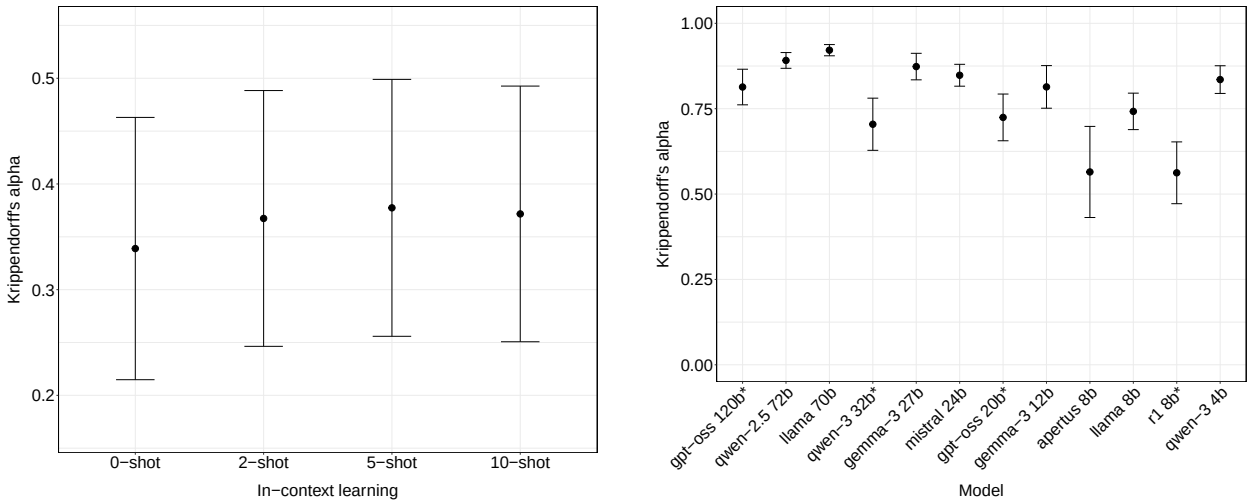
Figure 6. Original vs. LLM-derived estimates by study



Notes: The figure presents the distribution of estimates across studies. The original estimates are highlighted in red. The shaded regions indicate +1.96 and -1.96 estimate/original se ratios. Note that the range of the x-axis is different for each plot.

Panel (a) of Figure 7 plots the changes in intercoder reliability. We observe a modest increase in the mean Krippendorff's alpha as the number of in-context examples goes from 0-shot ($\alpha = 0.34$) to 5-shot ($\alpha = 0.38$). However, the performance slightly dips at the 10-shot setting. Importantly, the confidence intervals are wide and overlap substantially across all four conditions, suggesting that the observed improvements in inter-coder reliability are not statistically significant. In SM H, we plot the changes in intercoder reliability by study and find that long texts (e.g., Choi, Harris and Shen-Bayh (2022) has a median token count of 1450[9]) tend to have decreasing mean Krippendorff's alpha, likely because LLMs struggle with multiple long texts in the same prompt. The inclusion of annotated examples also comes at a cost as it increases the input token count. This increases monetary cost for proprietary models and slows computation for open-weight models. SM B includes a more detailed cost and speed comparison.

FIGURE 7. EFFECTS OF IN-CONTEXT LEARNING AND PROMPT FORMAT ON INTERCODER RELIABILITY



(A) AVG. INTERCODER RELIABILITY BY IN-CONTEXT LEARNING

(B) INTERCODER RELIABILITY BETWEEN MAIN AND ALTERNATIVE PROMPT

Notes: Panel (a) shows the mean Krippendorff's alpha for 0-, 2-, 5-, and 10-shot learning. Panel (b) shows the intercoder reliability between the main and alternative prompts for a given LLM. Cluster-bootstrapped standard errors are used in both panels.

Table 4 shows that the moderate increase in annotation agreement has a marginal impact

[9]Using gpt-4o's tokenizer. Each LLM's tokenizer is slightly different and may give different result.

on the extent of sign and statistical significance congruence between the original and the LLM estimates. Across both prompt formats, the percentage of estimates that match the original in both sign and statistical significance (the "Same Sign, Same Sig." column) hovers in a narrow range. For the main prompt, this figure only increases from 62.6% in the 0-shot setting to a peak of 64.9% in the 5-shot setting. Moreover, there is no monotonic relationship between the number of in-context examples and congruence; for instance, the 2-shot setting performs slightly worse than the 0-shot setting for both prompt formats. This lack of substantial improvement, combined with the increased costs and computational overhead discussed earlier, suggests that while in-context learning can offer a small benefit under specific conditions, it is not a silver bullet for improving the reliability of LLM-based annotations.

5.5. Marginal effect of prompt format

We also investigate how sensitive LLMs are to minor changes in the annotation workflow. Specifically, we assess the effect of small variations in prompt format, comparing annotations based on our main prompt (Figure 2) and an alternative format (SM A).

The results are presented in Panel (b) of Figure 7. For each LLM, we calculate the intercoder reliability between the annotations produced by the two different prompts. The panel shows that changes in prompt format have only a marginal effect on annotation agreement, as most intercoder reliability scores are high (> 0.75). Interestingly, smaller models and reasoning models are more sensitive to these changes. For example, reasoning models like gpt-oss 120b and qwen-3 32b have lower intercoder reliability than non-reasoning models of a similar size. Small models like apertus 8b and llama 8b also show lower intercoder reliability than larger models.

This marginal effect is also reflected in downstream estimates (Table 4). Across different in-context learning settings, the two prompt formats produced similar results in terms of overall congruence of sign and statistical significance.

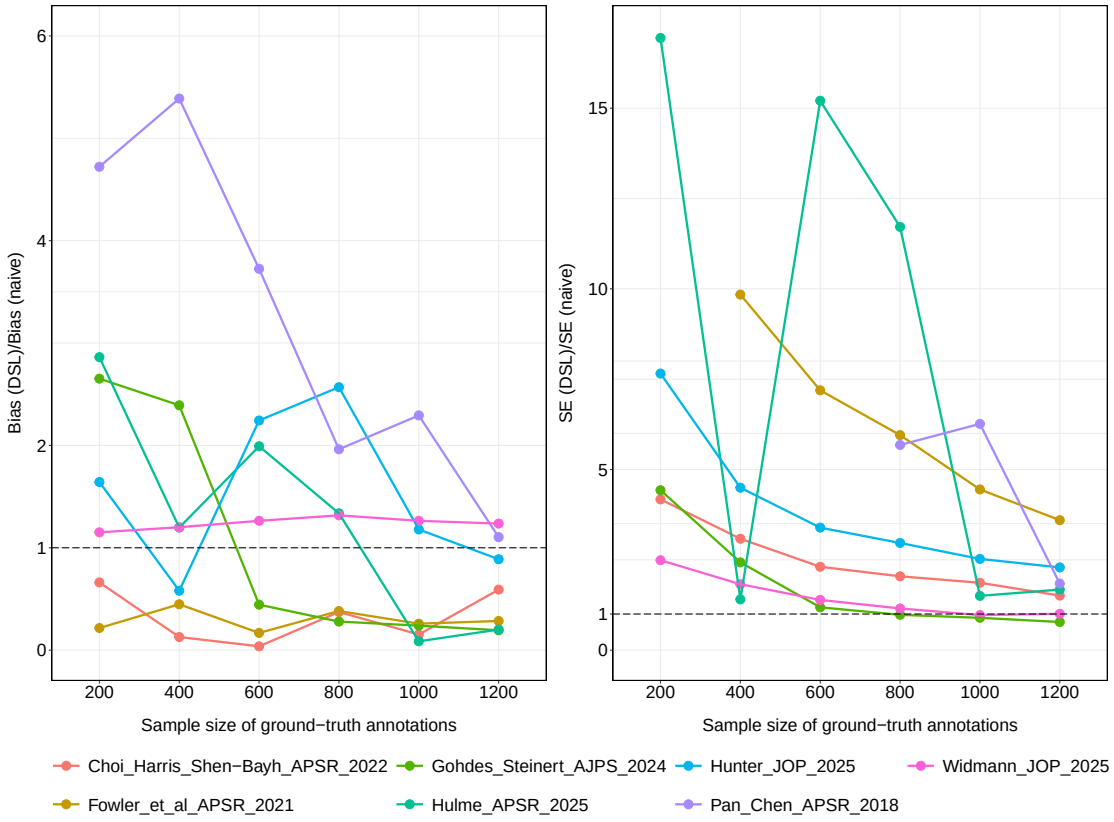
5.6. Bias-correction reduces bias but has costs

Lastly, we evaluate the effectiveness of two recently proposed bias-correction methods, DSL and PRISA, in addressing measurement errors in LLM annotations. Both methods assume that ground-truth annotations are available for a subset of the data. Accordingly, we treat the original annotations and estimates as the ground truth. We quantify the benefits of these methods by measuring the reduction in bias in the coefficient estimates. We also assess their costs by examining the sample size required for the methods to be effective and the efficiency loss they may introduce. For our analysis, we use annotations from llama 70b as the LLM annotations and test the methods across various sample sizes of the ground-truth data. For a given sample size, we randomly sample the ground-truth data 100 times to obtain the distribution of bias-corrected estimates. We report the DSL result in the main text and include the result for PRISA in SM K.

We first report the applicability of both methods to the studies in our sample. Of the 14 studies, seven are compatible with DSL and ten with PRISA. The main reason for DSL's limited applicability is that certain estimators are not currently supported in DSL. For PRISA, the limitation arises when the annotated variable serves as the independent variable, which the method does not currently support.

Figure 8 presents the ratios of bias and standard error between the DSL-corrected and the native LLM estimates. Note that the ratios are averaged across all model specifications within each study. The left panel of the figure shows that DSL can be effective at reducing bias, although its performance depends on the sample size of the ground-truth annotations. At smaller sample sizes, such as 200 or 400, DSL can be counterproductive and even increase bias, as shown by several studies with bias ratios greater than one. On the other hand, the plot reveals a clear and consistent trend: as the sample size of the ground-truth data increases, the bias ratio for nearly all studies decreases. For most studies, a sample size of around 800 to 1,000 is required for DSL to become beneficial, at which point their bias ratios

FIGURE 8. BIAS AND STANDARD ERROR COMPARISONS: DSL VS. NAIVE ESTIMATES



Notes: The figure presents bias and standard error comparisons between DSL and naive estimates. The y-axis shows the ratio of the DSL estimate's bias (left panel) or standard error (SE) (right panel) to that of the naive estimate from predicted annotations. Ratios below the dotted line (y=1) indicate that the DSL estimates have smaller bias or standard errors, respectively. Each colored line represents a different study, plotted against the sample size of the ground-truth annotations used for correction. Some SE ratios for Pan and Chen (2018) and Fowler et al. (2021) are excluded because their values exceed 17.

fall below one, indicating a reduction in bias compared to the naive estimate.

However, this improvement in bias comes at the cost of statistical efficiency, as shown in the right panel. The standard errors of the DSL-corrected estimates are consistently larger than those of the naive estimates. This inflation of variance is particularly severe with smaller ground-truth samples, where the standard error ratio can be extremely large - in some cases exceeding 10. This means that while the corrected point estimates are closer to the true parameter value, their confidence intervals become so wide that they may offer little inferential value. Even with a ground-truth sample size of 1200, the standard error remains 1.5 to 3.5 times larger for most studies. These results highlight the bias-variance

trade-off: while bias-correction methods can address measurement error, they require a large ground-truth dataset to achieve bias reduction without an unacceptable loss in precision.

5.7. Summary

Our findings present a cautionary picture regarding the use of LLMs for social science annotation. While all tested LLMs demonstrate excellent instruction-following capabilities, with annotation validity rates exceeding 99%, their annotations diverge from those produced by original human coders and supervised models. Furthermore, annotation disagreement exists among different LLMs, indicating that the choice of model can introduce an "annotator effect." We show that this variability is not random: agreement tends to be higher for larger models and on tasks grounded in explicit textual evidence, whereas tasks requiring nuanced interpretation yield greater divergence.

These annotation disagreements have consequences for downstream analyses. The choice of LLM leads to variable coefficient estimates, frequently altering the statistical and substantive conclusions of the original studies. We find that common mitigation strategies offer limited relief. In-context learning provides marginal improvements in agreement while increasing computational costs. Bias-correction methods like DSL can reduce bias but require a large ground-truth sample to be effective and may introduce a loss in statistical efficiency.

6. Recommendations

LLMs provide a valuable tool for social scientists to uncover new insights from large, unstructured data. However, using LLMs as data annotators presents new challenges. Our findings are not intended to dissuade researchers from employing LLMs in their research. Rather, our aim is to guide researchers in conducting their analyses more transparently and credibly. To this end, we provide several recommendations based on the findings. To facilitate implementation, in addition to the `R` package, we summarize our recommendations into a checklist at the end.

Don't be afraid of LLMs but use selectively. LLMs can be tremendously useful for research and their utility will only increase in the future. Many of the problems with LLM annotation we highlight in this paper (e.g., measurement error, inconsistency) are also present in annotation by humans or supervised models. We are not arguing that one set of annotators is always better than another. Instead, our aim is to provide researchers with empirical evidence on which LLMs show the most promise, when they might be suitable, and what potential issues researchers should keep in mind.

In terms of annotation task, current LLMs excel at annotations that involve explicit textual evidence but struggle with more implicit, or "latent", concepts. For example, annotation tasks such as extracting specific entities from text, determining the occurrence of events, or identifying clearly expressed sentiment are well-suited for LLMs. In contrast, they tend to have more disagreements with tasks that require deep inferential reasoning, understanding cultural nuance, or detecting subtle forms of rhetoric like irony and nostalgia. For the latter tasks, researchers should carefully compare different annotators (human, LLM, supervised models) before making a choice.

In terms of LLMs, our analysis demonstrates that smaller models and reasoning models are more sensitive to minor prompt changes. Smaller models also tend to have much lower intercoder reliability. Therefore, when computational resources allow, we recommend prioritizing larger models. While models in the 70b-parameter class and above generally offer more robust performance, we consider a model size of at least 12b parameters to be a minimum for producing reliable research outputs.

Transparency and replicability are key. Given the variable and stochastic nature of LLM annotations, it is paramount that the entire annotation workflow is as transparent and replicable as possible. To achieve this, we advocate for three practices.

First, we echo Barrie, Palmer and Spirling (2024) in advocating for the adoption of large open-weight models over proprietary models. Our analysis shows that there is no no-

table difference between proprietary and large open-weight models in annotation quality and downstream effect. However, open-weight models are much more amenable to replication, whereas proprietary models can be updated without notice or discontinued, jeopardizing future replication efforts.

Second, we advocate for the use of replicable LLM inference engines. Popular frameworks such as `SGLang`[10] and `vllm` have implemented deterministic inference, which allows LLM annotations to be exactly replicated given the same hardware and software. The ideal practice is to perform offline inference with open-weight models using these engines. If computational resources are limited, we recommend finding an LLM provider that guarantees replicable inference.

Lastly, we emphasize that documentation is especially important in ensuring the transparency and replicability of LLM annotation. An LLM workflow has many moving parts: prompt design, model versions, hyperparameter values, software versions, and hardware specifics. Any missing information can significantly hinder replication. Therefore, we strongly recommend detailed record-keeping. To aid this process, our `R` package, `localLLM`, provides a function to automatically generate a complete annotation report.

Explore and validate. There are many LLMs and many prompt designs a researcher can choose from. It is best to treat this as an iterative process where a researcher can explore different LLMs and designs and improve on the choice by manually inspecting a sample of the corresponding LLM-generated annotations. This preliminary exploration is crucial for selecting the most suitable LLM, refining the prompt to mitigate misunderstandings, and, importantly, identifying the nature and magnitude of any systematic bias. Once the researcher is confident in the choice of LLM and prompt, they should conduct a systematic validation against a sample of high-quality annotations (ideally labeled by expert coders) to quantify the model's performance and reliability. This process involves calculating standard intercoder reliability metrics, such as Krippendorff's Alpha, to assess the level of agreement

[10] https://docs.sglang.ai/

between the LLM and human coders. When ground-truth annotations are available, researchers could also compute metrics that are robust to dataset imbalance, such as the F1 score. Beyond a single score, we recommend calculating the confusion matrix to understand if the LLM struggles with particular categories. While human coders may no longer be the "gold standard" for some annotation tasks, validation – or the involvement of humans in the annotation workflow – is still valuable in surfacing potential issues. In this regard, LLMs are no different from other data annotators: validation is always essential (Grimmer and Stewart, 2013).

Account for Measurement Error. Finally, given the unpredictable nature of LLMs' measurement errors, it may be difficult to fully account for systematic errors in the annotation process. If expert coders are available and can be trusted to generate high-quality data, we recommend using them to generate a large sample of annotations and applying bias-correction methods to directly address systematic measurement errors. While the number of required annotations is likely context-dependent, a minimum of 1000 annotations serves as a reasonable starting point. Methods like DSL also include power analysis functions that can guide researchers in deciding the sample size. When expert coders are not available or a sufficiently large sample cannot be generated, we recommend conducting sensitivity analyses to quantify the effect of measurement errors on downstream coefficient estimates (Imai and Yamamoto, 2010; Duarte et al., 2024; Bisbee and Spirling, 2025).

6.1. LLM annotation checklist

To help researchers implement these recommendations, we provide the following checklist. While an affirmative answer to every question represents the ideal scenario for justifying the use of an LLM, we recognize that research contexts vary. Therefore, this checklist should be viewed not as a rigid set of prerequisites but as a guiding framework to aid in decision-making and justify methodological choices.

Table 5. Checklist for Using LLMs for Annotation

Phase	Guideline / Question
1. Scoping & Suitability	Does the annotation task involve explicit textual evidence (e.g., words or phrases that are strong predictors of annotation decisions)? Is the dataset large enough that manual annotation by expert coders is infeasible or prohibitively expensive?
2. Model selection & implementation	Are you using a large language model with at least 12b parameters? If not, provide justification for the choice of a smaller model. Are you using a large, open-weight model? If using a proprietary model, provide a justification and acknowledge the potential challenges for future replication. Have you taken steps to ensure the LLM's output is reproducible? (e.g., using a deterministic inference engine). Is the entire annotation workflow thoroughly documented, including prompt, code, and metadata such as software versions?
3. Validation & Analysis	Was a preliminary exploration conducted to test different models and prompts on a data sample to identify the best approach and potential biases? Has the LLM's performance been validated against a high-quality, expert-coded sample using imbalance-robust metrics? Have potential measurement errors been accounted for in the downstream statistical analysis, either through: • Bias-correction methods (if a large ground-truth sample is available)? • Sensitivity analyses (e.g., Bisbee and Spirling, 2025)?

References

Al Kuwatly, Hala, Maximilian Wich and Georg Groh. 2020. Identifying and measuring annotator bias based on annotators' demographic characteristics. In *Proceedings of the fourth workshop on online abuse and harms*. pp. 184–190.

Angelopoulos, Anastasios N, Stephen Bates, Clara Fannjiang, Michael I Jordan and Tijana Zrnic. 2023. "Prediction-powered inference." *Science* 382(6671):669–674.

Barrie, Christopher, Alexis Palmer and Arthur Spirling. 2024. "Replication for language models problems, principles, and best practice for political science." *URL: https://arthurspirling. org/documents/BarriePalmerSpirling TrustMeBro. pdf* .

Barrie, Christopher, Elli Palaiologou and Petter TÃķrnberg. 2024. "Prompt stability scoring for text annotation with large language models." *arXiv preprint arXiv:2407.02039* .

Baumann, Joachim, Paul Röttger, Aleksandra Urman, Albert Wendsjö, Flor Miriam Plaza-del Arco, Johannes B Gruber and Dirk Hovy. 2025. "Large Language Model Hacking: Quantifying the Hidden Risks of Using LLMs for Text Annotation." *arXiv preprint arXiv:2509.08825* .

Bisbee, J. and A. Spirling. 2025. "What to Do When Humans Are No Longer the Gold Standard: Large Language Models, State of the Art and Robustness.".

Bor, Alexander and Michael Bang Petersen. 2022. "The psychology of online political hostility: A comprehensive, cross-national test of the mismatch hypothesis." *American Political Science Review* 116(1):1–18.

Breuer, Adam, Bryce J Dietrich, Michael H Crespin, Matthew Butler, JA Pryse and Kosuke Imai. 2025. "Using AI to Summarize US Presidential Campaign TV Advertisement Videos, 1952-2012." *arXiv preprint arXiv:2503.22589* .

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al. 2020. "Language models are few-shot learners." *Advances in neural information processing systems* 33:1877–1901.

Burnham, Michael. 2024. "Stance detection: a practical guide to classifying political beliefs in text." *Political Science Research and Methods* pp. 1–18.

Choi, Donghyun Danny, J Andrew Harris and Fiona Shen-Bayh. 2022. "Ethnic bias in judicial decision making: Evidence from criminal appeals in Kenya." *American Political Science Review* 116(3):1067–1080.

Duarte, Guilherme, Noam Finkelstein, Dean Knox, Jonathan Mummolo and Ilya Shpitser. 2024. "An automated approach to causal inference in discrete settings." *Journal of the American Statistical Association* 119(547):1778–1793.

Egami, Naoki, Musashi Hinck, Brandon M Stewart and Hanying Wei. 2024. "Using large language model annotations for the social sciences: A general framework of using predicted variables in downstream analyses." *Preprint from November* 17:2024.

Egami, Naoki, Musashi Hinck, Brandon Stewart and Hanying Wei. 2023. "Using imperfect surrogates for downstream inference: Design-based supervised learning for social science applications of large language models." *Advances in Neural Information Processing Systems* 36:68589–68601.

Ennser-Jedenastik, Laurenz and Thomas M Meyer. 2018. "The impact of party cues on manual coding of political texts." *Political Science Research and Methods* 6(3):625–633.

Fowler, Erika Franklin, Michael M Franz, Gregory J Martin, Zachary Peskowitz and Travis N Ridout. 2021. "Political advertising online and offline." *American Political Science Review* 115(1):130–149.

Fowler, Erika Franklin, Michael M. Franz, Travis N. Ridout, Laura Baum, Colleen Bogucki and Breeze Floyd. 2025. "2018, 2020, and 2022 Political TV Advertising.". Combining Versions 1.0, 1.0 & 1.0 [dataset].

Gilardi, Fabrizio, Meysam Alizadeh and Maël Kubli. 2023. "ChatGPT outperforms crowd workers for text-annotation tasks." *Proceedings of the National Academy of Sciences* 120(30):e2305016120.

Gohdes, Anita R. 2020. "Repression technology: Internet accessibility and state violence." *American Journal of Political Science* 64(3):488–503.

Gohdes, Anita R and Zachary C Steinert-Threlkeld. 2025. "Civilian behavior on social media during civil war." *American Journal of Political Science* 69(3):1099–1114.

Grimmer, Justin and Brandon M Stewart. 2013. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political analysis* 21(3):267–297.

Guo, Chuan, Geoff Pleiss, Yu Sun and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR pp. 1321–1330.

Halterman, Andrew and Katherine A. Keith. 2025. "Codebook LLMs: Evaluating LLMs as Measurement Tools for Political Science Concepts." *Political Analysis* p. 1–17.

Hulme, M Patrick. 2025. "War and Responsibility." *American Political Science Review* pp. 1–24.

Hunter, Tom. 2025. "Credit claiming in the European Union." *The Journal of Politics* 87(3):889–904.

Imai, Kosuke and Teppei Yamamoto. 2010. "Causal inference with differential measurement error: Nonparametric identification and sensitivity analysis." *American Journal of Political Science* 54(2):543–560.

Krippendorff, Klaus. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Le Mens, Gaël and Aina Gallego. 2025. "Positioning political texts with large language models by asking and averaging." *Political Analysis* 33(3):274–282.

Li, Zeren. 2023. "Connections as liabilities: The cost of the politics–business revolving door in China." *British Journal of Political Science* 53(4):1252–1272.

Lin, Gechun. 2025*a*. "Using cross-encoders to measure the similarity of short texts in political science." *American Journal of Political Science* .

Lin, Shengqiao. 2025*b*. "Addressing Risk by Doing Good: Business Response to Government Policy Initiative." *The Journal of Politics* 87(4):000–000.

Mellon, Jonathan, Jack Bailey, Ralph Scott, James Breckwoldt, Marta Miori and Phillip Schmedeman. 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.

Milliff, Aidan. 2024. "Making sense, making choices: How civilians choose survival strategies during violence." *American Political Science Review* 118(3):1379–1397.

Müller, Stefan and Naofumi Fujimura. 2025. "Campaign communication and legislative leadership." *Political Science Research and Methods* 13(3):545–566.

Müller, Stefan and Sven-Oliver Proksch. 2024. "Nostalgia in European party politics: a text-based measurement approach." *British Journal of Political Science* 54(3):993–1005.

Pan, Jennifer and Kaiping Chen. 2018. "Concealing corruption: How Chinese officials distort upward reporting of online grievances." *American Political Science Review* 112(3):602–620.

Raleigh, Clionadh, Rew Linke, Håvard Hegre and Joakim Karlsen. 2010. "Introducing ACLED: An armed conflict location and event dataset." *Journal of Peace Research* 47(5):651–660.

Rozenas, Arturas and Denis Stukal. 2019. "How autocrats manipulate economic news: Evidence from Russia's state-controlled television." *The Journal of Politics* 81(3):982–996.

Sap, Maarten, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi and Noah A Smith. 2021. "Annotators with attitudes: How annotator beliefs and identities bias toxic language detection." *arXiv preprint arXiv:2111.07997* .

Timoneda, Joan C and Sebastián Vallejo Vera. 2025. "Memory Is All You Need: Testing How Model Memory Affects LLM Performance in Annotation Tasks." *arXiv preprint arXiv:2503.04874* .

Widmann, Tobias. 2025. "Do politicians appeal to discrete emotions? The effect of wind turbine construction on elite discourse." *The Journal of Politics* 87(1):335–346.

Ziems, Caleb, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang and Diyi Yang. 2024. "Can large language models transform computational social science?" *Computational Linguistics* 50(1):237–291.

Appendix *for*

Data Annotation with Large Language Models: Lessons from A Large Reanalysis Study

Table of Contents

A. Alternative prompt format

Instead of the Markdown format we used for the main prmopts, the alternative prompt format uses XML tags (<> and </>) to enclose each section. An example of the alternative prompt format is shown in Figure A1.

FIGURE A1. PROMPT FORMAT

<annotation_task>
Your task is to classify the target text, a sentence from a political party manifesto, based on whether it contains nostalgic rhetoric.
</annotation_task>

<coding_rules>
Nostalgia is defined as a predominantly positive emotion associated with recalling important or momentous past events, usually experienced collectively with close others or as part of a national identity. Return 1 if the text is nostalgic, and 0 if it is not.

* Not Nostalgic (Code 0): The text does not contain a positive, emotional reference to the past. This includes statements that are future-oriented, purely factual descriptions of the past, or negative critiques of past events or governments.
* Nostalgic (Code 1): The text positively and emotionally references a collective past. The text should express a longing for or a proud recollection of a nation's history, heritage, traditions, or a bygone era.
</coding_rules>

<examples>
Better protection of data transferred to the US warden
Answer: 0

We build the Estonian National Museum and the Tallinn Music High School, new buildings.
Answer: 1
</examples>

<target_text>
{{target-text}}
</target_text>

<output_format>
{"answer": "Your choice here”}
</output_format>

Remember to replace the placeholder text in the "answer" field with your actual annotation. Your annotation must be one of the numerical codes (1 or 0). Respond only with a valid JSON object and nothing else.

Notes: The figure illustrates an example of the alternative prompt format used in the annotation procedure. The prompt is organized into sections, with each enclosed by the <> and </> tags. It includes the task definition, coding rules, illustrative examples, the target text placeholder, and the required output format. The prompt is also an example of “2-shot learning”, where two annotated examples are provided as demonstrations before the target text.

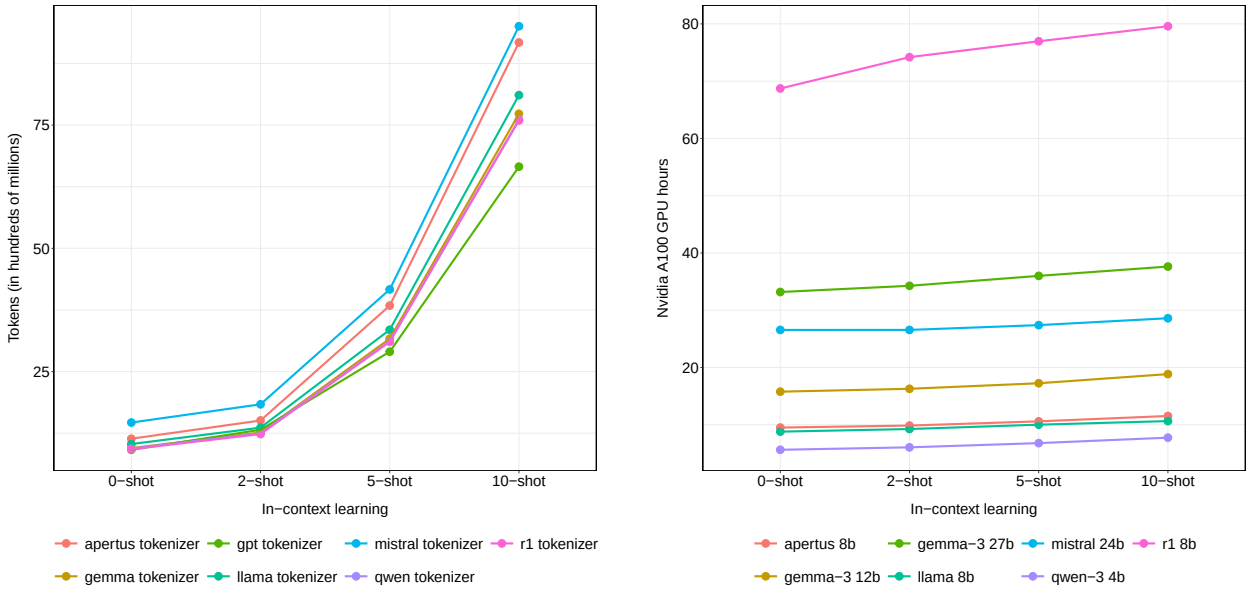
B. Additional annotation details

We download all open-weight models in the study from Hugging Face. We use `vllm` as the inference engine for all annotations using open-weight models. We follow the reproducibility guide from `vllm` (https://docs.vllm.ai/en/v0.10.1/usage/reproducibility.html) by setting the "VLLM_ENABLE_V1_MULTIPROCESSING" environment variable to 0. For non-reasoning models, we use greedy decoding by setting the temperature to zero. For reasoning models, we follow the best practice by using the default temperature of each model. We do not set the temperature to zero for reasoning models as it can degrade the chain of thought quality and the overall performance of these models. We verify that annotations from reasoning models are still reproducible in this setting. For reasoning models, we additionally set the maximum number of output tokens to 4096. Annotations based on proprietary models were obtained through OpenAI's API from June to October 2025. Similar to open-weight models, we set the temperature to zero for non-reasoning proprietary models (gpt-4o mini, gpt-4.1 mini) and use the default temperature for the reasoning model (gpt-5).

We use Nvidia A100 GPUs for inference with the following models: gemma-3 27b, mistral 24b, gemma-3 12b, apertus 8b, llama 8b, r1 8b, qwen-3 4b. We use Nvidia H100 GPUs for inference with the following models: gpt-oss 120b, qwen-2.5 72b, llama 70b, qwen3 32b, gpt-oss 20b. The total GPU hours used for the study are 2448.

As Panel (a) of Figure A2 shows, in-context learning drastically increases the total number of input tokens. The total number of input tokens for 10-shot learning can be as many as eight times that for 0-shot learning. Panel (b) shows that the larger input also slows computation as the total inference time increases steadily from 0-shot to 10-shot.

Figure A2. Token count and inference time comparison



(a) Token count by tokenizer and in-context learning

(b) Inference time by model and in-context learning

Notes: Panel (a) shows the total token count (in hundreds of millions) by tokenizer and in-context learning. Panel (b) shows the inference time by model and in-context learning. Larger models are not included in Panel (b) because they require multiple GPUs for inference and the speed partly depends on the number of GPUs used.

C. Replication notes

Table A1. Replication notes by study

Study	Notes
Choi, Harris & Shen-Bayh (2022)	Exactly replicated.
Fowler et al. (2021)	The replication code was based on both the original code for replicating Figure B.4.(c) provided by the paper authors, as well as the codes for merging ad content from DSL authors. We used the actual human annotations to replace the ML coding results used in the original code.

Continued on next page

Study	Notes
Gohdes (2020)	Estimates cannot be exactly replicated as the data processing script contained a sampling step with no random seed. We manually added a random seed while replicating this work.
Gohdes & Steinert-Threlkeld (2025)	Exactly replicated.
Hulme (2025)	Exactly replicated.
Hunter (2025)	Replicated results based on the original dataset and code are slightly different from the regression table in the main text.
Li (2023)	Replication was done in R (original analysis was done in STATA). Results based on the original dataset are slightly different from the regression table in the main text (incl. estimate, observations, adjusted R square).
Lin (2025)	The IDs of Q&A texts in two replication datasets were not consistent. Therefore, for duplicated texts, we used the first annotation. Fortunately, the original annotations were always the same for the duplicated texts. Thus, the replicated estimates were the same as the outputs in the paper.
Milliff (2024)	This study used Bayesian Multinomial Logistic Regression, which did not have standard error, but standard deviation.
Müller & Fujimura (2025)	Exactly replicated.
Pan & Chen (2018)	Exactly replicated.
Müller & Proksch (2024)	The replication dataset has five fewer sentences (N = 1,192,675) than the reported number of observations (N = 1,192,680). The replicated result for M5 (column 5) is slightly different from that reported in the paper.
Rozenas & Stukal (2019)	Estimates cannot be exactly replicated as one of the required packages is not longer available.
Widmann (2025)	Exactly replicated.

D. Annotation details for each study

Choi, Harris & Shen-Bayh (2022). The annotation procedure involved classifying two different types of outcomes from the text of legal judgments. The text data consists of a corpus of 9,545 criminal appeal rulings from the Kenyan High Court between 2003 and 2017. The primary outcome of each appeal - whether it was allowed or denied - was annotated. This was a hybrid annotation process: the authors initially used an automated method with regular expressions to classify the verdicts, but for cases where this was insufficient due to varied judicial writing styles, human coders were used to complete the classification. This binary "allowed/denied" annotation served as the main dependent variable in their statistical analysis to determine if a coethnic match between a judge and an appellant affected the case outcome.

Fowler et al (2021). The text data is derived from television ad creatives, which include transcribed audio. This data was initially annotated by human coders at the Wesleyan Media Project. These coders classified each TV ad based on a variety of characteristics, most notably its tone (whether it was positive, attack, or contrast) and the specific policy issues that were mentioned. This human-annotated data on TV ads, along with a similar human-coded sample of Facebook ads, was then used as a training set to build a supervised learning classification model. The final "annotations" used in the downstream statistical analysis were the predicted probabilities for tone and issue content generated by this model for every ad in the full dataset. These model-generated predictions served as the dependent variables in the authors' regression analyses to compare advertising content across platforms.

Gohdes (2020). The text data used in the article consists of over 65,000 aggregated reports on individual killings committed by the Syrian regime, compiled from four different human rights documentation groups. The purpose of the annotation was to classify each killing as either targeted or untargeted based on the textual descriptions of the event. The

annotation process was a hybrid of human and machine effort: the author first manually classified a training set of 2,347 records based on operational definitions (e.g., executions were "targeted," shelling was "untargeted"). This human-annotated data was then used to train a supervised machine learning model (xgboost) to automatically classify the remaining records. The final annotations (counts of targeted vs. untargeted killings) were aggregated by governorate and time period and used as the dependent variable in a binomial regression analysis to test how internet accessibility affects the proportion of targeted state violence.

Gohdes & Steinert-Threlkeld (2025). The text data consists of Arabic-language tweets from users in Syria collected before and after the siege of Aleppo in 2016. The data was annotated for two main features: topic (pro-Assad vs. anti-Assad) and sentiment (positive, negative, neutral). The annotation was performed through a multi-step process. Initially, human annotators (three native Syrian Arabic undergraduates) labeled a "gold standard" set of 6,000 tweets for their topic. This human-labeled data was then used to train and validate several classifiers, with a supervised large language model (ARBERT) being selected to annotate the topic of the entire dataset. Sentiment was also assigned using a fine-tuned ARaBERT model. In the downstream statistical analysis, these topic and sentiment annotations were used as the primary dependent variables in logistic regression models to test how the content of civilian posts changed after the shift in territorial control.

Hulme (2025). The annotation procedure was designed to measure congressional sentiment on the use of military force from congressional floor speeches. The text data consists of speeches from the Congressional Record: approximately 30,000 speeches from key foreign policy leaders. For each speech, annotators coded for expressions of support or opposition to military action, broken down by type (e.g., general force, ground troops, air assets). The corpus was hand-labeled by a team of 15 undergraduate research assistants. These annotations were then aggregated to create a quantitative "Congressional Support Score" for each crisis, which was used as a key independent variable in downstream statistical analyses to

predict the level of U.S. military force used.

Hunter (2025). The text data used in the study consists of over 6,000 paragraphs, or "statements," extracted from 414 speeches given by heads of government from seven EU member states between 2005 and 2018. These speeches presented the outcomes of European Council summits to their respective national media. The annotation was performed by human hand-coders, who classified each statement into one of four attributional categories: "credit claiming" (by the national government), "credit sharing" (with the EU or other states), "blame shifting" (onto the EU), or "descriptive" (no attribution). The statements were also annotated with their corresponding policy area. For downstream statistical analysis, these categorical annotations were converted into binary dependent variables (e.g., a statement was coded as 1 for "credit claiming" and 0 otherwise) and used in a series of multilevel logistic regression models to test the author's hypotheses.

Li (2023). The text data consists of over 1.2 million equity research reports published by major financial institutions, from which the author extracted 570,000 sentences specifically pertaining to publicly listed firms. The property being annotated from this text is investor sentiment. The annotation was performed by a supervised learning model trained to classify the sentiment of each sentence into one of two categories: "positive" or "negative." These machine-generated annotations were then aggregated to create a firm-level variable for downstream statistical analysis. Specifically, the author calculated the ratio of positive sentiment reports to the total number of reports for each firm in each year. This ratio was then used as a dependent variable in a regression model to empirically test whether politically connected firms suffered from more negative external perceptions.

Lin (2025). The text data consists of 418,480 question-and-answer (Q&A) transcripts from meetings between institutional investors and the leadership of publicly listed Chinese firms from 2012 to 2019. Each Q&A conversation was annotated with a binary label, classifying

it as either "political" or "nonpolitical," where a "political" conversation was defined as one that explicitly mentioned governments, public policies, or politicians. The annotation was performed in two stages: first, a sample of 4,000 Q&As was labeled by three trained human annotators to create a training dataset. Then, this human-annotated data was used to fine-tune a supervised machine learning model (BERT), which classified the entire dataset. For downstream statistical analysis, these annotations were used to calculate a Political Risk Index (PRI) for each firm-year, representing the percentage of its Q&As that were political. This PRI variable was then used as the key independent variable in difference-in-differences models to measure its effect on firms' spending on poverty alleviation programs.

Milliff (2024). The text data consists of transcripts from over 500 oral histories of Sikh survivors of political violence, collected by the 1984 Living History Project, supplemented by 30 original interviews conducted by the author. The goal was to annotate two key pieces of information from this text: 1) the survivor's chosen survival strategy (categorized as Flee, Fight, Hide, or Adapt), and 2) their situational appraisals, specifically their sense of "control" and "predictability" regarding the violence. The annotation was performed using a dual-method approach to ensure robustness. First, the author acted as a human annotator, manually labeling appraisals and strategies in 221 histories based on pre-defined coding rules. Second, a supervised machine learning model (MuRIL) was fine-tuned on thousands of human-labeled sentences to automatically classify appraisals across the larger corpus. In the downstream statistical analysis, these annotated appraisals of control and predictability were used as the primary independent variables in multinomial logistic regression models to predict the probability of a civilian choosing a specific survival strategy.

Müller & Fujimura (2025). The annotation procedure was a multi-stage process designed to classify policy emphasis in Japanese political manifestos. The text data consisted of over 46,900 individual statements (sentences or bullet points) segmented from 1,270 candidate manifestos collected across five elections. The goal was to annotate each statement

with one of eleven specific policy areas (e.g., "Agriculture, Forestry, and Fisheries," "Foreign Affairs") that correspond to government ministries and Diet committees. The annotation was performed in two main steps: first, a sample of 3,000 statements was manually coded by two trained human annotators to create a reliable ground-truth dataset. This human-annotated data was then used to train and fine-tune a supervised transformer-based (BERT) machine learning model, which subsequently classified the entire corpus of statements. For the downstream statistical analysis, these annotations were aggregated for each manifesto to create the primary independent variable, "Manifesto Salience," which measured the proportion of a candidate's manifesto dedicated to a specific policy area. This variable was then used in logistic regression models to predict whether a candidate would later secure a legislative leadership post in that same policy area.

Müller & Proksch (2024). The text data consists of 1,648 party manifestos from 24 European countries, which were machine-translated into English. The unit of annotation was the individual sentence. The core task was to classify each sentence as either containing nostalgic rhetoric or not. This was performed by a combination of human and automated annotators. Initially, a training set of 1,200 sentences was hand-coded by four human research assistants to create a gold-standard dataset, with inter-coder reliability being measured to ensure consistency. This human-annotated data was then used to train and validate several automated methods, most notably two supervised machine learning models: a Support Vector Machine (SVM) and a Transformer-based classifier (DistilBERT). For the downstream statistical analysis, the sentence-level classifications were aggregated to create a "nostalgia score" for each manifesto (the number of nostalgic sentences per 1,000). This score was then used as the dependent variable in regression models to investigate which factors, such as party ideology and party family, predict the level of nostalgia in political communication.

Pan & Chen (2018). The text data consists of 3,423 "negative sentiment issues," which are essentially citizen complaints, extracted from 643 Online Sentiment Monitoring Reports

produced by the J. Prefecture Propaganda Department between 2012 and 2014. After deduplication, the final analysis is based on 1,412 unique complaints from 2014. The researchers performed manual annotation on these complaints to create several key variables for their analysis. Specifically, they annotated whether a complaint accused the prefecture-level government of wrongdoing, whether it accused a subordinate county-level government of wrongdoing, if it was based on personal experience, pertained to a group issue, or involved collective action or petitions. These human-generated annotations were converted into binary variables and used as the primary independent and control variables in a logistic regression model to predict the likelihood of a complaint being reported upward to provincial authorities.

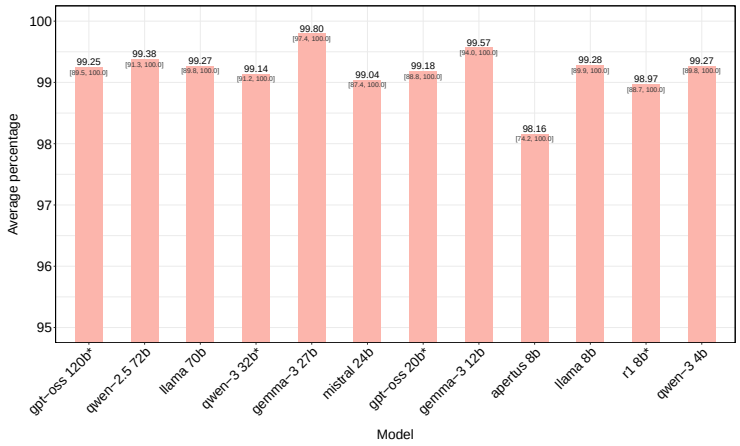
Rozenas & Stukal (2019). The annotation procedure was conducted on a corpus of daily news reports from Russia's largest state-owned television network, Channel 1, from 1999 to 2016. The specific text data annotated was a random sample of 6,706 short news fragments (3-10 sentences each) concerning the Russian economy. The annotation was performed by 544 Russian-speaking human workers via the crowdsourcing platform CrowdFlower. These annotators were tasked with two main judgments: 1) identifying specific economic events, labeling them as "good" or "bad" news, and identifying the actor to whom the event was directly attributed (e.g., Putin, foreign powers); and 2) assessing the overall sentiment (positive, neutral, or negative) of the entire news fragment. These human-generated annotations were then used as the primary data in downstream statistical analyses, such as probit regressions, to quantitatively test the hypothesis that good news is systematically attributed to domestic leaders while bad news is blamed on external factors.

Widmann (2025). The text data for the study consists of parliamentary speeches from German Members of Parliament (MPs) from 2017 to 2020. The objective was to annotate these speeches for the presence of eight discrete emotional appeals, specifically anger, fear, disgust, sadness, joy, enthusiasm, pride, and hope. The annotation was performed by a su-

pervised, transformer-based machine learning model (an Electra classifier). This model had been previously trained on a separate corpus of nearly 10,000 German political sentences that were manually labeled for the eight emotions by human crowd-workers. For the downstream statistical analysis, the model's sentence-level annotations were aggregated to create a quantitative variable: the average proportion of sentences appealing to each specific emotion, calculated per MP per month. This proportion then served as the dependent variable in staggered difference-in-difference regression models to measure the effect of wind turbine construction on politicians' emotional rhetoric.

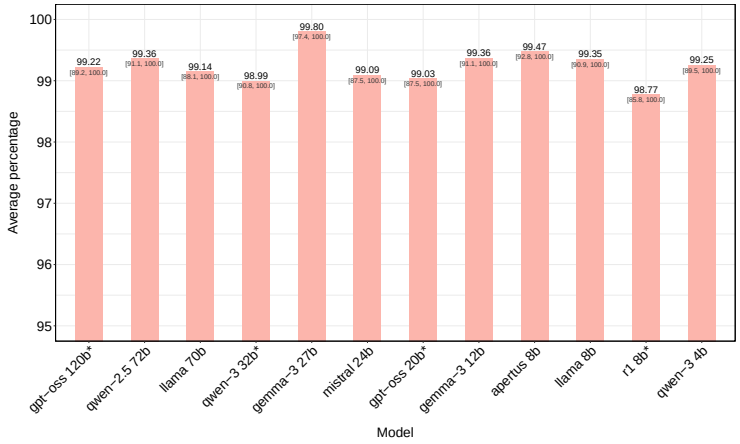
E. Additional annotation results

FIGURE A3. AVERAGE PERCENTAGE OF VALID LABELS BY MODEL (2-SHOT, MAIN PROMPT)



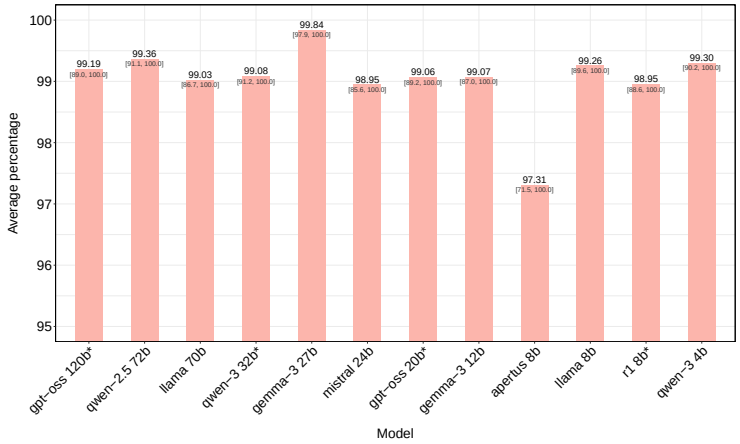
Notes: The figure presents the average percentage of valid labels by LLM for 2-shot annotations. The number on top of each bar indicates the average percentage for the corresponding LLM and the numbers in brackets indicate the range of percentages across the 14 studies.

FIGURE A4. AVERAGE PERCENTAGE OF VALID LABELS BY MODEL (5-SHOT, MAIN PROMPT)



Notes: The figure presents the average percentage of valid labels by LLM for 5-shot annotations. The number on top of each bar indicates the average percentage for the corresponding LLM and the numbers in brackets indicate the range of percentages across the 14 studies.

Figure A5. Average Percentage of Valid Labels by Model (10-shot, main prompt)

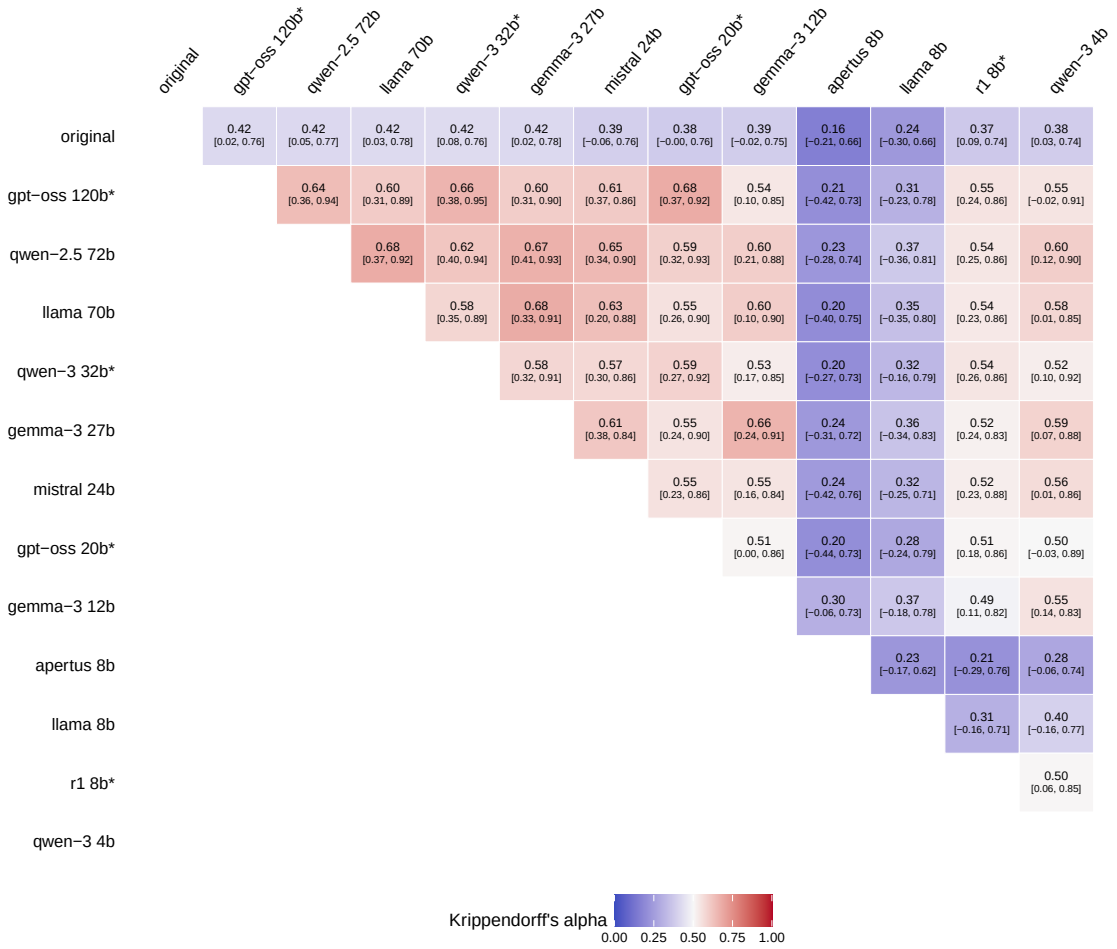


Notes: The figure presents the average percentage of valid labels by LLM for 10-shot annotations. The number on top of each bar indicates the average percentage for the corresponding LLM and the numbers in brackets indicate the range of percentages across the 14 studies.

Table A2. Percentage of Valid Labels by Model (alternative prompt)

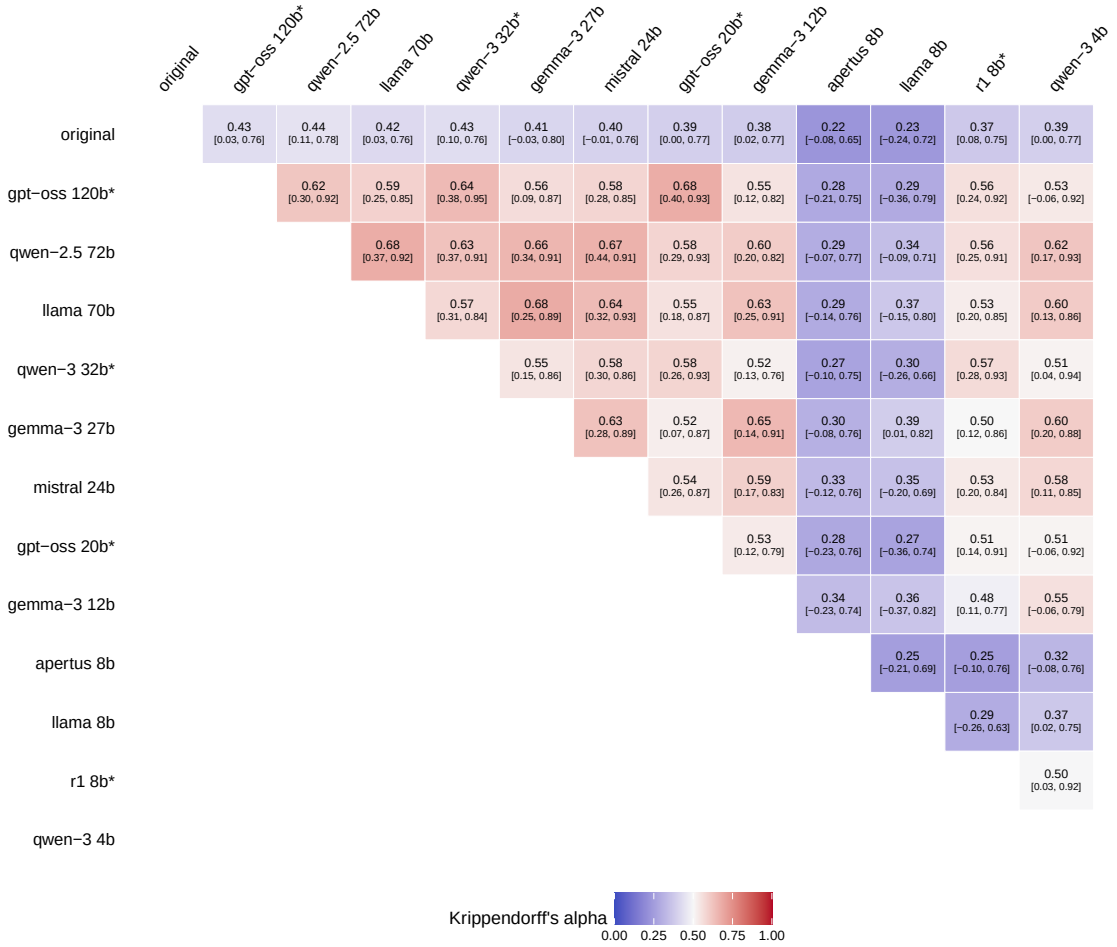
	0-Shot		2-Shot		5-Shot		10-Shot	
Model	**Avg.**	**Low.**	**Avg.**	**Low.**	**Avg.**	**Low.**	**Avg.**	**Low.**
gpt-oss 120b*	99.4	91.9	99.2	89.6	99.3	89.7	99.2	89.6
qwen-2.5 72b	99.7	95.2	99.3	90.7	99.3	90.6	99.3	90.7
llama 70b	99.4	91.6	99.1	88.0	99.1	87.7	99.0	86.6
qwen-3 32b*	99.4	94.1	99.2	91.6	99.1	91.5	99.2	92.4
gemma-3 27b	99.8	97.0	99.7	95.7	99.7	96.3	99.8	97.0
mistral 24b	99.6	93.9	99.2	88.6	99.2	89.0	99.2	88.4
gpt-oss 20b*	99.4	92.2	99.0	86.6	98.9	85.1	98.8	87.1
gemma-3 12b	99.8	96.5	99.6	93.9	99.2	89.1	99.2	89.0
apertus 8b	98.3	76.6	99.1	87.4	99.8	97.0	99.7	96.6
llama 8b	99.3	90.7	99.4	92.1	99.2	88.6	99.2	89.2
r1 8b*	99.3	92.7	99.0	89.3	98.8	87.0	99.1	90.2
qwen-3 4b	99.8	96.9	99.3	90.8	99.3	89.6	99.3	90.7

Figure A6. Heatmap of pairwise intercoder reliability (2-shot)



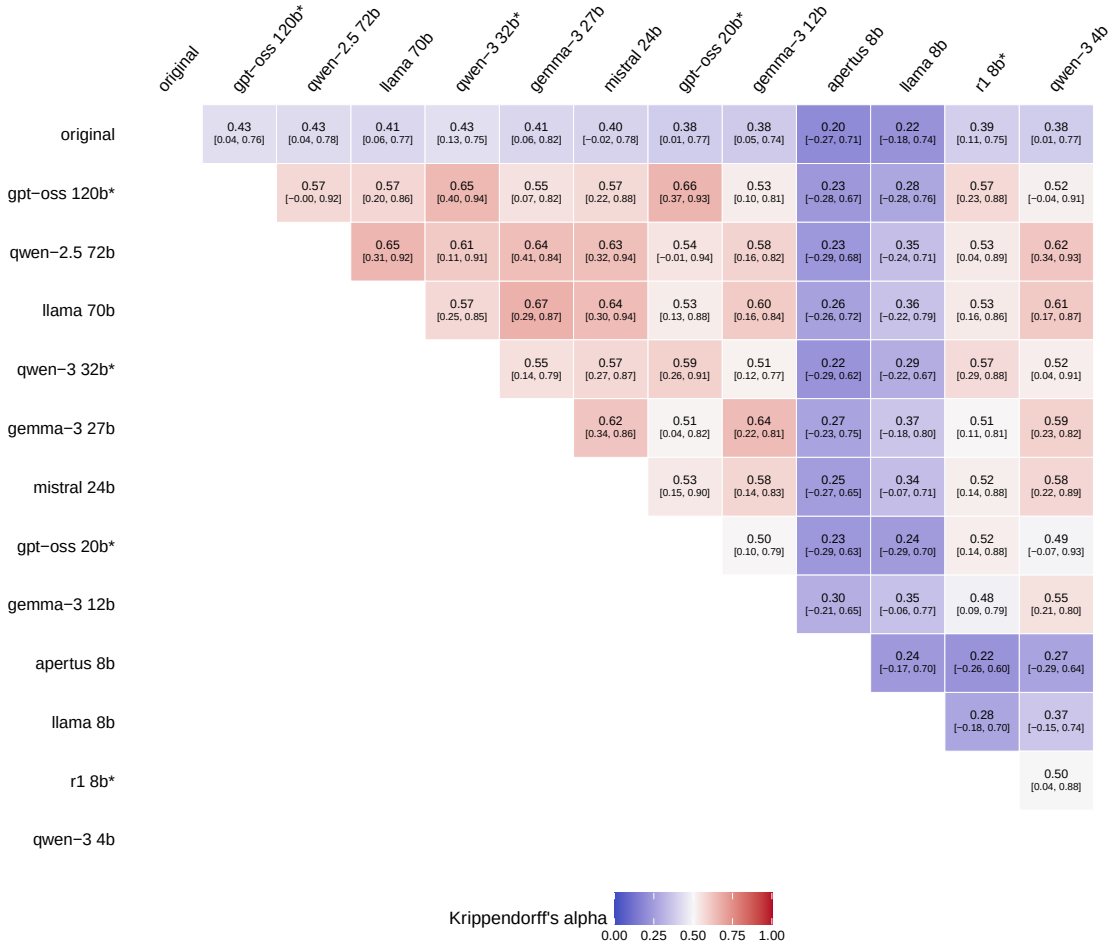
Notes: The figure presents Krippendorff's alphas for all pairs of annotators, averaged across the 14 studies. The numbers in brackets indicate the minimum and maximum alphas for each pair. "Original" indicates the original annotator(s) of the 14 studies. Reasoning models are suffixed with an asterisk (*).

Figure A7. Heatmap of pairwise intercoder reliability (5-shot)



Notes: The figure presents Krippendorff's alphas for all pairs of annotators, averaged across the 14 studies. The numbers in brackets indicate the minimum and maximum alphas for each pair. "Original" indicates the original annotator(s) of the 14 studies. Reasoning models are suffixed with an asterisk (*).

FIGURE A8. HEATMAP OF PAIRWISE INTERCODER RELIABILITY (10-SHOT)



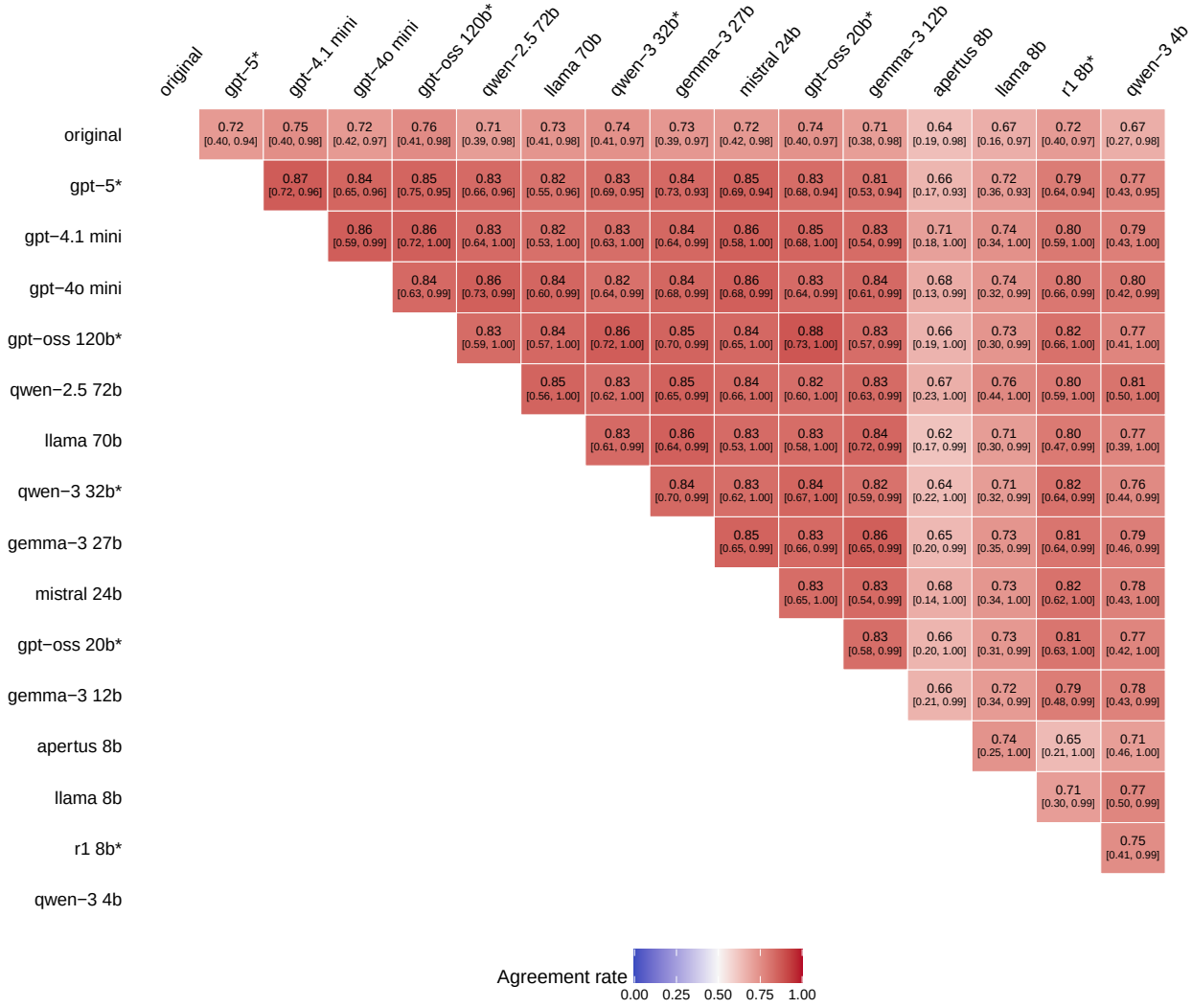
Notes: The figure presents Krippendorff's alphas for all pairs of annotators, averaged across the 14 studies. The numbers in brackets indicate the minimum and maximum alphas for each pair. "Original" indicates the original annotator(s) of the 14 studies. Reasoning models are suffixed with an asterisk (*).

F. Results using simple agreement rate

Figure A9 examines the simple agreement rates among pairs of annotators. In stark contrast to the results using intercoder reliability, LLMs show high simple agreement rates, both with the original annotators and among the LLMs. This is in accordance with existing studies that document "high performance" of LLMs for annotation tasks (Gilardi, Alizadeh and Kubli, 2023).

However, Table A3 reveals that simple agreement rate may not be the most suitable metric to quantify reliability for political science annotation tasks. Table A3 shows that, in our sample of studies, most annotation datasets are highly imbalanced in their category distribution. In these cases, even when LLMs have an extremely high agreement rate (e.g., $> 99\%$), if the errors/disagreements are mostly coming from the minority category, it will have a big influence on the downstream estimates.

FIGURE A9. HEATMAP OF PAIRWISE SIMPLE AGREEMENT RATE (0-SHOT)



Notes: The figure presents simple agreement rates for all pairs of annotators, averaged across the 14 studies. The numbers in brackets indicate the minimum and maximum alphas for each pair. "Original" indicates the original annotator(s) of the 14 studies. Reasoning models are suffixed with an asterisk (*).

TABLE A3. DISTRIBUTION OF LABELS ACROSS STUDIES

Code	Label	Count	Percentage
Choi_Harris_Shen-Bayh_APSR_2022			
0	Appeal unsuccessful	4134	43.31
1	Appeal successful	5411	56.69
Fowler_et_al_APSR_2021			
0	None/Other	2	0.01
1	Contrast	2566	17.76
2	Promote	10,818	74.85
3	Attack	1066	7.38
Gohdes_AJPS_2020			
1	Untargeted Killing	52,339	80.18
2	Targeted Killing	10,489	16.07
3	Other	2446	3.75
Gohdes_Steinert_AJPS_2024			
-1	NEGATIVE	11,969	34.78
0	NEUTRAL	7502	21.80
1	POSITIVE	14,941	43.42
Hulme_APSR_2025			
0	NULL	20,743	74.59
1	GenSupp	3690	13.27
2	GrdSupp	99	0.36
3	AirSupp	281	1.01

Code	Label	Count	Percentage
4	NavSupp	57	0.20
5	GenOpp	2545	9.15
6	GrdOpp	244	0.88
7	AirOpp	121	0.44
8	NavOpp	31	0.11
Hunter_JOP_2025			
1	Credit Claiming	1158	19.49
2	Credit Sharing	1176	19.79
3	Blame Shifting	43	0.72
4	Descriptive	3566	60.00
Li_BJPS_2022			
0	Negative	175,074	30.27
2	Positive	403,337	69.73
Lin_JOP_2024			
0	Non-political	364,459	87.09
1	Political	54,021	12.91
Milliff_APSR_2023			
0	Both Low CONTROL and PREDICTABILITY	1495	40.96
1	High CONTROL but Low PREDICTABILITY	584	16.00
2	Low CONTROL but High PREDICTABILITY	499	13.67
3	Both High CONTROL and PREDICTABILITY	1072	29.37
Muller_Fujimura_PSRM_2024			

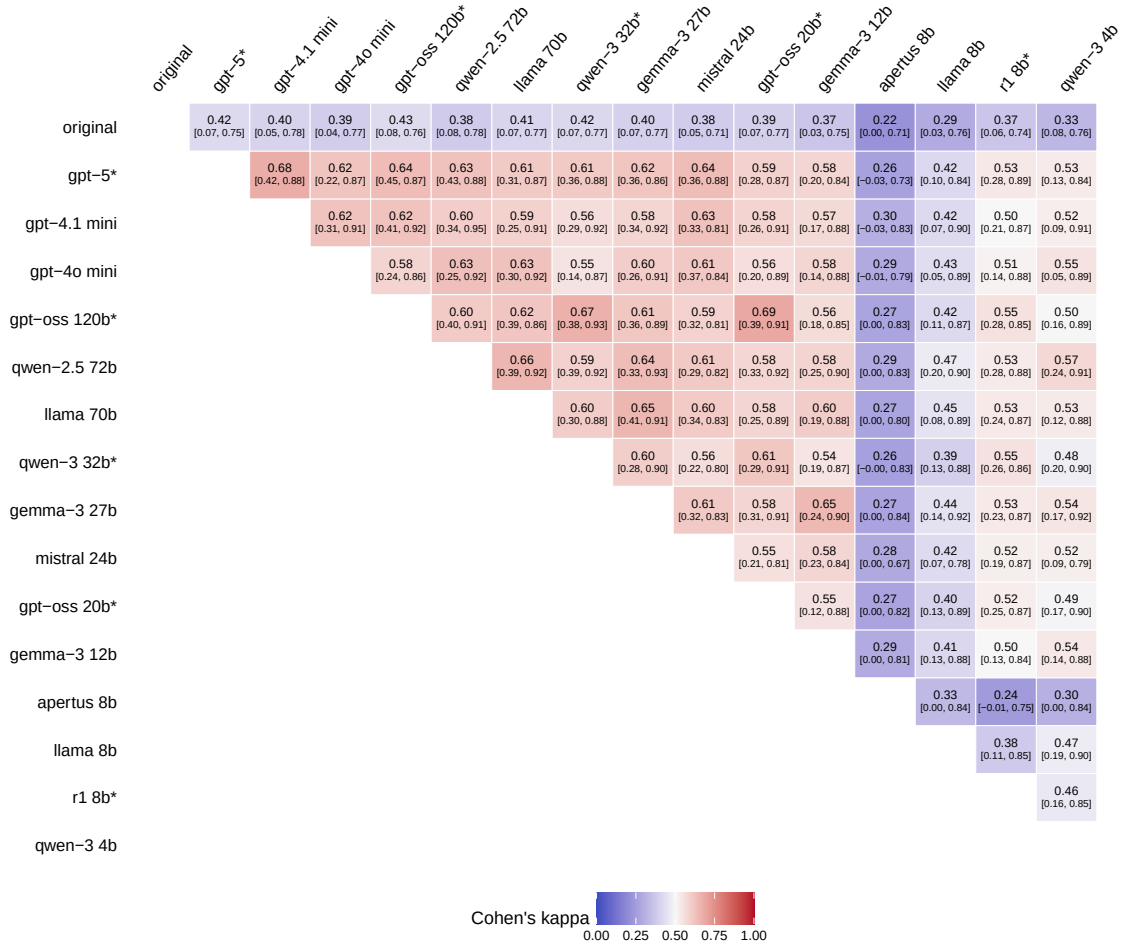
Code	Label	Count	Percentage
1	Agriculture, Forestry and Fisheries	2590	4.34
2	Committees on Cabinet	5847	9.81
3	Economy, Trade and Industry	2930	4.91
4	Education, Culture, Sports, Science and Technology	3793	6.36
5	Environment	886	1.49
6	Financial Affairs	2965	4.97
7	Foreign Affairs	1540	2.58
8	Health, Labor and Welfare	9205	15.44
9	Internal Affairs and Communications	2069	3.47
10	Land, Infrastructure, Transport and Tourism	2635	4.42
11	Security	1338	2.24
12	No specific policy area/Other	23,821	39.96
Muller_Proksch_BJPS_2024			
0	Not Nostalgic	1,163,397	97.55
1	Nostalgic	29,278	2.45
Pan_Chen_APSR_2018			
0	Neither Prefecture nor County Wrongdoing	1013	71.74
1	Only Prefecture Wrongdoing	76	5.38
2	Only County Wrongdoing	321	22.73
3	Both Prefecture and County Wrongdoing	2	0.14
Rozenas_Stukal_JOP_2018			
1	V.Putin personally	562	13.02
2	RUSSIAN authorities/officials	2144	49.65

Code	Label	Count	Percentage
3	Large RUSSIAN business companies	550	12.74
4	FOREIGN governments	139	3.22
5	FOREIGN economies or large business	357	8.27
6	Neither/Not applicable	98	2.27
Widmann_JOP_2025			
0	Neither Anger nor Disgust	373,812	59.61
1	Only Anger	245,488	39.15
2	Only Disgust	22	0.00
3	Both Anger and Disgust	7780	1.24

G. Results using Cohen's kappa

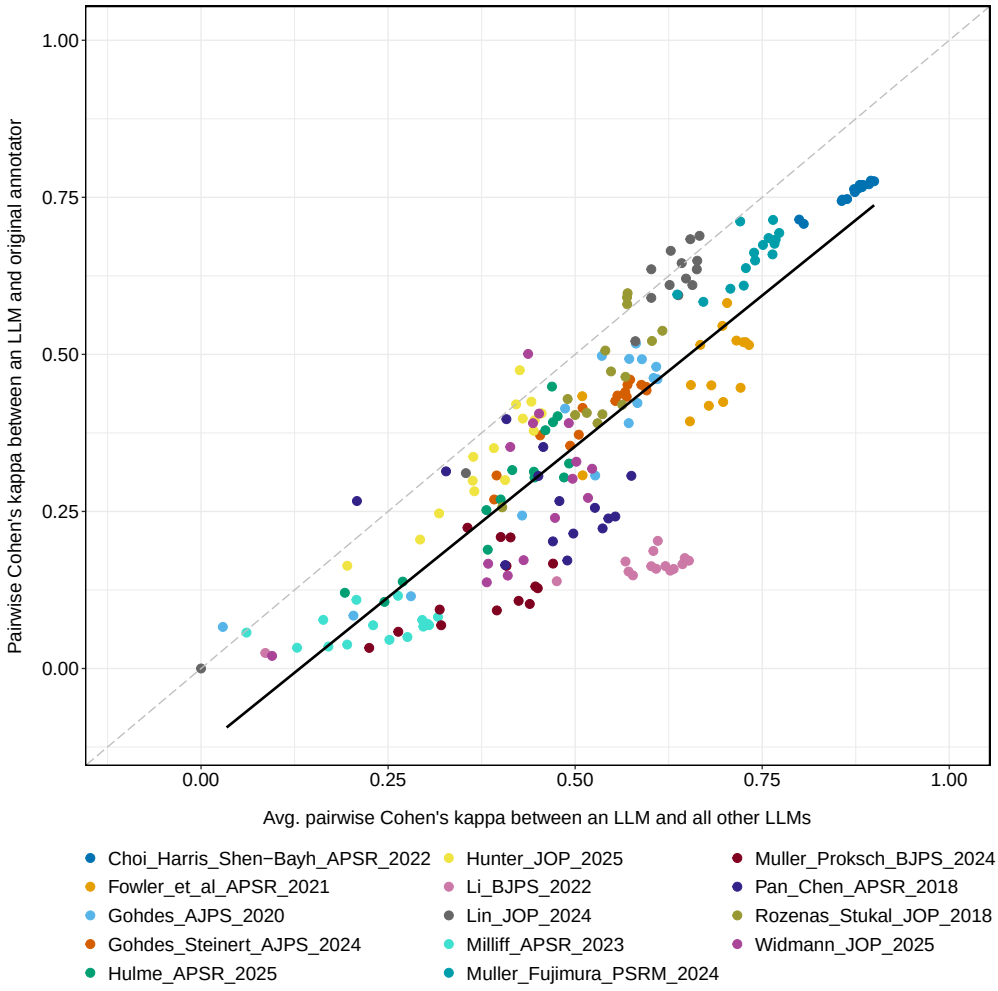
Here we present the pairwise intercoder reliability, correlation, in-context learning, and prompt format results using Cohen's kappa as the measure. Results are similar to those using Krippendorf's alpha.

FIGURE A10. HEATMAP OF PAIRWISE INTERCODER RELIABILITY (0-SHOT)



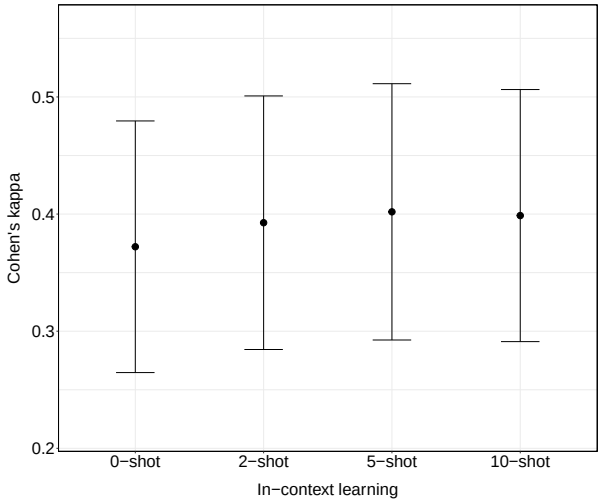
Notes: The figure presents Cohen's kappa for all pairs of annotators, averaged across the 14 studies. The numbers in brackets indicate the minimum and maximum alphas for each pair. "Original" indicates the original annotator(s) of the 14 studies. Reasoning models are suffixed with an asterisk (*).

Figure A11. Scatter plot of LLM-LLM and LLM-original intercoder reliability

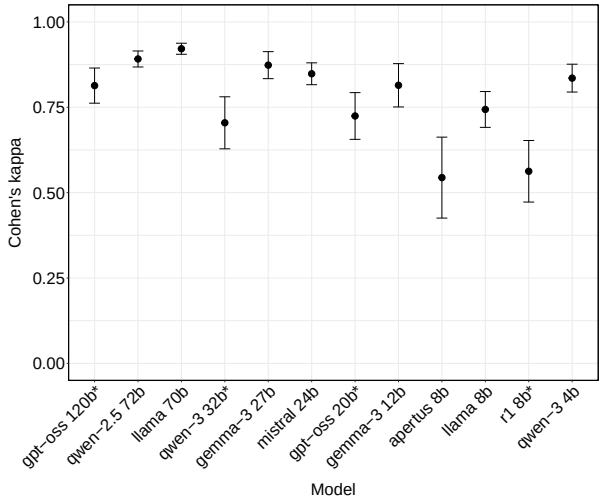


Notes: The figure presents a scatter plot of LLM-LLM and LLM-original annotator intercoder reliability. "LLM-LLM" is defined as the intercoder reliability between an LLM and all other LLMs for a given study. Each dot represents an LLM-study pair. Dots on the 45-degree dotted line indicate equal LLM-LLM and LLM-original annotator intercoder reliability. Dots below the 45-degree line indicate that the LLM-original annotator intercoder reliability is lower than that for LLM-LLM. A best-fit line (black) is plotted to facilitate interpretation.

Figure A12. Effects of in-context learning and prompt format on intercoder reliability (Cohen's kappa)



(a) Avg. intercoder reliability by in-context learning

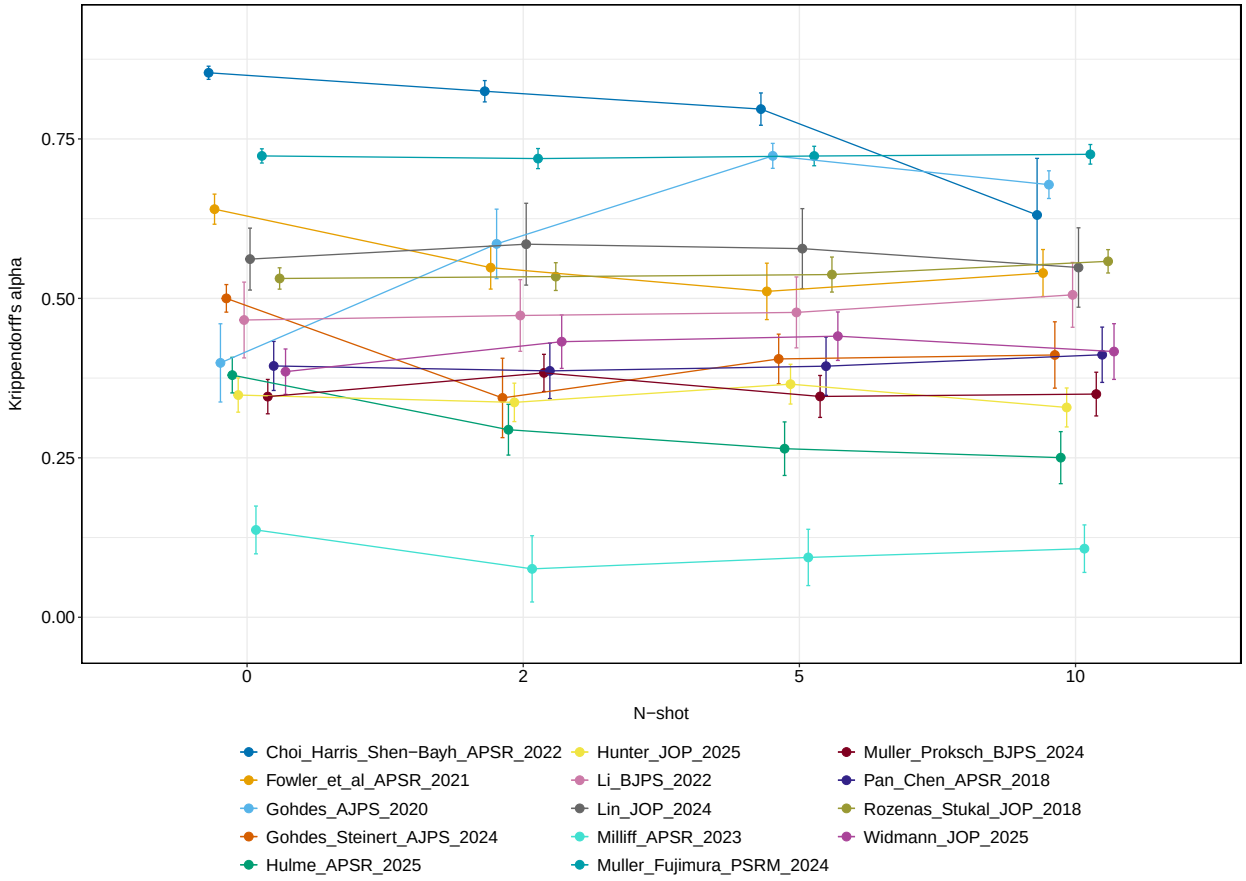


(b) Intercoder reliability between main and alternative prompt

Notes: Panel (a) shows the mean Cohen's kappa for 0-, 2-, 5-, and 10-shot learning. Panel (b) shows the intercoder reliability between the main and alternative prompts for a given LLM. Cluster-bootstrapped standard errors are used in both panels.

H. In-context learning result by study

FIGURE A13. AVERAGE INTERCODER RELIABILITY BY STUDY



Notes: The figure shows for each study the mean Krippendorff's alpha for 0-, 2-, 5-, and 10-shot learning.

I. Correlation between intercoder reliability and estimate variability

Table A5 reports the correlation between intercoder reliability and estimate variability for the main and alternative prompt designs. Estimate variability is defined as the standard deviation of the LLM-derived estimates for each coefficient. The negative correlation shows that a higher Krippendorf's alpha is correlated with lower estimate variability.

Table A5. Correlation between intercoder reliability and estimate variability

	Estimate variability	
	Main prompt	Alt. prompt
(Intercept)	4.721***	5.085**
	(1.013)	(1.534)
Krippendorf's alpha	−7.070***	−6.973*
	(1.945)	(3.040)
Num.Obs.	63	63
R2	0.178	0.079
R2 Adj.	0.165	0.064
Std.Errors	IID	IID

J. Notes for DSL and PRISA

Table A6. Notes for PRISA and DSL

Study	Variable type	Aggregation	Function	PRISA	Notes	DSL	Notes
Choi, Harris & Shen-Bayh (2022)	DV	✗	felm()	✓		✓	From no fixed effects to twoways (use lm() and felm() instead). However, for M5, M6, error: LU factorization of .gCMatrix failed: out of memory or near-singular.
Fowler et al. (2021)	DV	✓	felm()	✓		✓	
Gohdes (2020)	DV	✓	glm(), cbind(x,y)	✓	Clustered SEs are calculated post-estimation, rather than being computed within the estimation function. Thus not included in prisa() calculations.	✗	Model not supported by DSL.
Gohdes & Steinert-Threlkeld (2025)	DV	✗	glm()	✓	Clustered SEs are calculated post-estimation, rather than being computed within the estimation function. Thus not included in prisa() calculations.	✓	
Hulme (2025)	IV	✓	lm()	✗	Cannot process when the annotated variable is the independent variable.	✓	Filtered out observations with zero aggregated sampling probability.
Hunter (2025)	DV	✗	glm()	✓		✓	

Continued on next page

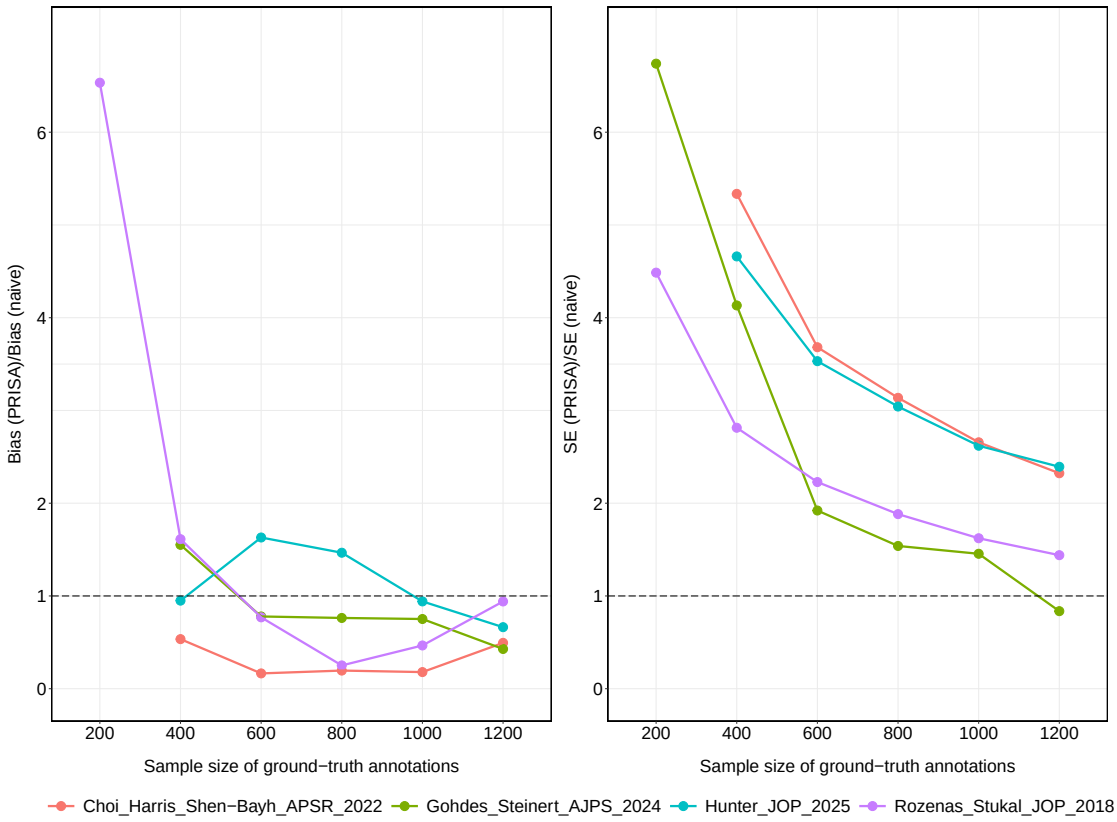
Study	Variable type	Aggregation	Function	PRISA	Notes	DSL	Notes
Li (2023)	DV	✓	plm()	✓		✗	Cannot process due to too many interaction terms.
Lin (2025)	DV	✓	feols()	✓		✗	Annotated variable is used in an interaction.
Milliff (2024)	IV	✓	zelig() mlogit.bayes	✗	Cannot process when the annotated variable is the independent variable.	✗	Model not supported by DSL.
Müller & Fujimura (2025)	IV	✓	feglm()	✗	Cannot process when the annotated variable is the independent variable.	✗	Model not supported by DSL.
Müller & Proksch (2024)	DV	✓	lmer()	✓		✗	Model not supported by DSL.
Pan & Chen (2018)	IV	✗	glm()	✗	Cannot process when the annotated variable is the independent variable.	✓	
Rozenas & Stukal (2019)	DV	✗	feols()	✓		✗	More than twoways. In DSL, felm() only supports oneway/ twoways
Widmann (2025)	DV	✓	plm()	✓		✓	Use felm() twoways instead.

K. PRISA result

Figure A14 presents the results for PRISA, which are consistent with our findings for DSL, highlighting a similar trade-off between bias reduction and increased variance. The analysis is limited to four of the ten PRISA-compatible studies. The remaining six studies are excluded because they require data aggregation (e.g., aggregating from sentence-level annotations to speaker-level for analysis). Since PRISA does not currently support such sampling designs, we cannot precisely control the ground-truth sample size at the unit of

analysis for these studies.

FIGURE A14. BIAS AND STANDARD ERROR COMPARISON: PRISA VS. NAIVE ESTIMATES



Notes: The figure presents bias and standard error comparisons between PRISA and naive estimates. The y-axis shows the ratio of the PRISA estimate's bias (left panel) or standard error (SE) (right panel) to that of the naive estimate from predicted annotations. Ratios below the dotted line (y=1) indicate that the PRISA estimates have smaller bias or standard errors, respectively. Each colored line represents a different study, plotted against the sample size of the ground-truth annotations used for correction. Some ratios for Choi, Harris and Shen-Bayh (2022) and Hunter (2025) are excluded because of estimation errors.