Political Actor Agent: Simulating Legislative System for Roll Call Votes Prediction with Large Language Models

Hao Li, Ruoyuan Gong, Hao Jiang*

Wuhan University, Wuhan, 430072 China
whulh@whu.edu.cn, GongRuoyuan@whu.edu.cn, jh@whu.edu.cn

Abstract

Predicting roll call votes through modeling political actors has emerged as a focus in quantitative political science and computer science. Widely used embedding-based methods generate vectors for legislators from diverse data sets to predict legislative behaviors. However, these methods often contend with challenges such as the need for manually predefined features, reliance on extensive training data, and a lack of interpretability. Achieving more interpretable predictions under flexible conditions remains an unresolved issue. This paper introduces the Political Actor Agent (PAA), a novel agent-based framework that utilizes Large Language Models to overcome these limitations. By employing role-playing architectures and simulating legislative system, PAA provides a scalable and interpretable paradigm for predicting roll call votes. Our approach not only enhances the accuracy of predictions but also offers multi-view, human-understandable decision reasoning, providing new insights into political actor behaviors. We conducted comprehensive experiments using voting records from the 117-118th U.S. House of Representatives, validating the superior performance and interpretability of PAA. This study not only demonstrates PAA's effectiveness but also its potential in political science research.

Introduction

Legislative actions, such as proposing, reviewing, and voting on bills, enable political actors to influence national and societal development. Modeling these actors has emerged as an interdisciplinary focus within quantitative political science and computer science. The representations obtained from modeling political actors are applied to downstream tasks such as roll-call vote prediction (Feng et al. 2022; Mou et al. 2021), and political stance prediction (Feng et al. 2021; Li and Goldwasser 2019).

In this paper, we primarily focus on the problem of predicting legislators' roll call votes.With voting records, bill texts, and background knowledge, there are two main approaches to modeling political actors. The ideal point model, one of the most widely used methods for roll call vote prediction, represents legislators and bills as points in one or multiple dimensions (Clinton, Jackman, and Rivers 2004; Kraft, Jain, and Rush 2016). Recently, some studies have employed heterogeneous information graphs to represent legislators, bills, and contextual knowledge, including complex relationships between party affiliations, lobbying (Davoodi, Waltenburg, and Goldwasser 2020), and assets (Feng et al. 2022, 2021; Mou et al. 2024). These studies then use heterogeneous graph neural networks to generate embeddings for nodes within the graph and predict voting outcomes. Both approaches embed legislators and bills into a vector space and use neural networks or similarity measures to predict results.

However, the aforementioned embedding-based methods exhibit several limitations: 1. **Limitations of predefined features:** The model's training relies solely on predefined features, preventing natural extension to new, untrained relationships. 2. **Volume of training data:** Most models depend on large datasets to achieve optimal performance, which is not feasible in real-world scenarios, such as predicting votes of newly elected legislators. 3. **Interpretability of predictions:** Predictions based on embeddings lack interpretability, particularly in providing insights in a manner understandable to humans.

To address these challenges, we have turned our attention to the accomplishments of agent research based on Large Language Models (LLMs) (Wang et al. 2024). With designed profile, planning, and action modules, LLM agents can exhibit intelligent decision-making behaviors. Some studies have applied LLM Agents in fields such as economics (Kim et al. 2024; Zhou et al. 2024a), social simulation (Mou, Wei, and Huang 2024; Chuang et al. 2024; Dai et al. 2024), and voting decision-making (Yang et al. 2024; Majumdar, Elkind, and Pournaras 2024a). As shown in figure 1, by reframing the problem of modeling political actors as constructing political agents, we introduce deeper insights into this field.

Specifically, we designed the Political Actor Agent (PAA) based on a role-playing architecture (Li et al. 2023), which offers several advantages: 1. **Scalable Politician Profile:** Each agent is equipped with a scalable profile. This profile is more flexible and easier to manage compared to manually designed relational rules. 2. **Multi-view Planning:** From various views, such as the delegate and trustee views (Alexander 2019), the PAA can formulate different voting plans. Decision-making reasons understandable to hu-

*Corresponding author

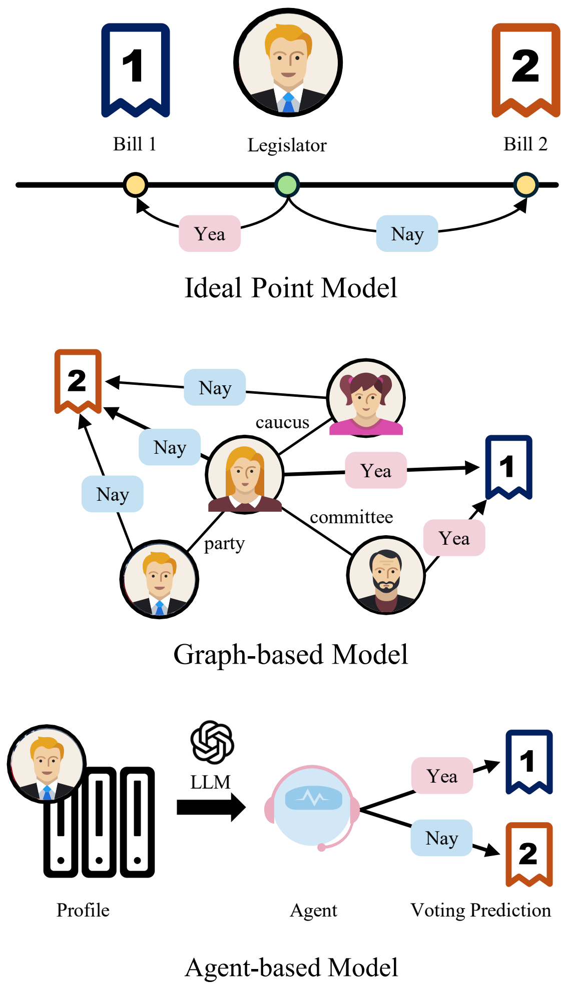

Figure 1: Examples of different political actor modeling methods include: The ideal point model represents legislators and bill entities as vectors. The graph-based model embeds nodes from heterogeneous information graphs into vectors using a graph embedding model. Our agent-based model does not rely on distances between embeddings; instead, it directly generates voting outcomes using LLM agents.

mans can better provide new insights for political science research. 3. **Simulated Legislative Action:** Based on legislators' voting strategies, we developed an Influence Mechanism that simulates parliamentary dynamics. Legislators are categorized as leader agents and follower agents, with follower agents being influenced by the voting outcomes of leader agents. This mechanism allows for accurate vote predictions even with limited known data. Our contributions are as follows:

1. We propose a new agent-based paradigm for political actor research. Compared to embedding-based methods, our role-playing framework for legislator simulation offers more accurate and interpretable results for corresponding downstream tasks.
2. We introduce the Political Actor Agent (PAA) for roll-call vote prediction. Our approach, through the design of scalable profiles, multi-view planning, and simulated legislator actions, significantly enhances prediction accuracy and offers interpretable decision-making insights for political science research.
3. We conduct comprehensive experiments using voting records from the 117-118th House of Representatives. Our experiments demonstrate that our method not only achieves high prediction accuracy but also provides interpretable political insights.

Related Works

This section introduces roll call voting prediction based on political actor modeling and the use of LLM agents for decision simulation.

Modeling Political Actors for Voting Prediction

Legislators' voting behavior in parliament has been a primary research focus due to its characteristic transparency and significance. One of the most popular techniques in political science is the ideal point model, constructed based on voting records and typically used to represent unidimensional or multidimensional ideological positions (Poole and Rosenthal 1985; Clinton, Jackman, and Rivers 2004). The ideal point model has been expanded in several studies. For instance, (Gerrish and Blei 2011) employ a topic model to perform a detailed analysis of legislative texts, enhancing the contextual understanding of votes. Further extending this approach, researchers have developed a topic factorized ideal point model that assigns ideal points for each topic rather than globally, allowing for more nuanced interpretations of legislative behavior (Gu et al. 2014). Additionally, efforts have been made to learn multidimensional embeddings of legislators and bills to improve prediction accuracy (Kraft, Jain, and Rush 2016), and to integrate broader political texts, such as speeches and tweets, into the ideal point model to enrich the dataset (Vafa, Naidu, and Blei 2020).

In terms of incorporating richer contextual information, graph-based methods, driven by advancements in knowledge graphs and graph neural networks, have gained popularity. External knowledge is introduced into voting prediction in the form of heterogeneous information graphs. Compared to ideal point models, graph-based models can more flexibly capture complex political relationships, such as cosponsorship (Yang et al. 2021), donors (Davoodi, Waltenburg, and Goldwasser 2020), and stakeholders (Davoodi, Waltenburg, and Goldwasser 2022). Additionally, more expert knowledge has been incorporated, such as news data (Feng et al. 2021), Twitter statements (Mou et al. 2021), wiki pages, and political think tanks (Feng et al. 2022).

Political decision-making often involves highly complex mechanisms and background knowledge. Recent studies have focused on leveraging large language models (LLMs). For instance, (Mou et al. 2024) constructed a multi-view political knowledge graph to enhance the domain knowledge of LLMs. Similarly, (Mou et al. 2023) developed a pretraining architecture that maps language to actor representations.

These attempts demonstrate the capability of LLMs in related tasks but remain complementary to embedding-based methods. This paper proposes a novel approach to modeling legislators as actors from an agent view, aiming to advance the prediction and understanding of voting behavior.

LLM Agent in Decision Simulation

Roll call voting can be viewed as a decision-making behavior. As LLM agents demonstrate the potential for human-level intelligence (Wang et al. 2024), many studies have applied them in fields such as natural sciences (Boiko, MacKnight, and Gomes 2023), software engineering (Qian et al. 2023), and embodied intelligence (Wu et al. 2023). Some researchers are exploring the integration of large language models with social sciences, particularly in simulating decision-making behaviors. In economics, for instance, (Horton 2023) treat LLM agents as economic agents, observing their economic decisions under different conditions and scenarios. (Zhou et al. 2024b) propose a financial bias indicator framework, analyzing irrational biases in LLM agents through behavioral finance theories.

In the realm of voting decision simulation, studies like (Argyle et al. 2023) investigate the feasibility of using large language models to simulate human samples, designing agents based on demographic data to model the reactions of different populations. The article (Yang et al. 2024) finds that the input method and presentation of choices can affect the voting behavior of LLMs. (Majumdar, Elkind, and Pournaras 2024b) delve deeper into the issue of bias in the voting process, discovering that equal-share methods can lead to fairer voting outcomes.

The above research on decision simulation indicates that LLM agents can offer new insights into social sciences. Unlike agents based on demographic data, this paper explores a more sophisticated decision simulation. By setting detailed political profiles and voting mechanisms, we aim to predict the roll-call voting results of political actors. Our goal is to investigate the ability of LLM agents to simulate complex, diverse outcomes and to summarize their behavior based on specific political perspectives.

Method

In this section, we detail the framework of the Political Actor Agent (PAA) for roll call vote prediction. As illustrated in figure 2, we begin by collecting data from Wikipedia and legislative bills from the Congress. We then construct scalable agent profiles for each legislator, incorporating personal information, constituency details, and data on bills they have sponsored and voted on. Subsequently, we design a multi-view planning module for each agent to perform reasoning based on various political science views. Finally, in the simulated legislative action module, we implement an influence mechanism that allows leader agents associated with the bill to act first, with the remaining agents making their choices after learning the leaders' decisions. This approach leverages political science perspectives and LLM agents to achieve more accurate and interpretable voting predictions.

Profile Construction Module

Using a role-playing architecture, we simulate characters where agents assume specific roles to make voting decisions. The profile construction module is integrated into the prompt to influence the design and behavior of the LLMs. The design of the profile is highly scalable and can synthesize information from different data sources under various conditions, such as personal basic information, career history, voting records, etc. Specifically, the profiles we use include the following personal details:

Personal information This refers to the basic information of legislators such as party affiliation, committee memberships, core group affiliations, educational background, number of children, place of birth, and other relevant details.

Constituency details This section details the legislator's constituency, including information such as the median family income of the district, urban-rural population distribution, and total population. This information helps model the social and economic context in which the legislator operates.

Sponsorship activity Each bill is initiated by a sponsor and possibly several cosponsors. Historical records of bill sponsorship are extremely useful for modeling a legislator's specific preferences. Past studies have also demonstrated the significance of sponsorship information in political actor modeling (Yang et al. 2021).

Voting records: Historical voting records provide the most direct data on a legislator's political stance. Unlike previous political actor models, the PAA integrates past voting records directly into the profile without explicit model training. Leveraging the capabilities of large language models, we can predict voting outcomes based on a limited amount of known information, which is particularly useful when modeling newly elected legislators with less data available.

Multi-view Planning Module

Inspired by political science, the planning module decomposes the task of voting decision-making into three main views, aiding the agent in decision-making. In the appendix, we present the prompts used for these different views (Alexander 2019).

Trustee view The Trustee view implies that the legislator relies on their expertise, making decisions based on what they believe to be the best policies for their constituents and the country.

Delegate view The Delegate view indicates that the legislator sees their role as expressing the will of the majority of their constituents, acting in accordance with the wishes of their voters.

Follower view Legislators in this category follow the opinions of their party leaders. They are not keen on reflecting public opinion directly and often lack substantial personal insights.

Practical evidence shows that legislators' decisions are influenced by multiple views. After planning through these

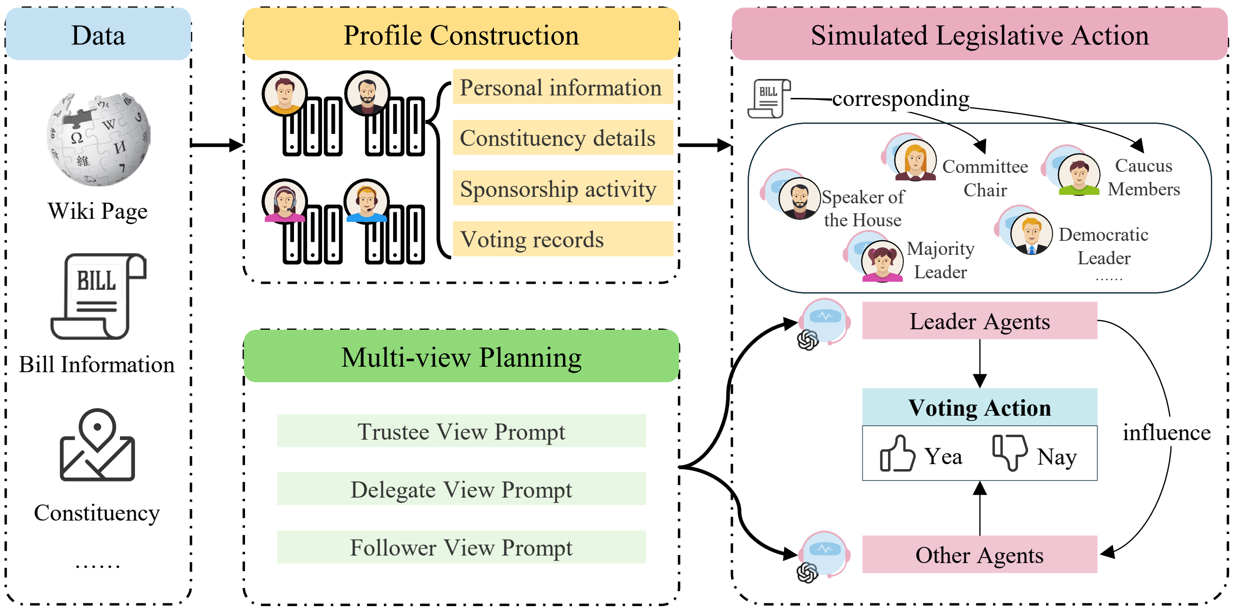

Figure 2: Framework of PAA

various views, the agent synthesizes these results to arrive at the final decision.

Simulated Legislative Action Module

Previous political actor models, while recognizing differences among legislators, typically generated predictions in a single step. Embedding-based methods fail to intricately simulate the decision-making processes of real legislators and do not effectively model the influence of leader figures. In the PAA, we have designed an "influence mechanism" to model how leading agents influence other agents.

Leader agents Leader agents, denoted as L , comprise the following agents:

$$L = \{S, R, D, CC, CM\}, \tag{1}$$

where S represents the Speaker of the House, R and D represent the Republican and Democratic Leader, respectively, with information sourced from the official website [1]. CC denotes the Chairperson of the committee introducing the bill, and CM represents caucus members related to the bill. A congressional caucus is a group of members of the United States Congress who meet to pursue common legislative objectives, often actively pushing legislation through their actions.

Influence mechanism In the influence mechanism, leader agents first vote based on the multi-view planning module.

$$V_l = p(L), \tag{2}$$

where V_l represents the voting prediction of leader agents, and p represents the multi-view planning module. The voting prediction for the remaining agents O is as follows:

$$V_o = p(O|V_l), \tag{3}$$

[1] https://www.house.gov/leadership

where V_o represents the voting predictions of other agents, made under the condition of knowing V_l, $p(O|V_l)$ indicates that we include V_l in the prompt to predict the vote of agent O. This method effectively approximates the real-world legislative process. It is worth noting that this influence mechanism is highly flexible and can be adapted to different conditions for selecting leadership agents. The configuration presented here is just one of the possible scenarios.

Experiments

In this section, we conduct a detailed evaluation of the Political Actor Agent (PAA) in predicting roll call votes. Section 4.1 introduces the datasets selected for the experiment and the baselines used. Section 4.2 details the experimental results across different dataset splits. Sections 4.3 and 4.4 feature extensive ablation studies, analyzing the impact of various PAA modules on the outcomes, with a deep dive into the profile module. Section 4.5 demonstrates the consistency of the PAA's results. Section 4.6 provides an interpretable example to illustrate how the PAA generates its predictions.

Datasets and Baselines

We selected voting data from the 117th to 118th House of Representatives, covering 432 legislators. In addition to the voting data for bills, we collected additional data for constructing profiles and heterogeneous information graphs. This includes the most recent Wikipedia pages of the legislators (as of March 2024), Wikipedia pages for all constituencies, data on the sponsors and cosponsors of bills, and Twitter posts by the legislators. We selected five methods as baselines, including a variant of the ideal point model, three graph-based models, and one that incorporates a pre-trained model.

1. **ideal-vector** (Kraft, Jain, and Rush 2016): A multi-dimensional ideal point model based on word embeddings, learning politician representations from bill texts.
2. **LSTM+GCN** (Yang et al. 2021): A graph-based model using a Graph Convolutional Network (GCN) to generate representations of legislators and an LSTM to generate representations of bill texts.
3. **Vote+MTL** (Mou et al. 2021): A graph-based model that incorporates Twitter data, using a Relational Graph Convolutional Network (RGCN) to generate representations of politicians.
4. **PAR** (Feng et al. 2022): A graph-based model that combines various socio-contextual information, integrating representational models with expert knowledge to generate politician representations.
5. **UPPAM** (Mou et al. 2023): A contrastive learning framework based on the social network of legislators and bill texts, using pre-trained models to generate politician representations.

Roll Call Vote Prediction

Implementation We divided the dataset in chronological order and selected three different ratios for dataset splits.

1. $split_{244}$: 20% training, 40% validation, 40% testing.
2. $split_{433}$: 40% training, 30% validation, 30% testing.
3. $split_{622}$: 60% training, 20% validation, 20% testing.

As an agent-based approach, the PAA does not train models. Instead, we sample 20 voting records from the training set to construct the profile for each agent and directly evaluate the results on the test set.

Our methods were tested using Llama-3-70B (PAA_L) and GPT-4o-mini (PAA_G) as the base models. We chose accuracy and macro-averaged F1-score as metrics to evaluate the model's performance on a three-class task: in favor, against, and abstaining. We conducted experiments for PAA_L on a four NVIDIA RTX A6000 GPUs, while the PAA_G experiments were carried out using the OpenAI API[2].

Results The results are presented in table 1. The primary findings of the experiment indicate that PAA_G consistently outperformed across all dataset splits, achieving superior results compared to state-of-the-art methods. PAA_L performed particularly well in terms of the macro-averaged F1 score. This is largely due to the label imbalance in the three-class voting problem. As the proportion of the training set decreases within the dataset, the performance of embedding-based methods declines more rapidly, whereas PAA remains more stable. This suggests that PAA is better adapted to scenarios with limited data, such as predicting the voting behavior of new legislators.

PAR and UPPAM performed next best to PAA_G in the $split_{622}$. PAR learns representations of legislators through external knowledge and social media corpora, while UPPAM utilizes extensive social media data to train a pre-trained model. Their performance in the voting prediction task also validates the capabilities of large-scale external knowledge and pre-trained models. Unlike these methods, our approach does not utilize any social media data. With the aid of LLMs, PAA achieves comparable or even superior results under less time and data constraints.

Ablation Studies

Implementation We designed ablation experiments to further verify the impact of various modules on the performance of the PAA. On the $split_{244}$, while keeping other conditions constant, we individually removed the profile module, the planning module, and the acting module to assess their individual contributions to the PAA's effectiveness.

1. PAA w/o Pro: Remove profile module.
2. PAA w/o Pla: Remove planning module.
3. PAA w/o Act: Remove acting module.

Result The experimental results are shown in table 2. The results indicate that different modules impact the performance of the PAA to varying degrees. Notably, the profile module has the greatest effect, as it contains a wealth of crucial information, including personal details, voting records, and more. Next, we will further explore the influence of the profile on the performance of the PAA.

Analysis of Profile Module

Implementation The profile module comprises four key components: legislator personal information, constituency details, legislative sponsorship activity, and voting records. We have designed experiments to address specific concerns related to the performance of the PAA. **RQ1**: The outstanding performance of PAA might be attributed to the fact that the agent, based on a large language model, had already been exposed to legislative information and legislator-related corpora during the pre-training phase. **RQ2**: Which part of the agent profile plays a more significant role in predicting outcomes? **RQ3**: Does an excessive length of voting records dilute the importance of other components and adversely affect the performance of the experiments?

Based on the considerations above, for RQ1 and RQ2, we conducted detailed experiments on the $split_{244}$ dataset with PAA_G. For RQ3, we devised an experiment to analyze the effect of the length of voting records on the performance of the PAA. One approach involves sampling 20 data points from the training set to add to the profile, while another uses the entire training set. The PAA variants used in the experiment are as follows:

1. PAA_{ano}: Anonymize the names of legislators and bill numbers. Names and bill numbers are replaced with random numbers.
2. PAA_{dec}: Introduce incorrect information to deceive the large language model; we swapped names of legislators with differing stances to observe whether PAA predicts based on pre-trained data.
3. PAA w/o Per: Remove basic information about legislators from the profile.
4. PAA w/o Con: Remove constituency information from the profile.

[2]https://platform.openai.com/docs/api-reference/introduction

Method	split$_{244}$		split$_{433}$		split$_{622}$	
	acc	f1	acc	f1	acc	f1
ideal-vector	80.9 ± 0.12	79.2 ± 0.13	82.2 ± 0.23	80.6 ± 0.36	86.1 ± 0.32	84.4 ± 0.25
LSTM+GCN	83.5 ± 0.13	82.5 ± 0.12	85.3 ± 0.12	83.2 ± 0.35	87.8 ± 0.08	85.2 ± 0.12
Vote+MTL	83.2 ± 0.04	83.5 ± 0.12	84.2 ± 0.20	84.3 ± 0.03	89.5 ± 0.14	86.2 ± 0.06
PAR	85.3 ± 0.11	80.2 ± 0.12	85.8 ± 0.12	85.2 ± 0.12	90.2 ± 0.23	87.5 ± 0.03
UPPAM	86.5 ± 0.10	80.5 ± 0.07	88.5 ± 0.03	85.9 ± 0.03	91.7 ± 0.08	86.3 ± 0.09
PAA$_L$	85.9 ± 0.10	87.2 ± 0.08	85.7 ± 0.10	88.1 ± 0.07	87.7 ± 0.05	89.6 ± 0.05
PAA$_G$	**91.8 ± 0.15**	**92.2 ± 0.10**	**91.3 ± 0.20**	**91.7 ± 0.12**	**92.1 ± 0.10**	**93.0 ± 0.12**

Table 1: Results of various methods on different splits. The best results are highlighted in bold, and the second-best results are underlined. Each experimental group was run five times, and we show the mean and standard deviation in the table.

Method	split$_{244}$	
	acc	f1
PAA w/o Pro	78.6 ± 0.10	73.4 ± 0.07
PAA w/o Pla	80.1 ± 0.12	74.1 ± 0.13
PAA w/o Act	80.7 ± 0.19	74.2 ± 0.16
PAA$_G$	91.8 ± 0.15	92.2 ± 0.10

Table 2: The results of ablation experiments conducted on the profile, planning, and action modules.

5. PAA w/o Spo: Remove information about Sponsored and Cosponsored Bills from the profile.
6. PAA w/o Rec: Remove voting records from the profile.
7. PAA$_G^F$, PAA$_L^F$: Use the entire training set data for voting in the profile.

Results The results are presented in table 3. From the findings, we can see that both PAA$_{ano}$ and PAA$_{dec}$ exhibit a decrease in predictive performance when names and bill identifiers are modified. The fact that PAA$_{dec}$ performs slightly worse than PAA$_{ano}$ suggests that legislator information has a certain impact on the experimental results. Yet, both variants still significantly outperform the current baselines. Our findings indicate that model performance is only slightly affected by the names, suggesting that PAA likely relies on the information in our profile module for predictions.

Additionally, the study of the profile reveals that all four types of information we selected are effective for prediction. Basic information, sponsorship, and voting records have the most significant impact, while constituency information has the least. This might be because the relationship between constituency and voting behavior is less apparent compared to other information, suggesting the need for more explicit mechanisms to better utilize this data.

The experimental results, as shown in figure 3, indicate that as the size of the training set increases, the performance of PAA$_L$ and PAA$_G$ improves, achieving the best results on the split$_{622}$ dataset. However, the performance of PAA$_L^F$ and PAA$_G^F$ gradually declines, with the worst results observed on the split$_{622}$ dataset. This suggests that an excessive number of voting records can confuse the large language model, preventing it from capturing other important information. Therefore the PAA opts for a sampling method to construct profiles, achieving better performance with fewer resources.

Method	split$_{244}$	
	acc	f1
PAA-ano	90.8 ± 0.10	91.3 ± 0.08
PAA-Dec	90.1 ± 0.08	90.7 ± 0.11
PAA w/o Per	82.9 ± 0.17	76.3 ± 0.23
PAA w/o Con	90.9 ± 0.06	91.3 ± 0.09
PAA w/o Spo	83.6 ± 0.09	78.9 ± 0.05
PAA w/o Rec	83.4 ± 0.13	77.8 ± 0.19
PAA$_G$	91.8 ± 0.15	92.2 ± 0.10

Table 3: The results of the ablation experiments that modified the profile module.

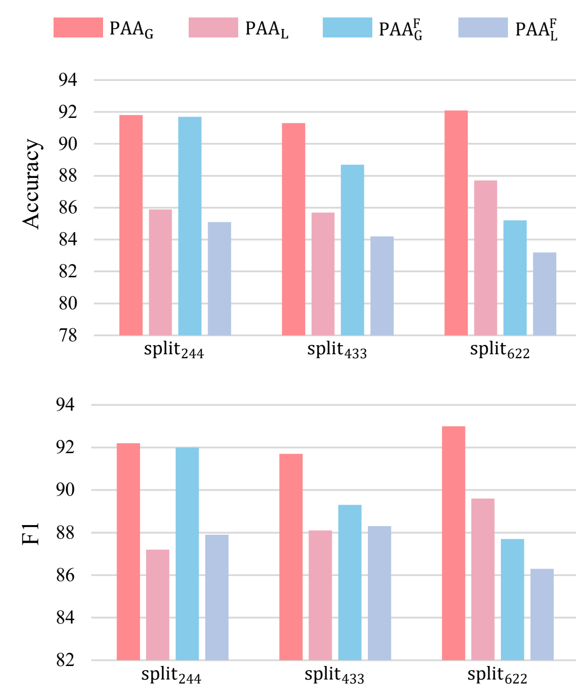

Figure 3: The results on the impact of profile length on the performance of the PAA.

Consistency Analysis

Implementation Unlike other embedding-based models that generate representations of legislators, large language models are prone to underlying hallucinations, potentially generating different results across multiple runs. Therefore, we analyzed the consistency of results produced by the PAA. We randomly selected 50 legislator agent and bill pairs and repeated the experiment 20 times.

Results The experimental results using PAA_G and PAA_L are shown in figure 4 respectively. In the heatmaps, red indicates correct predictions, while blue denotes incorrect predictions. The horizontal axis represents different experimental runs, and the vertical axis represents different agent-bill pairs. Overall, we observed that the hallucination phenomenon commonly associated with large language models was not significant. Specifically, PAA_L showed stronger consistency, with only a few instances where the same agent-bill pair produced different results across experiments. Meanwhile, PAA_G achieved a higher number of correct predictions, though its consistency was not as strong as that of Llama-3-70B. Thus, conducting multiple experiments to mitigate the effects of hallucinations is highly beneficial.

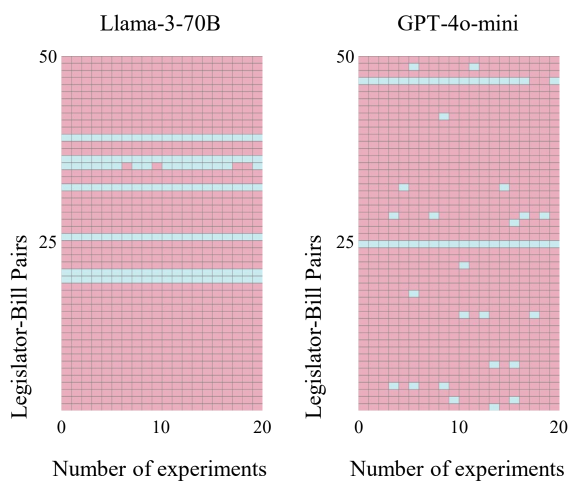

Figure 4: The consistency experiment results.

Interpretability

As shown in figure 5, we present an example to illustrate the interpretability of the PAA's voting prediction results. In this example, the Agent explains its choice from three different views, each backed by factual evidence. It is noteworthy that he mentioned his conservative inclinations and the corresponding choice of leadership agent, which highlights the effectiveness of our multi-view planning module and influence mechanism.

Conclusion

In this study, we introduced the Political Actor Agent (PAA), an innovative approach leveraging Large Language Models (LLMs) for the predictive modeling of legislative be-

Legislator 026- "250th Anniversary of the United States Marine Corps Commemorative Coin Act (H.R. 1096)":
Real result: Yea
Prediction result: Yea
Reason:
Trustee: As a proud Republican with deep respect for our military, supporting this bill aligns with my career focus on honoring those who serve. I believe it's essential to recognize and celebrate the sacrifices made by our armed forces, which reflects my commitment to upholding our nation's values.
Delegate: My district has a strong military presence, and many of my constituents have served or have family members in the military. By supporting this bill, I am not only honoring their service but also ensuring that their legacy is remembered, which is something my voters deeply care about.
Follower: The Republican leadership has shown strong support for this bill, emphasizing its importance in celebrating our military's heritage. Their endorsement reinforced my decision, as it reflects our party's shared commitment to the armed forces.

Figure 5: An example demonstrating how an agent cast a vote and subsequently summarize the reasons for its decision.

havior. By incorporating agents in a role-playing architecture, PAA uniquely simulates the dynamics of legislative decision-making, providing a robust framework for understanding and predicting roll call votes. The utilization of extensive pre-existing knowledge and reasoning capabilities from LLMs ensures high accuracy and interpretability without relying on massive bespoke training datasets. Additionally, although our experiments focused on U.S. legislators, PAA can be easily extended to other countries.

However, the PAA also has several limitations: 1. **Data Diversity:** Despite achieving superior performance compared to baselines with fewer data types, the current architecture lacks support for integrating diverse data sources like social media commentary and news. 2. **Task Diversity:** Compared to existing political actor modeling methods, PAA primarily supports the task of roll call vote prediction. Developing mechanisms to support more downstream tasks remains an unresolved issue. 3. **Hallucination:** Stable and consistent prediction results are essential, although we have analyzed PAA's results from a consistency perspective, The hallucination issue in LLMs is complex, and consistency is only one measure of it.

In the future, we plan to further explore the integration of more diverse data types, such as real-time social media analytics and global news events, to enhance the predictive power of PAA. Additionally, we aim to design more versatile mechanisms to accommodate different downstream tasks, broadening the applicability and utility of the PAA in computational political science.

Acknowledgments

This work was supported by the Key R&D Program of China under Grant 2022YFB2901202.

References

Alexander, R. M. 2019. *Representation and the Electoral College*. Oxford University Press.

Argyle, L. P.; Busby, E. C.; Fulda, N.; Gubler, J. R.; Rytting, C.; and Wingate, D. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3): 337–351.

Boiko, D. A.; MacKnight, R.; and Gomes, G. 2023. Emergent autonomous scientific research capabilities of large language models. *arXiv preprint arXiv:2304.05332*.

Chuang, Y.-S.; Studdiford, Z.; Nirunwiroj, K.; Goyal, A.; Frigo, V. V.; Yang, S.; Shah, D.; Hu, J.; and Rogers, T. T. 2024. Beyond Demographics: Aligning Role-playing LLM-based Agents Using Human Belief Networks. *arXiv preprint arXiv:2406.17232*.

Clinton, J.; Jackman, S.; and Rivers, D. 2004. The statistical analysis of roll call data. *American Political Science Review*, 98(2): 355–370.

Dai, G.; Zhang, W.; Li, J.; Yang, S.; Rao, S.; Caetano, A.; Sra, M.; et al. 2024. Artificial Leviathan: Exploring Social Evolution of LLM Agents Through the Lens of Hobbesian Social Contract Theory. *arXiv preprint arXiv:2406.14373*.

Davoodi, M.; Waltenburg, E.; and Goldwasser, D. 2020. Understanding the language of political agreement and disagreement in legislative texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5358–5368.

Davoodi, M.; Waltenburg, E.; and Goldwasser, D. 2022. Modeling US state-level policies by extracting winners and losers from legislative texts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 270–284.

Feng, S.; Chen, Z.; Zhang, W.; Li, Q.; Zheng, Q.; Chang, X.; and Luo, M. 2021. Kgap: Knowledge graph augmented political perspective detection in news media. *arXiv preprint arXiv:2108.03861*.

Feng, S.; Tan, Z.; Chen, Z.; Wang, N.; Yu, P.; Zheng, Q.; Chang, X.; and Luo, M. 2022. PAR: Political actor representation learning with social context and expert knowledge. *arXiv preprint arXiv:2210.08362*.

Gerrish, S. M.; and Blei, D. M. 2011. Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*.

Gu, Y.; Sun, Y.; Jiang, N.; Wang, B.; and Chen, T. 2014. Topic-factorized ideal point estimation model for legislative voting network. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 183–192.

Horton, J. J. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research.

Kim, J.; Kovach, M.; Lee, K.-M.; Shin, E.; and Tzavellas, H. 2024. Learning to be Homo Economicus: Can an LLM Learn Preferences from Choice. *arXiv preprint arXiv:2401.07345*.

Kraft, P.; Jain, H.; and Rush, A. M. 2016. An embedding model for predicting roll-call votes. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2066–2070.

Li, C.; and Goldwasser, D. 2019. Encoding social information with graph convolutional networks forpolitical perspective detection in news media. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2594–2604.

Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008.

Majumdar, S.; Elkind, E.; and Pournaras, E. 2024a. Generative AI Voting: Fair Collective Choice is Resilient to LLM Biases and Inconsistencies. *arXiv preprint arXiv:2406.11871*.

Majumdar, S.; Elkind, E.; and Pournaras, E. 2024b. Generative AI Voting: Fair Collective Choice is Resilient to LLM Biases and Inconsistencies. *arXiv preprint arXiv:2406.11871*.

Mou, X.; Li, Z.; Lyu, H.; Luo, J.; and Wei, Z. 2024. Unifying Local and Global Knowledge: Empowering Large Language Models as Political Experts with Knowledge Graphs. In *Proceedings of the ACM on Web Conference 2024*, 2603–2614.

Mou, X.; Wei, Z.; Chen, L.; Ning, S.; He, Y.; Jiang, C.; and Huang, X.-J. 2021. Align voting behavior with public statements for legislator representation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1236–1246.

Mou, X.; Wei, Z.; and Huang, X. 2024. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. *arXiv preprint arXiv:2402.16333*.

Mou, X.; Wei, Z.; Zhang, Q.; and Huang, X.-J. 2023. Uppam: A unified pre-training architecture for political actor modeling based on language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11996–12012.

Poole, K. T.; and Rosenthal, H. 1985. A spatial model for legislative roll call analysis. *American journal of political science*, 357–384.

Qian, C.; Cong, X.; Yang, C.; Chen, W.; Su, Y.; Xu, J.; Liu, Z.; and Sun, M. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Vafa, K.; Naidu, S.; and Blei, D. 2020. Text-Based Ideal Points. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5345–5357.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wu, Z.; Wang, Z.; Xu, X.; Lu, J.; and Yan, H. 2023. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*.

Yang, J. C.; Korecki, M.; Dailisan, D.; Hausladen, C. I.; and Helbing, D. 2024. Llm voting: Human choices and ai collective decision making. *arXiv preprint arXiv:2402.01766*.

Yang, Y.; Lin, X.; Lin, G.; Huang, Z.; Jiang, C.; and Wei, Z. 2021. Joint representation learning of legislator and legislation for roll call prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 1424–1430.

Zhou, Y.; Ni, Y.; Liu, X.; Zhang, J.; Liu, S.; Ye, G.; and Chai, H. 2024a. Are Large Language Models Rational Investors? *arXiv preprint arXiv:2402.12713*.

Zhou, Y.; Ni, Y.; Liu, X.; Zhang, J.; Liu, S.; Ye, G.; and Chai, H. 2024b. Are Large Language Models Rational Investors? *arXiv preprint arXiv:2402.12713*.