Humanities & Social Sciences Communications

ARTICLE

https://doi.org/10.1057/s41599-025-05834-4 OPEN

LLMs as annotators: the effect of party cues on labelling decisions by large language models

Sebastián Vallejo Vera[1]✉ & Hunter Driggers[1]

Human coders can be biased. We test whether Large Language Models (LLMs) replicate those biases when used as text annotators. By replicating an experiment conducted by Ennser-Jedenastik and Meyer (2018), we find evidence that LLMs use political information, and specifically party cues, to evaluate political statements. Not only do LLMs use relevant information to contextualize whether a statement is positive, negative, or neutral based on the party cue, they also reflect the biases of the human-generated data upon which they have been trained. We also find that unlike humans, who are only biased when faced with statements from extreme parties, some LLMs exhibit significant bias even when prompted with statements from center-left and center-right parties. The implications of our findings are discussed in the conclusion.

[1] University of Western Ontario, London, ON, Canada. ✉email: sebastian.vallejo@uwo.ca

Introduction

The increasing sophistication of large language models (LLMs) has allowed for their more prominent presence in political science research. One particular area that has received significant attention in the field is the use of LLMs as annotators. Research has shown promising results, with LLMs often outperforming human coders (Gilardi et al. 2023) and providing comparable accuracy when labeling political text (Overos et al. 2024), across multiple languages (Heseltine and Clemm von Hohenberg, 2024). While researchers have evaluated the performance of LLMs as annotators across various domains, there is still insufficient information on how the known biases of LLMs (see Gallegos et al. 2024) can affect their performance.

For human text annotators, studies show that political cues, such as party, affect their coding decisions (Benoit et al. 2016; Ennser-Jedenastik and Meyer, 2018; Laver and Garry, 2000). In this article, we test whether this is also true for LLMs. We replicate the experimental design of Ennser-Jedenastik and Meyer (2018), who tested human coder bias by having annotators evaluate policy statements on immigration that were modified by party cues. We use the same treatment (i.e., the same policy statement with different party cues) to evaluate how two LLM families, OpenAI's ChatGPT and Meta's LLaMa, determine the sentiments behind policy statements (i.e., positive, negative, or neutral). Our results show important differences in internal consistency across LLM models, low agreement between LLM and human coders and, most importantly, significant discrepancies that are conditioned by party cue (i.e., treatment). This behavior remains even when we explicitly tell LLMs to not take party labels into consideration. These results are reflective of the way LLMs embed certain tokens (e.g., party names) with information, as well as how embedding interacts with highly polarized contexts (e.g., immigration) where there are clear expectations about the position of each party. The output mirrors human behavior when evaluating political statements. However, our analysis suggests that LLMs are more sensitive to partisan context than human coders. In our conclusion, we discuss the implications and limitations of our results with respect to the use of LLMs as political text annotators.

Bias in annotation

Numerous political behavior studies have shown that individuals' perceptions can be biased based on characteristics that are relevant to political preference formation: gender, education, racial identity, and partisanship. Since annotators are not immune to their political contexts, similar biases have been observed in coding tasks. In a wide-ranging meta-analysis, Webb Williams et al. (2023) show various sources of annotator bias, including partisanship and gender, when completing subjective and objective coding tasks.[1] Other research finds that partisanship can affect reactions of disgust (Ahn et al. 2014) and responses on opinion surveys (Bullock and Lenz, 2019; Schaffner and Luks, 2018). In the study that we replicate in this paper, Ennser-Jedenastik and Meyer (2018) show that human coders use heuristics from party labels when judging political statements.

Research on LLMs has also explored biases and incongruities in their responses. Despite promises and fanfare from LLM developers, studies have shown multiple sources of errors (Hicks et al. 2024), as well as political bias in their output (Rotaru et al. 2024; Urman and Makhortykh, 2025; Walker and Timoneda, 2024). Motoki et al. (2024), for example, find that LLMs tend to align more with left-of-center viewpoints, a result similar to the one obtained by Rozado (2024) when probing LLMs with political orientation tests. More broadly, studies on LLMs have consistently found biases based on contextual and cultural factors (Gallegos et al. 2024), leading to the misrepresentation of certain social groups (Yang et al. 2022), gender stereotyping (Dong et al. 2024), and the reinforcement of normativity (Bender et al. 2021).

Despite the known limitations of LLMs, most research on LLMs as annotators has focused solely on accuracy, comparing their performance with that of human annotators. Relevant political science research has shown that LLMs outperform human annotators, at a fraction of the price, with minimal effects on downstream performance (see Braylan et al. 2022; Gilardi et al. 2023; Heseltine and Clemm von Hohenberg, 2024; Overos et al. 2024). Although the results of these studies provide promising avenues for the application of LLM as coders, less attention has been paid to the effect of known LLM biases on performance.

In this article, we test the possible biases in LLM annotation resulting from political cues. We argue that LLMs use political contextual information to evaluate statements and produce responses.[2] As a simplified explanation, Transformer-based LLMs assign representations (i.e., embeddings) to tokens by relating their co-occurrences with other tokens in the text. To do this on a large scale, LLMs are fed enormous amounts of text from various sources.[3] These corpora are not created in a vacuum, instead reflecting social realities. Thus, if certain parties (e.g., far-right parties) are often mentioned in certain contexts (e.g., discriminatory practices), LLMs are more likely to associate these parties with those contexts. When LLMs annotate text, they will use high-information tokens (i.e., political parties) to guide their responses, just as humans use party cues as a heuristic to evaluate statements (Ennser-Jedenastik and Meyer, 2018). In the remainder of the paper, we first analyze differences in output across LLM models and consistency in output within each model across multiple iterations of the same prompt; then, we evaluate to what degree party cues affect the output of LLMs when used as annotators.

Replication setup

To test possible biases in LLMs as annotators, we adapt Ennser-Jedenastik and Meyer (2018)'s experiment on party cues and human annotators. In their study, Ennser-Jedenastik and Meyer (2018) enlisted ten coders to classify 200 policy statements on immigration and migrant integration from Austrian election manifestos produced between 1986 and 2013. The authors removed all party labels, references to previous or subsequent sentences, and gender-sensitive language. They then randomly assigned a party cue (i.e, the treatment) to each statement: Green Party (Grüne - Extreme Left), Social Democrats (SPÖ - Center Left), People's Party (ÖVP - Center Right), or Freedom Party (FPÖ - Extreme Right). The control statement was the version without a party cue. An example of a statement without a party cue is "We stand for a modern and objective immigration policy," while a randomly treated statement would be "We [Greens/Social Democrats/Christian Democrats/Freedomites ('Freiheitliche')] stand for a modern and objective immigration policy." Each coder received 200 statements with randomly assigned conditions and were subsequently asked whether the statement conveyed a positive or negative stance on immigration, or if the statement was neutral or unclear. The coding question that appeared below each statement was "Does this statement convey a positive or a negative stance on immigration and multiculturalism–or is the statement neutral or unclear, respectively?". Coders were also provided with examples for negative and positive messages towards immigration and multiculturalism: "By positive statements on immigration and cultural diversity we mean, for example, approval of immigration to Austria and the acceptance of refugees, appreciation of other cultures in Austria or demands

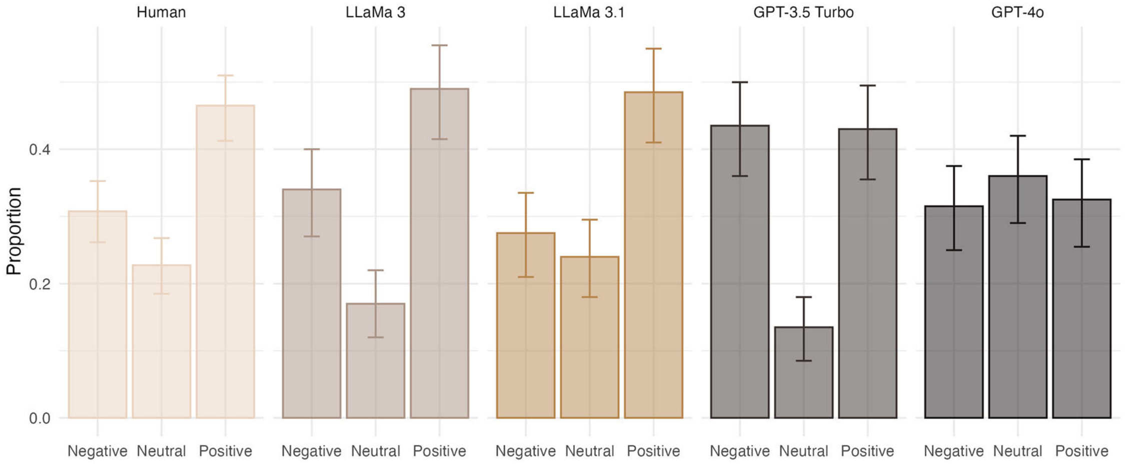

Fig. 1 Distribution of labels by coder with bias-corrected bootstrapped 95% confidence intervals (5000 samples) for statements containing *no party cues*.

for the opening of the labor market to foreign workers"; and "By negative statements on immigration and cultural diversity we mean, for example, demands for less immigration, skepticism towards other cultures and religions (Islam), the desire for preferential treatment of Austrians in the labor market or for restrictive asylum laws". Finally, Ennser-Jedenastik and Meyer (2018) tested whether the party cue had an effect on annotation. They found that human coders were more likely to evaluate statements with the Greens party cue (i.e., extreme left) positively, and more likely to evaluate statements with the FPÖ party cue (i.e., extreme right) negatively, with no effect from statements with centrist-party cues (e.g., SPÖ and ÖVP). We reproduce the results of their experiment in Table 2.

For our analysis, we use the exact same statements and instructions that were given to human annotators.[4] We test two LLM families: OpenAI's ChatGPT (ChatGPT-3.5 Turbo and ChatGPT-4o), and Meta's LLaMa (LLaMa 3-70B and LLaMa 3.1-70B).[5] Each LLM is prompted using the same text that an annotator would have read in Ennser-Jedenastik and Meyer (2018)'s study (including the examples of positive and negative statements). Each prompt is run in a fresh instance of the LLM client, making each response independent from the previous one. Given the independence of each instance, rather than randomly assigning one party cue to each statement (i.e., prompt), we show each LLM all 200 statements with each party cue, including the control. In total, each LLM labels 1000 statements (200 statements x 5 party cues). Since the output from LLMs is stochastic, we increase the determinism of answers by setting the temperature to 0 for all runs.[6] Additionally, we replicate each run (200 statements) 10 times, allowing us to measure within model consistency.[7]

To evaluate coding bias, we follow Ennser-Jedenastik and Meyer (2018) and use an ordered logistic regression to estimate the effect of the treatment on the labeling decisions of the LLMs. To that end, we estimate the following model:

$$y_{ijk} = cue_j + content_i + \epsilon_{ijk}$$

where y_{ijk} is the response of LLM k to statement i with party cue treatment j. LLMs categorize each statement as negative, neutral/unclear, or positive. Our main variable of interest is cue_j, an indicator variable that takes the value of the four party labels (i.e., Greens, SPÖ, ÖVP, and FPÖ), with the *no label* statements as the reference category. Following Ennser-Jedenastik and Meyer (2018), we control for content-related factors using either fixed effects at the sentence level or random intercepts at the statement level.

Inter-LLM and inter-coder reliability

We begin with Fig. 1, which plots the distribution of positive, negative, and neutral labels for statements containing *no party cues* (i.e., control group) of LLMs and human coders. Following a similar distribution as human coders, LLaMa 3 and LLaMa 3.1 models are more likely to label statements as positive. The same is not true for OpenAI models: ChatGPT-3.5 and ChatGPT-4o predict positive and negative labels at similar rates. The former is least likely to predict neutral labels, while the latter is most likely to do so. To further evaluate some of these discrepancies, Table 1 compares statements at different degrees of agreement. Statement where there is high agreement require no additional information or context to determine their valance. For example, the statement 'We want to create incentives to encourage people with a migration background to participate in teacher training courses' has a clear positive sentiment towards immigrants and their integration in Austrian society. There seems to be greater disagreement between LLMs and human coders in statements where context about the Austrian immigration system is required to make a decision. Take, for example, the following statement: 'We believe it is sensible to prepare an annual needs assessment regarding the number and qualifications of potential immigrants and use this as a basis for migration', labeled as neutral by LLMs and negative by humans. Knowledge of the anti-immigration rhetoric in Austrian politics might allow coders to pick up on certain telling elements: deciding on the 'type of immigrants' and 'how many' have been issues of anti-immigration policy (Kolbe, 2021). In the Discussion Section, we show that statements labeled as *neutral* when there is *no party cues* are more likely to change their label when prompted with the same statement that includes a party cue (see Figure F.1 in Appendix F), suggesting that, similar to humans, LLMs use heuristics from party labels when judging political statements. They do so, however, at a higher rate than humans.

We now turn to the task of describing inter-LLM and inter-coder reliability. We describe within-model consistency by looking at inter-run reliability. To this end, we estimate Krippendorff's Alpha, an inter-coder reliability (ICR) score, for each model across the multiple runs (ten in total). Within model Krippendorff's Alpha scores by party cue (treatment) are shown in Fig. 2. Overall, within model consistency is relatively high, with most scores close to or above.8.[8] There are important differences across LLM families: LLaMa models are highly consistent across runs, with an average Alpha close to 1.0, while OpenAI models are less consistent. Within models, however, there are no statistically significant differences resulting from party cues.

Table 1 Labels for statements about immigration and multiculturalism.

Agreement	Statement	Label(s)
Complete Agreement	'Only a clearly regulated asylum procedure can prevent illegal migration and subsequent asylum applications from leading to immigration through the back door.'	Negative
Complete Agreement	'A distinction must be made between the fundamental right to asylum in cases of persecution (asylum entitlements) and regular immigration (labor migration and family reunification).'	Neutral
Complete Agreement	'We want to create incentives to encourage people with a migration background to participate in teacher training courses.'	Positive
High LLM-LLM Agreement/Low LLM-Human Agreement	'The asylum 'fast-track procedure' must be expanded.'	Negative/Neutral
High LLM-LLM Agreement/Low LLM-Human Agreement	'We believe it is sensible to prepare an annual needs assessment regarding the number and qualifications of potential immigrants and use this as a basis for migration.'	Neutral/Negative
High LLM-LLM Agreement/Low LLM-Human Agreement	'We welcome motivated individuals who are willing to contribute their skills to Austria and who appreciate our culture and way of life.'	Positive/Neutral
Low LLM-LLM Agreement/High LLM-Human Agreement	'The willingness of immigrants to integrate is an indispensable prerequisite for their permanent residence in Austria.'	Neutral /Negative
Low LLM-LLM Agreement/High LLM-Human Agreement	'We want to develop a model that will allow us to actively manage the influx of people that Austria needs to secure its economic development.'	Neutral/Positive
Low LLM-LLM Agreement/High LLM-Human Agreement	'We do not see asylum policy as part of immigration policy, but as human rights policy.'	Neutral/Positive

The statements contain *no party cues*.

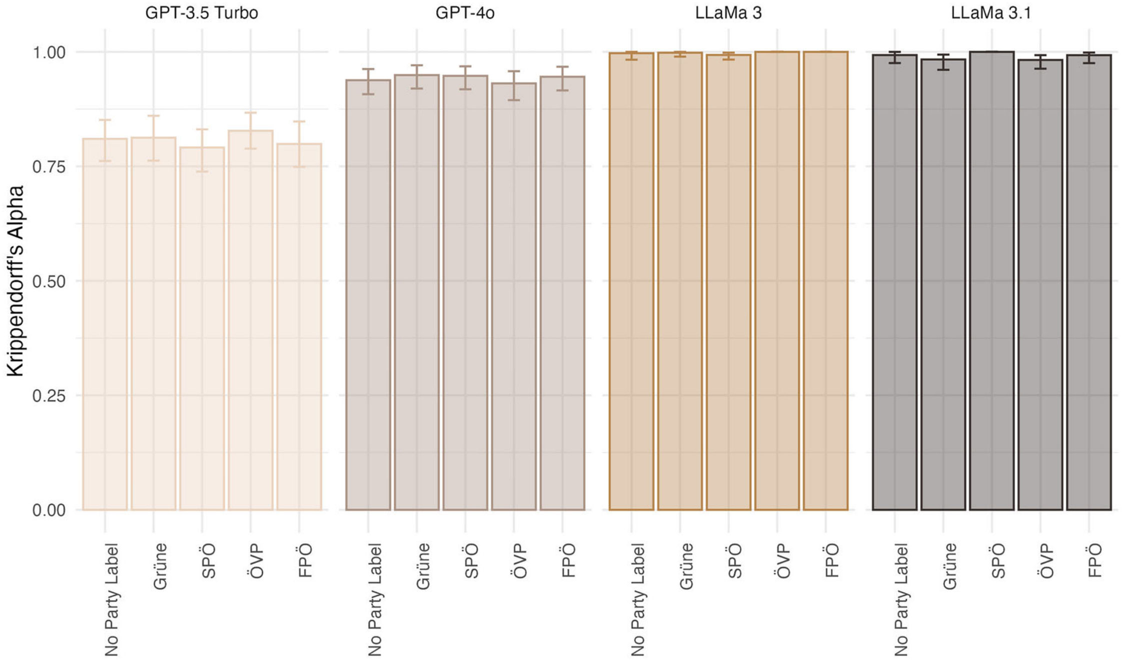

Fig. 2 Within-LLM Krippendorff's Alpha with bias-corrected bootstrapped 95% confidence intervals (5000 samples).

Finally, Fig. 3 depicts ICR scores between LLMs and the human annotators from Ennser-Jedenastik and Meyer (2018). Since every LLM labels each statement ten times, we adjudicate discrepancies following majority rules.[9] In terms of ICR, the best performing LLMs are LLaMa 3.1 and ChatGPT-4o, with an average Krippendorff's Alpha of 0.56 and 0.55, respectively; the worst performing LLM is ChatGPT-3.5 Turbo with a Krippendorff's Alpha of 0.43.[10] There are also differences in the variation of agreement between humans and LLMs across party cues. For the LLaMa family models, the agreement with human coders across party cues is similar, and we find no statistically significant differences. That is not the case for ChatGPT-4o, where there is a Krippendorff's Alpha of 0.68 when labeling statements with the center-left SPÖ label, but 0.37 when labeling statements with the extreme-right FPÖ label, a statistically significant difference at the 95% confidence level. In Figure D.1 (see Appendix D), we also show the estimates of Krippendorff's Alpha for each run from every LLM when compared to human coders. As expected, given the lower within-model consistency of ChatGPT models, there is greater variation in ICR scores across runs when compared to models from the LLaMa family. Note, however, that there are no overall performance gains from using adjudicated labels, nor labels from a specific run.

Replication results

In Models 1 and 2 from Table 2, we reproduce the results from Ennser-Jedenastik and Meyer (2018), who estimate an ordered logistic regression model where the dependent variable is the label

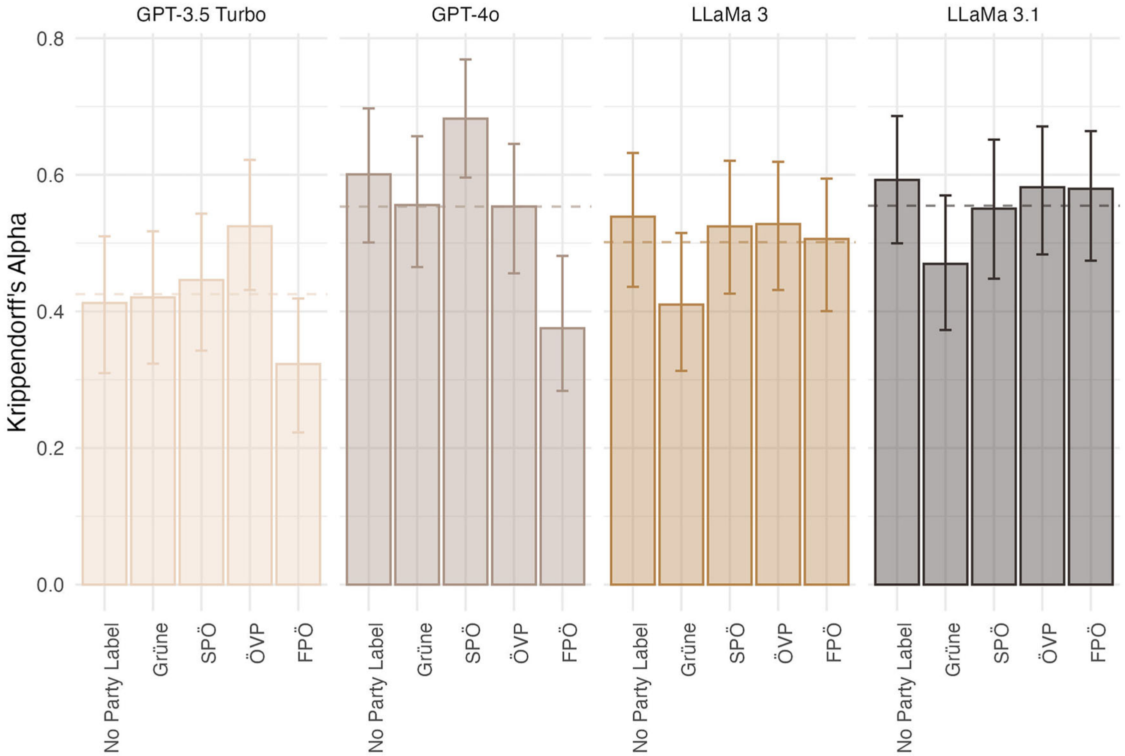

Fig. 3 Krippendorff's Alpha with bias-corrected bootstrapped 95% confidence intervals (5000 samples) between LLMs and human annotators. Within-model discrepancies across runs are adjudicated using majority rules. Model mean in dashed line.

Table 2 The Impact of Party Cues on Labeling Decisions: Results from Ennser-Jedenastik and Meyer (2018) and LLMs.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
	Original	Original	LLMs	LLMs	LLMs	LLMs
Grüne	0.96***	1.06***	0.56***	2.37***	0.56***	2.19***
	(0.20)	(0.20)	(0.10)	(0.18)	(0.10)	(0.17)
SPÖ	−0.01	−0.00	0.27**	1.10***	0.27**	1.03***
	(0.20)	(0.20)	(0.09)	(0.16)	(0.09)	(0.15)
ÖVP	0.01	−0.09	−0.09	−0.40**	−0.09	−0.37*
	(0.20)	(0.20)	(0.09)	(0.15)	(0.09)	(0.15)
FPÖ	−0.75***	−0.80***	−0.72***	−2.93***	−0.72***	−2.77***
	(0.20)	(0.20)	(0.09)	(0.17)	(0.09)	(0.17)
ChatGPT-4o			−0.06	−0.14		
			(0.08)	(0.14)		
LLaMa 3			0.28**	1.17***		
			(0.09)	(0.14)		
LLaMa 3.1			0.33***	1.52***		
			(0.08)	(0.15)		
Cut 1: Constant	−1.19	−3.15***	−0.60***	0.94	−0.73***	−2.39***
	(0.65)	(0.43)	(0.09)	(0.55)	(0.11)	(0.55)
Cut 2: Constant	2.02**	0.07	0.46***	5.39***	0.33**	1.78**
	(0.65)	(0.42)	(0.09)	(0.57)	(0.11)	(0.55)
Num.Obs.	2000	2000	4000	4000	4000	4000
Statement FE	Yes	No	No	Yes	No	No
Statement RE	No	Yes	No	No	No	Yes
Coder/LLM FE	Yes	Yes	No	No	Yes	Yes

Figures are coefficients from ordered logistic regression, with standard errors in parentheses; statement fixed-effects for all models not shown. Confidence levels reported as follows: $^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$. Grüne Green Party, SPÖ Social Democrats, ÖVP People's Party, FPÖ Freedom Party.

assigned by the human coder (i.e., positive, neutral/unclear, or negative), and the independent variable of interest is the party cue indicator (using the statement with no party label as the reference category).[11] In their main findings, Ennser-Jedenastik and Meyer (2018) show that party cues have an effect on coding decisions, but only when the treatment is a party on the ideological extreme (i.e., Greens and FPÖ). They find that human coders judge statements on immigration from the Green party more positively, while statements with the FPÖ label are more likely to be labeled negatively (see Models 1 and 2 of Table 2).

In Models 3 through 6 of Table 2, we estimate a similar ordered logistic regression model where the dependent variable is the label provided by the LLM,[12] and the independent variable of interest is the party cue indicator (i.e., treatment). Models 3 and 5 include all aggregated data and control for each LLM, while Models 4 and 6 also include statement fixed-effects and random intercepts at the statement level, respectively. The positive coefficients from the Green party and SPÖ cues (far-left and center-left) suggest that statements with those cues were judged as more positive by LLMs than statements without party cues. Similarly, LLMs evaluated statements with the ÖVP and FPÖ cues (center-right and far-right) more negatively than statements without party cues. As a robustness check, in Appendix C we estimate every model only using data of each LLM separately, and our results are consistent regardless of model specification.

The differences in labeling decisions are substantively (as well as statistically) significant. When compared to the neutral statements, the Green party (far-left) cue increases the probability of a statement being coded as 'Positive' by 13.9 percentage points, while it decreases the probability that LLMs use the 'Neutral' label by 3.0 percentage points and the 'Negative' label by 10.9 percentage points.[13] We observe a similar, yet less pronounced, effect when applying the SPÖ (center-left) cue: LLMs are 6.7 percentage points more likely to label a statement as 'Positive', and 1.0 and 5.6 percentage points less likely to label a statement as 'Neutral' and 'Negative', respectively. The opposite effect is observed from the FPÖ (far-right) cue. For example, LLMs are more 17.2 percentage points more likely to identify statements with the FPÖ label as negative and 15.9 percentage points less likely to code statements with the same label as 'Positive'. The results are robust to estimations using each LLM sub-sample.[14] The controls provide additional insight into the effect of party cues on labeling decisions. The results from Model 3 in Table 2 show that both LLaMa 3 and LLaMa 3.1 are more likely to label statements as 'Positive' than ChatGPT-3.5 Turbo ($p < 0.05$).

There are two important elements to note from our comparison of the effects of party cues on human and LLM coders. First, as Ennser-Jedenastik and Meyer (2018) suggest, coders use prior knowledge to contextualize information provided by text. For humans, prior knowledge comes from heuristic processing; for LLMs, from the context in which party labels appear in pre-training data (i.e., the massive corpora used to train LLMs). Both human and LLMs coders seem to understand contextual cues in the same way: the direction of bias for left-leaning parties and right-leaning parties matches prior expectations about their position vis-a-vis immigration (i.e., left-leaning parties are more likely to frame immigration in a positive light; right-leaning parties are more likely to frame immigration in a negative light). Second, for LLMs, party cues appear to have greater weight on the decision to label statements one way or another. For example, unlike with human coders, the center-left SPÖ and the center-right ÖVP party labels have a significant effect on labeling decisions. In Appendix D, we show that, when estimating the models using all the runs, the effect of cues is also statistically significant for all parties.

To test individual LLM differences in coding decisions, and to directly compare these decisions to the labels produced by human coders, we combine both data sets and interact our variable on interest (i.e., party cue) with each indicator variable for coder (including human coders). Table C.1 in Appendix C presents the complete model. For ease of interpretation, we plot the predicted probabilities for each coder in Fig. 4. Overall, the positive and statistically significant effect of the Green party, as well as the negative and statistically significant effect of the FPÖ cue ($p \leq 0.05$), on labeling decisions is present in all LLMs. Importantly, there are significant individual differences in terms of the magnitude of estimated effect, as well as differences in relation to human coders. While all coders, whether humans or machine, are more likely to code statements with the Green party cue more positively, both LLaMa models do so at a higher rate than humans and OpenAI models. FPÖ cues negatively bias LLMs' labeling decisions more than humans', yet ChatGPT-3.5 and ChatGPT-4o models are 26.8 and 24.9 percentage points, respectively, more likely to do so than humans. Similar variation is observed when looking at the more centrist parties. All LLMs are more likely to label statements carrying the SPÖ (center-left) cue ($p \leq 0.05$) positively, but because OpenAI models start at a lower threshold (i.e., the No Party Label labeling tends to be less positive), they label these positive at a rate that is similar to humans. For the ÖVP (center-right), only ChatGPT-3.5 is not negatively biased when prompted with the cue.[15] LLaMa 3 and LLaMa 3.1 are the most biased models when cued with the ÖVP, even though ChatGPT-4o is the most likely to predict negative labels when prompted with ÖVP. When no party labels are present, OpenAI models have a more negative bias compared to both humans and LLaMa models ($p \leq 0.05$).

Understanding LLM biases

Though it is difficult to understand the processes that lead LLMs to be more susceptible to party cues than human coders, we estimate the impact of three alternative specifications on output in order to uncover possible explanations. We argue that LLMs use political contextual information to evaluate statements and produce responses, and that this contextual information is therefore responsible for biased output. To test the possible effect of information, we first look at the effect of LLMs using all their pre-training information on the label output. We then modify our original prompt by asking all LLMs to answer as (a) an 'average citizen', (b) a 'citizen with low information', and (c) by 'not taking into account the political party mentioned in the prompt'. In Appendix E, we show within-prompt consistency and find no major differences from the original prompt; the same with intercoder reliability when compared to human coders. To show the differences in labeling behavior, we aggregate all the responses from the LLMs, as well as the human responses, and interact the indicator variables for each prompt with the indicator variables for the party cues.[16] We plot the predicted probabilities from all LLMs and prompts in Fig. 5. We find similar biases when compared to the original prompt, both in magnitude and direction. Somewhat surprising, while there are no statistically significant differences between the original and the alternative prompts when labeling statements with no party cues, all the alternative prompts produce greater positive bias when labeling statements with the Green Party and SPÖ cues (left and center-left).

Asking any of the LLMs to answer as an 'average citizen' or a 'citizen with low information' seems to slightly increase the magnitude of the bias, suggesting that the baseline for LLMs is to use political contextual information to evaluate statements, and that it deems 'average' and 'low-information' respondents as relying more on these high-information words. Even more relevant for annotation tasks is that when explicitly prompting LLMs to not consider the previously known source of bias into consideration (i.e., party cues), the responses from LLMs are still influenced by the party cues.

If LLMs are using party cues to assess information, these assessments may be made in conjunction with the policy contexts in which they are embedded–much like human coders. Immigration is a salient and divisive political debate in Austria, with parties adopting divergent and well-defined positions. The degree of bias from LLMs may therefore vary across different policy areas or political contexts, depending on prior expectations about

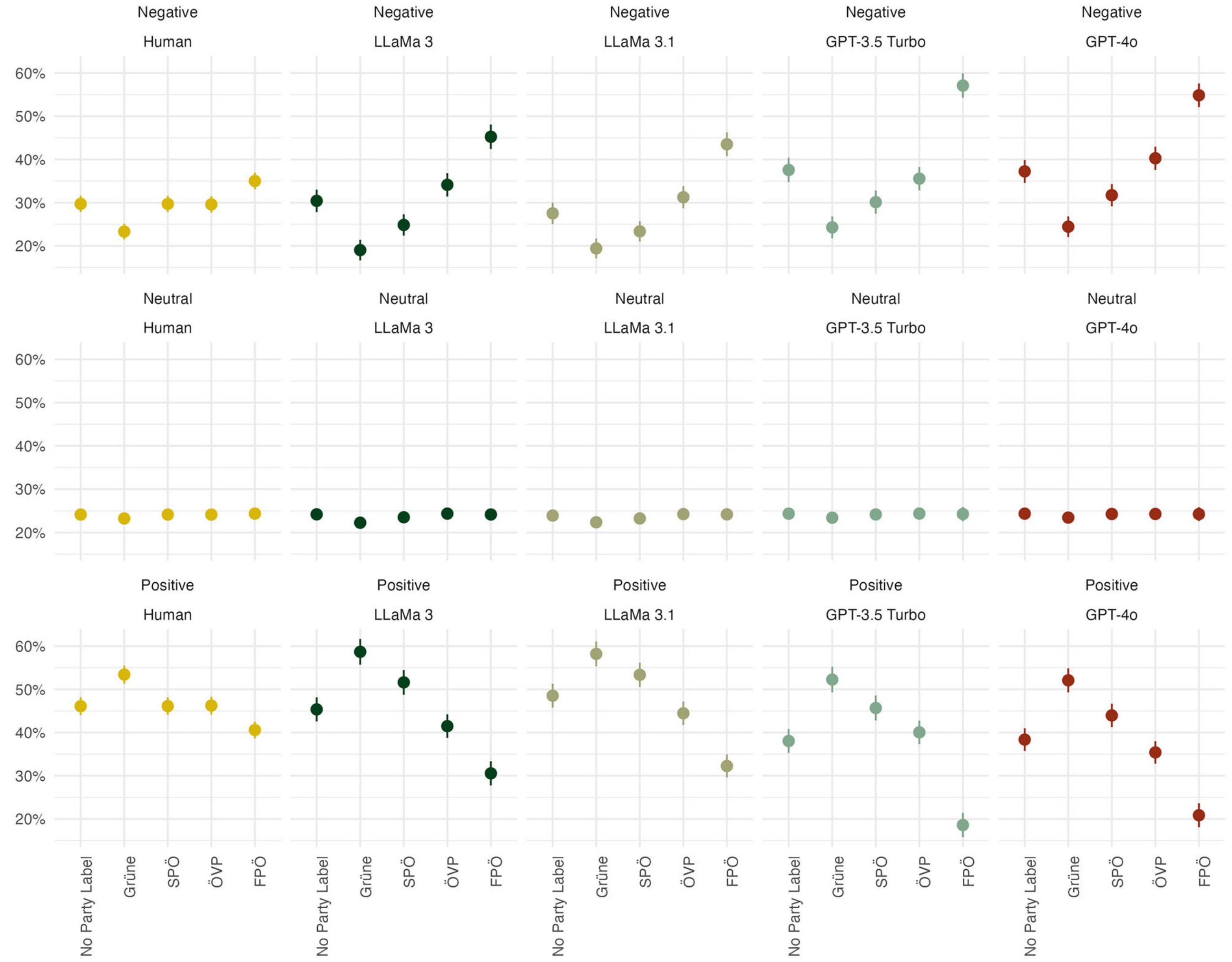

Fig. 4 Predicted probabilities for the effect of party cues on coding decisions by human coders and LLMs.

the political position of each party within these contexts. To further explore this intuition, we us an alternative dataset and run similar tests. The alternative sentiment data contains 12,329 labeled statements from Austrian parliamentary debates and party press releases between 1995-2013, covering various policy domains (Haselmayer and Jenny, 2017), from which we randomly sample 200 statements. Broadly, the statements address less politically divisive, and often less salient, issues.[17] Thus, for most statements, we have no apriori expectations about the policy stance of each party on these issues.

Following the same process as with the immigration dataset, we add party cues to each statement. In Appendix G, we plot the distribution of LLM labels and their correlation to human labels, and we find a similar behavior from most LLMs as with the immigration data.[18] We test the presence of bias from party cues by estimating the same models as we did with the Ennser-Jedenastik and Meyer (2018) data. Table G.2 in Appendix G presents the full results, and for ease of interpretation we plot the predicted probabilities in Figure G.2 in the same Appendix. Overall, party cues have no differentiating effect on labeling decisions, contrary to what we observe with the immigration data. We find no differences in the relative positiveness or negativeness across LLaMa models, all displaying a similar behavior across party prompts. ChatGPT models label more statements as neutral, and less positive, than LLaMa models. ChatGPT-3.5 is more likely to label a statement as negative (and less likely to label it as positive) in the presence of *any* party cue when compared to statement with no party labels, while ChatGPT-4o is more less likely to label a statement as negative in the presence of *any* party cue when compared to statement with no party labels ($p \leq 0.05$). The results suggest that LLMs embed party tokens with information that affects their behavior, and that LLMs are most reliant on this information when placed in a context where there are prior expectations about the position of the party. This is in line with research on the behavior of LLMs when addressing political topics, showing how LLMs react differently to highly-polarized and less polarized topics (see Pit et al. 2024; Vijay et al. 2025; Yang et al. 2024).

Finally, we look at the effect of different temperatures on model output. While a lower temperature makes model output more deterministic, it might also lock the available responses into one option, probability-wise. If this option places high attention on party cues, then there is an increased likelihood of bias. To address this concern, we run the original prompts with temperatures of 1 and 1.25.[19] In Appendix E, we plot within-prompt consistency and show that, as expected, there is greater variation in answers as we raise the temperature. Models 1 through 10 in Table E.2 (see Appendix E) present the results of the ordinal logistic model when using different temperatures, both in aggregate and for each LLM.[20] Overall, the results are mixed, even though we still find similar biases (in direction and statistical significance) resulting from changes to the party cue. In general, higher temperature decreases the likelihood that statements are evaluated positively. There are meaningful differences by LLM

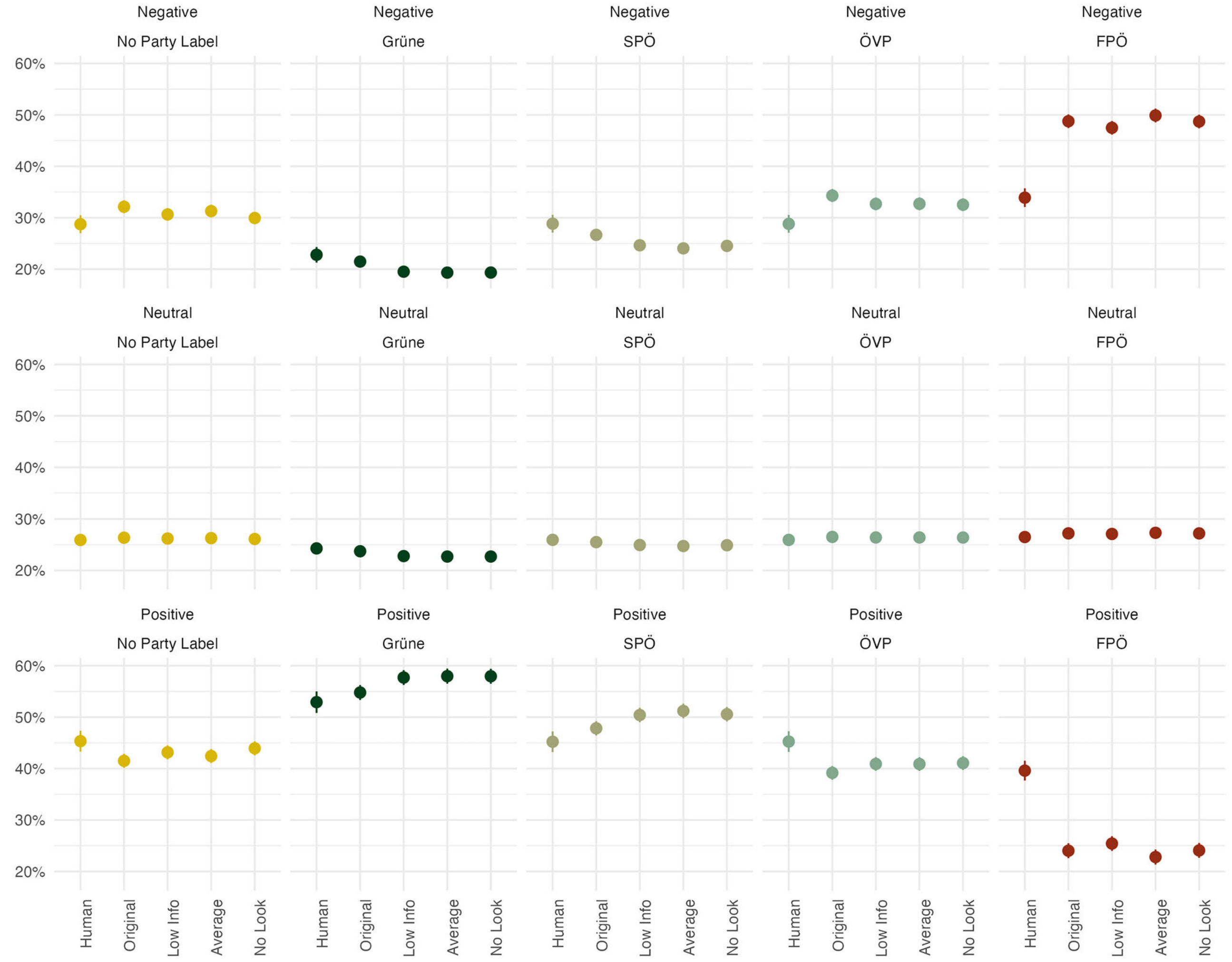

Fig. 5 Predicted probabilities for the effect of party cues on coding decisions across different prompts.

and party cue, however. For example, while the LLMs consistently evaluate statements with the Grüne Party cue more positively than statements without a party label, and having a temperature of 1 or 1.25 typically results in positive evaluations becoming less likely, in aggregate our results reflect that the negativity induced by increasing the temperature above 0 is effectively mitigated by the interaction between temperature and the Grüne Party label. For statements with the FPÖ label, which were already more likely to be seen by the LLMs as negative, temperature–and the interaction between temperature and the FPÖ party label–typically increases the likelihood of a negative evaluation. Even so, variation in results between LLMs means that our general findings about temperature and party label are not always consistent. For example, while ChatGPT-3.5 Turbo was estimated to produce a negative interaction effect when it was set to a temperature of 1 and given statements about the FPÖ, for LLaMa 3.1 this interaction effect was positive, as was the interaction effect observed from ChatGPT-3.5 Turbo when changing the temperature from 1 to 1.25. Variation in the influence of temperature–and the interactions between temperature and party cue–on our estimates is not suprising, given the much lower ICR scores of the LLMs at temperatures higher than 0. When looking at the results of temperature across all models, it seems that increasing the temperature above 0 is analogous to increasing the degree of randomness that appears in the data being generated.

Discussion and Conclusion

Our results show that, similar to human coders, contextual information affects responses from LLMs. In particular, we provide evidence that party cues affect the way that LLMs label policy statements: far-left and center-left party cues increase the probability an immigration related statement will be labeled as positive; far-right and center-right party cues increase the probability an immigration related statement will be labeled as negative. Unlike humans, this effect is not limited to parties at the ideological extremes. We also find that the magnitude of the effect is greater for LLMs than it is for human coders. Furthermore, LLMs are biased even when they are explicitly instructed to ignore party cues.

The context-based decision making observed from LLMs is not necessarily problematic. Informational cues can improve the validity of the data as long as the priors of LLMs are 'correct.' For example, consider the statement "We should take care of the situation with the immigrants." The sentiment of the statement, on its own, might be unclear; the sentiment of the message might have clear positive or negative connotations depending on whether the party is left- or right-leaning.[21] However, the impossibility of realistically perusing the data used to train LLMs, or of fully understanding the 'thought process' behind any given response, makes it unlikely for researchers to know what these priors are, and which tokens (or set of tokens) have these priors embedded.[22] With the constant updating of LLMs, as well as the proliferation of new LLMs, it is difficult to know whether priors will all be the same across platforms and time.

More importantly, our findings suggest that LLMs might process political information differently than humans, amplifying biases across the ideological spectrum rather than simply replicating human tendencies. Political scientists have long recognized that party cues shape public opinion, particularly on salient issues such as immigration. However, LLMs rely more on this partisan cues, and are more deterministic in their labeling decisions, than human coders. Additional analysis of the data suggests that LLaMa models, and OpenAI models to a lesser degree, are more likely than humans to change from labeling a statement as neutral when there are no party cues, to labeling a statement as positive or negatively when there is a Green Party or FPÖ cues, respectively ($p \le 0.05$).[23] When it comes to more centrist parties, LLMs are similarly deterministic in their behavior. For example, ChatGPT-4o changed 18% of its labels from neutral when there was no party cue to positive when there was an SPÖ cue, yet changed none of its labels to negative. In the same situation, human coders changed a similar percentage of labels from neutral to positive or to negative. This suggests that LLMs are filling in the information about sentiment when there are not many clues in the context with party cues (e.g., neutral statements). Additionally, OpenAI models are more likely than humans to change from labeling a statement as positive when there are no party cues, to labeling a statement as neutral when there is an FPÖ cue ($p \le 0.05$). OpenAI models might have embedded information on the party cue that they are leveraging against the contextual information at a higher rate than humans. Note that party cues also make LLMs lock into a labeling decision when, for example, the statement without party cue suggests a negative sentiment and the party cue is the FPÖ. In that situation, all models will label close to 100% of the time the statement as negative (while human coders label them as negative 80% of the time).[24]

If LLMs are applying political heuristics more rigidly than human coders, then partisan framing might play a bigger role even in situations where context provides less information about political position and, thus, sentiment. In the highly polarized topic of immigration in Austria, we find that LLMs are applying partisan frames. When testing the behavior of LLMs with more general political text, we find there to be no systematic differences across party cues. Thus, the behavior is not generalizable to all contexts, but the interaction between context and partisan framing could be relevant when using LLMs as annotators. Still, further research is needed in order to understand more broadly the effects of context on the behavior of LLMs. In this paper, we test LLaMa and OpenAI models in German text about Austrian politics. How these results hold in other contexts, languages, and LLMs still remains an open question.

Ennser-Jedenastik and Meyer (2018) argue that human coders use prior knowledge to contextualize information, what they call heuristic processing, and that this contextualization will have an effect on policy-related labeling decisions. We argue for a similar intuition when it comes to LLMs. LLMs are trained using troves of human-generated text data, and high valence tokens, such as party labels, are more likely to appear under specific contexts. Since the training data reflects societal biases, these biases will inevitably be reflected by LLMs. Given the obscurity of how LLMs are trained, it is challenging to assess the degree to which our proposed mechanism is actually affecting the decisions taken by LLMs. Yet, research has shown that LLMs decode text by focusing on high information tokens, a characteristic that might lead to an over-reliance of that information when assessing prompts. Future research can expand on the types of cues, and types of tasks, that affect the behavior of LLMs, and how to mitigate unwarranted bias.

Combined with other research on biases, researchers should take into account that LLMs are not created in a vacuum. They are the partial product of *human-generated* data, that is embedded with human biases. As previously noted, using partisan frames that provide greater context and aid in improving annotation tasks are not necessarily bad. The fact that LLMs are less moderate in their reliance of these cues than humans in annotation tasks is an additional element researchers should pay close attention to when validating their data. Thus, while there are many benefits to using LLMs as annotators, such as their low cost and high accuracy (Gilardi et al. 2023; Heseltine and Clemm von Hohenberg, 2024), these should be put in perspective of possible biases. This is of particular importance to social scientists who study highly polarized contexts (e.g., elections, gendered institutions, racial identity), where LLMs might overly rely on certain cues. Given the expansion of the use of LLMs as annotators, social scientists should leverage their own theoretical understanding of the topic to anticipate or, at the very least, check for possible biases when there are clear expectations about the influence of cues (e.g., partisan frames on immigration).[25] Overall, we recommend researchers use a similar approach as the ones shown here on high information terms that could influence the results in undesirable ways to identify systematic sources of bias.

Finally, we provide additional suggestions for researchers employing LLMs as annotators. Since LLMs are non-deterministic in their answers, we run multiple iterations of each annotation task. This is similar to other recent work using LLMs as annotators, as well as research using Transformer-based models more broadly (Timoneda and Vallejo Vera, 2025). We find that internal consistency (i.e., the consistency of results across multiple runs) varies across LLM families and LLMs, with LLaMa 3.1 models yielding the best results overall in terms of performances *and* consistency (while remaining open-source). We also find that improvement in models can also lead to greater consistency, as is the case for the ChatGPT family. This has important implications, not only for the use of LLMs as coders, but in the replicability of outputs from LLMs in research. Best practices when using LLMs as annotators should include multiple iterations of the task, reporting on all the output from LLMs, and clearly explaining the adjudication strategy employed. Using multiple iterations of any task performed by LLMs allows researchers to replicate the results over repeated samples; being explicit about the adjudication strategy and reporting on the different outcomes from the type of adjudication strategy used can provide robustness to empirical findings using the output of LLMs.

Additionally, LLMs are constantly changing and being updated, and adapted versions are built on top of LLMs (e.g., DeepSeek-R1-Distill-Llama-70B is a DeepSeek model based on LLama). Older models often become obsolete, and platforms stop supporting them. The LLM environment makes replication complicated and, with time, impossible. Researchers should clearly report the version of the model used and, if possible, the platform on which it was deployed (or if it was locally deployed), as well as the date when the code was ran. This allows readers to understand possible variations in output.[26] Finally, we have taken various steps to provide the transparency and reproducibility that are often expected from academic work. To this end, we have made our code and data publicly available,[27] and encourage other researchers do the same.

We close by noting that this is not a thorough examination of biases in LLMs as annotators, but rather a starting point. We are testing these biases for a specific case, in a limited set of LLMs, in a particular point in time. As previously suggested, there are important limitations to the generalizability of the results. This study serves as a probe into the biases of LLMs, providing some best practices from the conclusions, and encouraging researchers to be cognizant, if not vigilant, of how these can affect their work. Ultimately, our research cautions researchers who are considering using LLMs as coders, echoing previous research that has highlighted biases of human coders (Benoit et al. 2016; Ennser-Jedenastik and Meyer, 2018; Laver and Garry, 2000). Our final recommendation goes back to Grimmer and Stewart (2013) key principle of text analysis more generally: "validate, validate, validate."

Data availability

All replication data and code is available at https://github.com/svallejovera/llm_as_coders.

Code availability

Full modeling details and code are available online at https://github.com/svallejovera/llm_as_coders.

Received: 30 December 2024; Accepted: 22 August 2025;
Published online: 29 September 2025

Notes

[1] Webb Williams et al. (2023) also find that there is low generalizability of their results. For example, gender biases found in U.S. annotators were not found in Dutch annotators.

[2] This is similar to heuristic processing in human annotators when using political party cues to evaluate statements (Ennser-Jedenastik and Meyer, 2018).

[3] For a more detailed explanation on the encoding-decoding infrastructure of Transformer-based model, see Timoneda and Vallejo Vera (2024).

[4] See Appendix A for an example of the prompt given to the LLMs.

[5] We focus on four widely available models that are relatively easy to implement through cloud-based servers. Unlike OpenAI's ChatGPT models, Meta's LLaMa family is open-source and free to use, an important element when considering our model of choice (Palmer et al. 2024).

[6] A temperature of zero does not mean a completely deterministic answer. A low temperature value makes models more determined by the text in the training data, which increases the likelihood of the most common token(s) to be generated next; a higher temperature makes the model less reliant on the training data, flattening the curve of the probability of the token(s) to be generated next, making the output seem less deterministic. In either scenario, the possibility of variation remains.

[7] As we explain in our Discussion section, replication of tests using LLMs can be complicated, given the stochastic nature of the models and that models are frequently updating and changing. Using multiple iterations of any task performed by LLMs will allow researchers to replicate results over repeated samples.

[8] According to Krippendorff (2018), a Krippendorff's Alpha above .8 is a satisfactory level of agreement, allowing for triangulated inferences based on the labeled data.

[9] In Appendix B we show all results using alternative adjudication methods, but the overall conclusions from the analysis remain unchanged.

[10] Krippendorff's Alpha below .67 is a poor level of agreement, which does not allow for triangulated inferences from the labeled data. These results suggest that annotators are not applying the coding scheme consistently (Krippendorff, 2018).

[11] The results in Model 1 and 2 are virtually identical to the ones presented in Ennser-Jedenastik and Meyer (2018).

[12] For Table 2, using the labels where we adjudicate discrepancies following majority rules. In Appendices B and D, we estimate the same model using alternative adjudication strategies, as well as all runs individually, and find similar results.

[13] We estimate all predicted probabilities from Model 3, using LLM random-effects, as these yield the more conservative effects.

[14] In the Appendix, we estimate the same model using the data from each individual run, rather than the adjudicated data. The results and conclusions remain unchanged.

[15] See Appendix C for estimate for models with individual LLMs.

[16] In Appendix E, we present the complete model. We also estimate each model with the individual LLMs and find no difference in the results.

[17] In Table G.1 (see Appendix G), we provide a selection of statements from this alternative dataset.

[18] ChatGPT-4o produced too few positive labels to estimate similar ordinal logistic models. In Appendix G we show, separately, the results for only two labels–negative and neutral–.

[19] For LLaMa 3.1, the temperature ranges between 0 (most deterministic) to 2 (most liberal). However, for our prompt, setting the temperature above 1.25 returned gibberish.

[20] Note that, with greater variation in output, the adjudication criteria becomes more relevant.

[21] We thank a reviewer for pointing out this issue and providing an example.

[22] Recent literature has explored 'reasoning' in LLMs, looking at advances on understanding the learning patterns of LLMs (for a review,see Bandyopadhyay et al. 2025; Zhang et al. 2024). By analyzing the output of LLMs, and rewarding/punishing certain responses, researchers can have a partial view of the 'thought process' and, most importantly, change the behavior of LLMs. Future research should explore how fine-tuning LLMs–using training data to customize the behavior of LLMs to a specific task–can adjust the priors of LLMs to provide more accurate responses.

[23] See Figure F.1 in Appendix F.

[24] It is also the case that some LLMs are changing their behavior in different texts than humans. In Figure F.2 in Appendix F we present a heatmap of the statements for which coders changed their label in the presence of party cues. LLaMa-3 and LLaMa-3.1 models change their behavior in 76.5% and 79% of the same statements as humans, respectively. GPT-3.5 and GPT-4o models change their behavior in 59% of the same statements as humans. Thus, the content that leads to greater bias will be different across LLMs.

[25] We show that, absent these clear expectations, partisan cues have no effect on labeling decisions from LLMs.

[26] Knowing changes in LLMs, either in training or data (or both), is still dependent on transparency from companies developing these LLMs.

[27] This includes the raw data as provided by Ennser-Jedenastik and Meyer (2018), as well as the output data from the LLMs.

References

Ahn WY, Kishida KT, Gu X, Lohrenz T, Harvey A, Alford JR, Smith KB, Yaffe G, Hibbing JR, Dayan P, Montague PR (2014) Nonpolitical images evoke neural predictors of political ideology. Curr Biol 24(22):2693–2699

Bandyopadhyay D, Bhattacharjee S, Ekbal A. (2025) "Thinking Machines: A Survey of LLM based Reasoning Strategies." *arXiv preprint arXiv:2503.10814*

Bender EM, Gebru T, McMillan-Major A, Shmitchell S (2021) On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. pp. 610–623

Benoit K, Conway D, Lauderdale BE, Laver M, Mikhaylov S (2016) Crowd-sourced text analysis: Reproducible and agile production of political data. Am Political Sci Rev 110(2):278–295

Braylan A, Alonso O, Lease M (2022) Measuring annotator agreement generally across complex structured, multi-object, and free-text annotation tasks. In Proceedings of the ACM Web Conference 2022. pp. 1720–1730

Bullock JG, Lenz G (2019) Partisan bias in surveys. Annu Rev Political Sci 22(1):325–342

Dong X, Wang Y, Yu PS, Caverlee J (2024) "Disclosure and mitigation of gender bias in llms." *arXiv preprint arXiv:2402.11190*

Ennser-Jedenastik L, Meyer TM (2018) The impact of party cues on manual coding of political texts. Political Sci Res Methods 6(3):625–633

Gallegos IO, Rossi RA, Barrow J, Tanjim MM, Kim S, Dernoncourt F, Yu T, Zhang R, Ahmed NK (2024) "Bias and fairness in large language models: A survey." Comput Linguist pp. 1–79

Gilardi F, Alizadeh M, Kubli M (2023) ChatGPT outperforms crowd workers for text-annotation tasks. Proc Natl Acad Sci 120(30):e2305016120

Grimmer J, Stewart BM (2013) Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Anal 21(3):267–297

Haselmayer M, Jenny M (2017) Sentiment analysis of political communication: Combining a dictionary approach with crowdcoding. Qual Quant 51:2623–2646

Heseltine M, Clemm von Hohenberg B (2024) Large language models as a substitute for human experts in annotating political text. Res Polit 11(1):20531680241236239

Hicks MT, Humphries J, Slater J (2024) ChatGPT is bullshit. Ethics Inf Technol 26(2):38

Kolbe M (2021) When politics trumps economics: Contrasting high-skilled immigration policymaking in Germany and Austria. Int Migr Rev 55(1):31–57

Krippendorff K (2018) *Content analysis: An introduction to its methodology*. Sage publications

Laver M, Garry J (2000) Estimating policy positions from political texts. Am J Polit Sci pp. 619–634

Motoki F, Pinho Neto V, Rodrigues V (2024) More human than human: measuring ChatGPT political bias. Public Choice 198(1):3–23

Overos HD, Hlatky R, Pathak O, Goers H, Gouws-Dewar J, Smith K, Chew KP, Birnir JK, Liu AH (2024) Coding with the machines: machine-assisted coding of rare event data. PNAS nexus 3(5):165

Palmer A, Smith NA, Spirling A (2024) Using proprietary language models in academic research requires explicit justification. Nat Comput Sci 4(1):2–3

Pit P, Ma X, Conway M, Chen Q, Bailey J, Pit H, Keo P, Diep W, Jiang YG (2024) "Whose side are you on? Investigating the political stance of large language models." *arXiv preprint arXiv:2403.13840*

Rotaru GC, Anagnoste S, Oancea V M (2024) How Artificial Intelligence Can Influence Elections: Analyzing the Large Language Models (LLMs) Political Bias. In Proceedings of the International Conference on Business Excellence. Vol. 18 pp. 1882–1891

Rozado D (2024) The political preferences of llms. PloS one 19(7):e0306621

Schaffner BF, Luks S (2018) Misinformation or expressive responding? What an inauguration crowd can tell us about the source of political misinformation in surveys. Public Opin Q 82(1):135–147

Timoneda JC, Vallejo Vera S (2025) BERT, RoBERTa or DeBERTa? Comparing Performance Across Transformer Models in Political Science Text. J Polit87(1):347–364

Urman A, Makhortykh, M (2025) "The silence of the LLMs: Cross-lingual analysis of political bias and false information prevalence in ChatGPT, Google Bard, and Bing Chat." OSF Preprints https://doi.org/10.31219/osf.io/q9v8f

Vijay S, Priyanshu A, KhudaBukhsh AR (2025) When Neutral Summaries Are Not That Neutral: Quantifying PoliticalNeutrality in LLM-Generated News Summaries (Student Abstract). In Proceedings of the AAAI Conference on Artificial Intelligence 39:29514-29516

Walker C. Timoneda JC (2024) "Identifying the sources of ideological bias in GPT models through linguistic variation in output." arXiv preprint arXiv:2409.06043

Webb Williams N, Casas A, Aslett K, Wilkerson JD (2023) "When Conservatives See Red but Liberals Feel Blue: Why Labeler-Characteristic Bias Matters for Data Annotation." Available at SSRN 4540742

Yang K, Li H, Chu Y, Lin Y, Peng TQ, Liu H (2024) "Unpacking political bias in large language models: insights across topic polarization." arXiv preprint arXiv:2412.16746

Yang Z, Yi X, Li P, Liu Y, Xie X (2022) "Unified detoxifying and debiasing in language generation via inference-time adaptive optimization." arXiv preprint arXiv:2210.04492

Zhang Y, Mao S, Ge T, Wang X, de Wynter A, Xia Y, Wu W, Song T, Lan M, Wei F (2024) Llm as a mastermind: A survey of strategic reasoning with large language models. arXiv preprint arXiv:2404.01230

Acknowledgements

We would like to thank Laurenz Ennser-Jedenastik and Thomas M. Meyer from the University of Vienna for kindly sharing their data and code with us.

Author contributions

S.V. prepared the code, analyzed the results, and wrote the main manuscript text. H.D. ran the code, checked the results, and helped write the supplementary material. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Ethical approval

This study does not involve human participants or their data.

Informed consent

This study does not involve human participants or their data.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1057/s41599-025-05834-4.

Correspondence and requests for materials should be addressed to Sebastián Vallejo Vera.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.