

PAR: Political Actor Representation Learning with Social Context and Expert Knowledge

Shangbin Feng^{♣♦} Zhaoxuan Tan[♣] Zilong Chen^{♣♡} Ningnan Wang[♣]
 Peisheng Yu^{♣◊} Qinghua Zheng[♣] Xiaojun Chang[○] Minnan Luo^{♣*◊}
 Xi'an Jiaotong University[♣], University of Washington[♣], Tsinghua University[♡]
 University of California San Diego[◊], University of Technology Sydney[○]
 contact: shangbin@cs.washington.edu

Abstract

Modeling the ideological perspectives of political actors is an essential task in computational political science with applications in many downstream tasks. Existing approaches are generally limited to textual data and voting records, while they neglect the rich social context and valuable expert knowledge for holistic ideological analysis. In this paper, we propose **PAR**, a Political Actor Representation learning framework that jointly leverages social context and expert knowledge. Specifically, we retrieve and extract factual statements about legislators to leverage social context information. We then construct a heterogeneous information network to incorporate social context and use relational graph neural networks to learn legislator representations. Finally, we train PAR with three objectives to align representation learning with expert knowledge, model ideological stance consistency, and simulate the echo chamber phenomenon. Extensive experiments demonstrate that PAR is better at augmenting political text understanding and successfully advances the state-of-the-art in political perspective detection and roll call vote prediction. Further analysis proves that PAR learns representations that reflect the political reality and provide new insights into political behavior.

1 Introduction

Modeling the perspectives of political actors has applications in various downstream tasks such as roll call vote prediction (Mou et al., 2021) and political perspective detection (Feng et al., 2021). Existing approaches generally focus on voting records or textual information of political actors to induce their stances. Ideal point model (Clinton et al., 2004) is one of the most widely used approach for vote-based analysis, while later works enhance the ideal point model (Kraft et al., 2016; Gu et al.,

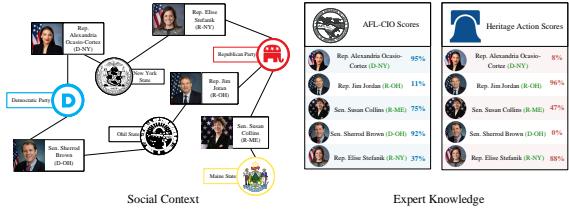


Figure 1: Social context information and political expert knowledge that helps to model political actors.

2014; Gerrish and Blei, 2011) and yield promising results on the task of roll call vote prediction. For text-based methods, text analysis techniques are combined with textual information in social media posts (Li and Goldwasser, 2019), Wikipedia pages (Feng et al., 2021), legislative text (Mou et al., 2021) and news articles (Li and Goldwasser, 2021) to enrich the perspective analysis process.

However, existing methods fail to incorporate the rich social context and valuable expert knowledge of political actors. As illustrated in Figure 1, social context information such as home state and party affiliation serves as background knowledge and helps connect different political actors (Yang et al., 2021). These social context facts about political actors also differentiate them and indicate their ideological stances. In addition, expert knowledge from political think tanks provides valuable insights and helps to anchor the perspective analysis process. As a result, political actor representation learning should be guided by domain expertise to facilitate downstream tasks in computational political science. That being said, social context and expert knowledge should be incorporated in modeling legislators to ensure a holistic evaluation.

In light of these challenges, we propose PAR, a legislator representation learning framework that jointly leverages social context and expert knowledge. We firstly collect a dataset of political actors by retrieving and extracting social context information from their Wikipedia homepages and

*Corresponding author.

adapting expert knowledge from two political think tanks AFL-CIO¹ and Heritage Action². After that, we construct a heterogeneous information network to model social context information and adopt relational graph neural networks for representation learning. Finally, we train the framework with three training objectives to leverage expert knowledge, model social and political phenomena, and learn representations of political actors in the process. We evaluate PAR on two computational political science tasks and examine the political behaviour of learned representations. Our main contributions are summarized as follows:

- We propose PAR, a graph-based approach to learn legislator representations with three training objectives, which aligns representation learning with expert knowledge, ensures stance consistency, and models the echo chamber phenomenon in socio-economic systems.
- Extensive experiments demonstrate that PAR advances the state-of-the-art on two computational political science tasks, namely political perspective detection and roll call vote prediction.
- Further analysis shows that PAR learns representations that reflect the ideological preferences and political behavior of political actors such as legislators, governors, and states. In addition, PAR provides interesting insights into the contemporary political reality.

2 Related Work

The ideological perspectives of political actors play an essential role in their individual behavior and adds up to influence the overall legislative process (Freedon, 2006; Bamman et al., 2012; Wilkerson and Casas, 2017). Political scientists first explored to quantitatively model political actors based on their voting behaviour. Ideal point model (Clinton et al., 2004) is one of the earliest approach to leverage voting records to analyze their perspectives. It projects political actors and legislation onto one-dimensional spaces and measure distances. Many works later extended the ideal point model. Gerrish and Blei (2011) leverages bill content to infer legislator perspectives. Gu et al. (2014) introduces topic factorization to model voting behaviour on different issues to establish a fine-grained approach. Kraft et al. (2016) models legislators with

multidimensional ideal vectors to analyze voting records. Mohammad et al. (2017) and Küçük and Can (2018) use SVM and handcrafted text features to augment the analysis.

In addition to voting, textual data such as speeches and public statements are also leveraged to model political perspectives (Evans et al., 2007; Thomas et al., 2006; Hasan and Ng, 2013; Zhang et al., 2022; Sinno et al., 2022; Liu et al., 2022; Davoodi et al., 2022; Alkiek et al., 2022; Dayanik et al., 2022; Pujari and Goldwasser, 2021; Villegas et al., 2021; Li et al., 2021; Sen et al., 2020; Baly et al., 2020). Volkova et al. (2014) leverage message streams to inference users political preference. Johnson and Goldwasser (2016) propose to better understand political stances by analyzing politicians' tweet and their temporal activities. Li and Goldwasser (2019) propose to analyze Twitter posts to better understand news stances. Prakash and Tayyar Madabushi (2020) and Kawintiranon and Singh (2021) leverage pre-trained language model for stance prediction. Augenstein et al. (2016) uses conditional LSTM to encode tweets for stance detection. Furthermore, CNN (Wei et al., 2016) and hierarchical attention networks (Sun et al., 2018a) are adopted for stance detection. Darwish et al. (2020) proposes an unsupervised framework for user representation learning and stance detection. Yang et al. (2021) proposes to jointly model legislators and legislations for vote prediction. Feng et al. (2021) introduces Wikipedia corpus and constructs knowledge graphs to facilitate perspective detection. Mou et al. (2021) proposes to leverage tweets, hashtags and legislative text to grasp the full picture of the political discourse. Li and Goldwasser (2021) designs pre-training tasks with social and linguistic information to augment political analysis. However, these vote and text-based methods fail to leverage the rich social context of political actors and the valuable expert knowledge of political think tanks. In this paper, we aim to incorporate social context and expert knowledge while focusing on learning representations of political actors to model their perspectives and facilitate downstream tasks.

3 Methodology

Figure 2 presents an overview of **PAR** (Political Actor Representation learning). We firstly collect a dataset of political actors from Wikipedia and political think tanks. We then construct a heterogeneous

¹<https://aflcio.org/>

²<https://heritageaction.com/>

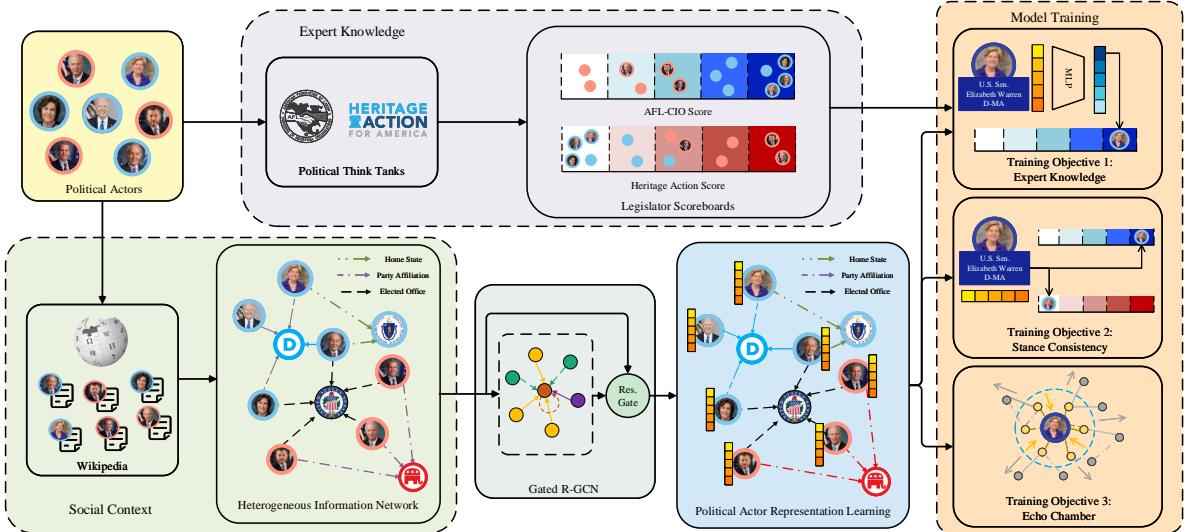


Figure 2: Overview of PAR, political actor representation learning with social context and expert knowledge.

information network to jointly model political actors and social context information. Finally, we learn graph representations with gated relational graph convolutional networks (gated R-GCN) and train the framework with three different objectives to leverage expert knowledge and model various socio-political phenomena.

3.1 Data Collection

We firstly collect a dataset about political actors in the United States that were active in the past decade. For **social context** information, we retrieve the list of senators and congresspersons from the 114th congress to the 117th congress³. We then retrieve their Wikipedia pages⁴ and extract these named entities: presidents, senators, congresspersons, governors, states, political parties, supreme court justices, government institutions, and office terms (117th congress etc.). In this way, we obtain 1,069 social and political entities. Based on these entities, we identify five types of relations: party affiliation, home state, political office, term in office, and appoint relationships. In this way, we obtain 9,248 heterogeneous edges. For **expert knowledge**, we use legislator scoreboards at AFL-CIO and Heritage Action, two political think tanks that lie in the opposite ends of the ideological spectrum. Specifically, we retrieve and extract each legislator’s score in each office term. In this way, we obtain 777 scores from AFL-CIO and 679 scores

from Heritage Action. We consolidate the collected social context and expert knowledge to serve as the dataset in our experiments. Although we focus on political actors in the United States in this paper, our data collection process is also applicable for other countries and time ranges.

3.2 Graph Construction

To better model the interactions between political entities and their shared social context, we propose to construct a heterogeneous information network (HIN) from the dataset.

3.2.1 Heterogeneous Nodes

Based on the collected dataset, we select diversified entities that are essential factors in modeling the political process. Specifically, we use eight types of nodes to represent political actors and diversified social context entities.

N1: Office Terms. We use four nodes to represent the 114th, 115th, 116th, 117th congress spanning from 2015 to 2022. We use these nodes to model different political scenarios and could be similarly extended to other time periods.

N2: Legislators. We retrieve senators and congresspersons from the 114th to 117th congress and use one node to represent each distinct legislator.

N3: Presidents. President is the highest elected office in the United States. We use three nodes to represent President Biden, Trump and Obama to match with the time range in *N1*.

N4: Governors. State and local politics are also essential in analyzing the political process. We use

³<https://www.congress.gov/>

⁴<https://github.com/goldsmith/Wikipedia>

one node to represent each distinct governor of 50 states within the time range of $N1$.

$\mathcal{N}5$: *States*. The home state of political actors is often an important indicator and helps connect different individuals. We use one node to represent each state in the United States.

$\mathcal{N}6$: *Government Institutions*. We use five nodes to represent the white house, senate, house of representatives, supreme court and governorship. These nodes enable our HIN to separate different political actors based on the office they hold.

$\mathcal{N}7$: *Supreme Court Justices*. Supreme court justices are nominated by presidents and approved by senators, which helps connect different types of political actors. We use one node to represent each supreme court justice within the time range of $N1$.

$\mathcal{N}8$: *Political Parties*. We use two nodes to represent the Republican Party and the Democratic Party in the United States.

For node features, we use pre-trained RoBERTa (Liu et al., 2019) to encode the first paragraph of Wikipedia pages and average all tokens.

3.2.2 Heterogeneous Relations

Based on $\mathcal{N}1$ to $\mathcal{N}8$, we extract five types of informative interactions between entities to complete the HIN structure. Specifically, we use five types of heterogeneous relations to connect different nodes and construct our political actor HIN.

$\mathcal{R}1$: *Party Affiliation*. We connect political actors and their affiliated political party with $R1$:

$$\mathcal{R}1 = (\mathcal{N}2 \cup \mathcal{N}3 \cup \mathcal{N}4) \times \mathcal{N}8 \quad (1)$$

$\mathcal{R}2$: *Home State*. We connect political actors with their home states with $R2$:

$$\mathcal{R}2 = (\mathcal{N}2 \cup \mathcal{N}3 \cup \mathcal{N}4 \cup \mathcal{N}7) \times \mathcal{N}5 \quad (2)$$

$\mathcal{R}3$: *Hold Office*. We connect political actors with the political office they hold with $R3$:

$$\mathcal{R}3 = (\mathcal{N}2 \cup \mathcal{N}3 \cup \mathcal{N}4 \cup \mathcal{N}7) \times \mathcal{N}6 \quad (3)$$

$\mathcal{R}4$: *Time in Office*. If a political actor holds office during one of the time stamps in $N1$, we connect them with $R4$:

$$\mathcal{R}4 = (\mathcal{N}2 \cup \mathcal{N}3 \cup \mathcal{N}4 \cup \mathcal{N}7) \times \mathcal{N}1 \quad (4)$$

$\mathcal{R}5$: *Appoint*. Besides from being elected, certain political actors are appointed by others. We denote this relation with $R5$:

$$\mathcal{R}5 = (\mathcal{N}3 \times \mathcal{N}7) \cup (\mathcal{N}4 \times \mathcal{N}2) \quad (5)$$

3.3 Representation Learning

Since nodes represent political actors, we learn node-level representations with gated R-GCN to jointly leverage social context and expert knowledge. Let $E = \{e_1, \dots, e_n\}$ be n entities in the HIN and v_i be the initial features of entity e_i . Let R be the heterogeneous relation set and $N_r(e_i)$ be entity e_i 's neighborhood with regard to relation type r . We firstly transform v_i to serve as the input of graph neural networks:

$$x_i^{(0)} = \phi(W_I \cdot v_i + b_I) \quad (6)$$

where ϕ is leaky-relu, W_I and b_I are learnable parameters. We then propagate entity messages and aggregate them with gated R-GCN. For the l -th layer of gated R-GCN:

$$u_i^{(l)} = \sum_{r \in R} \frac{1}{|N_r(e_i)|} \sum_{j \in N_r(e_i)} f_r(x_j^{(l-1)}) + f_s(x_i^{(l-1)}) \quad (7)$$

where f_s and f_r are parameterized linear layers for self loops and edges of relation r , $u_i^{(l)}$ is the hidden representation for entity e_i at layer l . We then calculate gate levels:

$$g_i^{(l)} = \sigma(W_G \cdot [u_i^{(l)}, x_i^{(l-1)}] + b_G) \quad (8)$$

where W_G and b_G are learnable parameters, $\sigma(\cdot)$ denotes the sigmoid function and $[\cdot, \cdot]$ denotes the concatenation operation. We then apply the gate mechanism to $u_i^{(l)}$ and $x_i^{(l-1)}$:

$$x_i^{(l)} = \tanh(u_i^{(l)}) \odot g_i^{(l)} + x_i^{(l-1)} \odot (1 - g_i^{(l)}) \quad (9)$$

where \odot is the Hadamard product. After L layer(s) of gated R-GCN, we obtain node features $\{x_1^{(L)}, \dots, x_n^{(L)}\}$ and the nodes representing political actors are extracted as learned representations.

3.4 Model Training

We propose to train PAR with a combination of three training objectives, which aligns representation learning with expert knowledge, ensures stance consistency and simulates the echo chamber phenomenon. The overall loss function of PAR is as follows:

$$L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \lambda_4 \sum_{w \in \theta} \|w\|_2^2 \quad (10)$$

where λ_i is the weight of loss L_i and θ are learnable parameters in PAR. We then present the motivation and detail of each loss function L_1 , L_2 and L_3 .

3.4.1 Objective 1: Expert Knowledge

The expert knowledge objective aims to align the learned representations with expert knowledge from political think tanks. We use the learned representations of political actors to predict their liberal and conservative stances, which are adapted from AFL-CIO and Heritage Action. Specifically:

$$\begin{aligned} l_i &= \text{softmax}(W_L \cdot x_i^{(L)} + b_L) \\ c_i &= \text{softmax}(W_C \cdot x_i^{(L)} + b_C) \end{aligned} \quad (11)$$

where l_i and c_i are predicted stances towards liberal and conservative values, W_L , b_L , W_C and b_C are learnable parameters. Let E_L and E_C be the training set of AFL-CIO and Heritage Action scores, \hat{l}_i and \hat{c}_i denote the ground-truth among D possible labels. We calculate the expert knowledge loss:

$$L_1 = - \sum_{e_i \in E_L} \sum_{d=1}^D \hat{l}_{id} \log(l_{id}) - \sum_{e_i \in E_C} \sum_{d=1}^D \hat{c}_{id} \log(c_{id}) \quad (12)$$

L_1 enables PAR to align learned representations with expert knowledge from political think tanks.

3.4.2 Objective 2: Stance Consistency

The stance consistency objective is motivated by the fact that liberalism and conservatism are opposite ideologies, thus individuals often take inversely correlated stances towards them. We firstly speculate entities' stance towards the opposite ideology by taking the opposite of the predicted stance:

$$\tilde{l}_i = \psi(D - \text{argmax}(l_i)), \quad \tilde{c}_i = \psi(D - \text{argmax}(c_i)) \quad (13)$$

where ψ is the one-hot encoder, $\text{argmax}(\cdot)$ calculates the vector index with the largest value, D is the number of stance labels, \tilde{l}_i and \tilde{c}_i are labels derived with stance consistency. We calculate the loss function L_2 measuring stance consistency:

$$L_2 = - \sum_{e_i \in E} \sum_{d=1}^D (\tilde{l}_{id} \log(l_{id}) + \tilde{c}_{id} \log(c_{id})) \quad (14)$$

where $E = E_L \cap E_C$. As a result, the loss function L_2 enables PAR to ensure ideological stance consistency among political actors.

3.4.3 Objective 3: Echo Chamber

The echo chamber objective is motivated by the echo chamber phenomenon (Jamieson and Cappella, 2008; Barberá et al., 2015), where social entities tend to reinforce their narratives by forming

small and closely connected interaction circles. We simulate echo chambers by assuming that neighboring nodes in the HIN have similar representations while non-neighboring nodes have different representations. We firstly define the positive and negative neighborhood of entity e_i :

$$\begin{aligned} P_{e_i} &= \{e \mid \exists r \in R \text{ s.t. } e \in N_r(e_i)\} \\ N_{e_i} &= \{e \mid \forall r \in R \text{ s.t. } e \notin N_r(e_i)\} \end{aligned} \quad (15)$$

We then calculate the echo chamber loss:

$$\begin{aligned} L_3 &= - \sum_{e_i \in E} \sum_{e_j \in P_{e_i}} \log(\sigma(x_i^{(L)^T} x_j^{(L)})) \\ &\quad + Q \cdot \sum_{e_i \in E} \sum_{e_j \in N_{e_i}} \log(\sigma(-x_i^{(L)^T} x_j^{(L)})) \end{aligned} \quad (16)$$

where Q denotes the weight for negative samples. L_3 enables PAR to model the echo chamber phenomenon in real-world socio-economic systems.

4 Experiment

After learning political actor representations with PAR, we study whether they are effective in computational political science and reveal real-world political behavior. Specifically, we test out PAR on political perspective detection and roll call vote prediction, two political text understanding tasks that emphasizes political actor modeling. We leverage PAR in these tasks to examine whether it could contribute to the better understanding of political text. We then study the learned representations of PAR and whether they provide new insights into real-world politics.

4.1 Political Perspective Detection

Political perspective detection aims to detect stances in text such as public statements and news articles, which generally mention many political actors to provide context and present arguments. Existing methods model political actors with masked entity models (Li and Goldwasser, 2021) and knowledge graph embedding techniques (Feng et al., 2021). We examine whether PAR is more effective than these models and consequently improve political perspective detection.

4.1.1 Datasets

We follow previous works (Li and Goldwasser, 2021; Feng et al., 2021) and adopt two political perspective detection benchmarks: SemEval and Allsides. SemEval (Kiesel et al., 2019) aims to

Method	Setting	SemEval		AllSides	
		Acc	MaF	Acc	MaF
CNN	GloVe	79.63	N/A	N/A	N/A
	ELMo	84.04	N/A	N/A	N/A
HLSTM	GloVe	81.58	N/A	N/A	N/A
	ELMo	83.28	N/A	N/A	N/A
	Embed	81.71	N/A	76.45	74.95
	Output	81.25	N/A	76.66	75.39
BERT	base	84.03	82.60	81.55	80.13
MAN	GloVe	81.58	79.29	78.29	76.96
	ELMo	84.66	83.09	81.41	80.44
	Ensemble	86.21	84.33	85.00	84.25
KGAP	TransE	89.56	84.94	86.02	85.52
	TransR	88.54	83.45	85.15	84.61
	DistMult	88.51	83.63	84.47	83.90
	Hole	88.85	83.68	84.78	84.24
	RotatE	88.84	84.04	85.61	85.11
KGAP	PAR	91.30	87.78	86.81	86.33

Table 1: Political perspective detection performance on two benchmark datasets. Acc and MaF denote accuracy and macro-averaged F1-score. N/A indicates that the result is not reported in previous works.

identify whether a news article follows hyperpartisan argumentation. We follow the 10-fold cross validation setting and the exact same folds established in [Li and Goldwasser \(2021\)](#) so that the results are directly comparable. Allsides ([Li and Goldwasser, 2019](#)) provides left, center, or right labels for three-way classification. We follow the 3-fold cross validation setting and the exact same folds established in [Li and Goldwasser \(2019\)](#) so that the results are directly comparable.

4.1.2 Baselines

We compare PAR with competitive baselines:

- **CNN** ([Jiang et al., 2019](#)) achieves the best performance in the SemEval 2019 Task 4 contest ([Kiesel et al., 2019](#)). It uses convolutional neural networks along with word embeddings **GloVe** ([Jiang et al., 2019](#)) and **ELMo** ([Peters et al., 2018](#)) for perspective detection.
- **HLSTM** ([Yang et al., 2016](#)) is short for hierarchical long short-term memory networks. [Li and Goldwasser \(2019\)](#) combines it with **GloVe** and **ELMo** word embeddings. [Li and Goldwasser \(2021\)](#) leverages Wikipedia2Vec ([Yamada et al., 2020](#)) and masked entity models while using different concatenation strategies (**HLSTM_Emb** and **HLSTM_Output**).

Method	Setting	
	random	time-based
majority	77.48	77.40
ideal-point-wf	85.37	N/A
ideal-point-tfidf	86.48	N/A
ideal-vector	87.35	N/A
CNN	87.28	81.97
CNN+meta	88.02	84.30
LSTM+GCN	88.41	85.82
Vote	90.22	89.76
RoBERTa	87.59	87.56
TransE	82.70	80.06
PAR	90.33	89.92

Table 2: Roll call vote prediction performance (accuracy) with two experiment settings. N/A indicates that the result is not reported in previous works.

- **BERT** ([Devlin et al., 2019](#)) is fine-tuned on the task of political perspective detection.
- **MAN** ([Li and Goldwasser, 2021](#)) leverages social and linguistic information for pre-training a BERT-based model and fine-tune on the task of political perspective detection.
- **KGAP** ([Feng et al., 2021](#)) models legislators with RoBERTa and knowledge graph embeddings: **TransE** ([Bordes et al., 2013](#)), **TransR** ([Lin et al., 2015](#)), **DistMult** ([Yang et al., 2015](#)), **Hole** ([Nickel et al., 2016](#)), and **RotatE** ([Sun et al., 2018b](#)). It then constructs document graphs with text and legislators as nodes and uses GNNs for stance detection.
- **KGAP_PAR**: To examine whether PAR is effective in political perspective detection, we replace the knowledge graph embeddings in **KGAP** with political actor representations learned with PAR.

4.1.3 Results

We evaluate PAR and baselines and present model performance in Table 1, which demonstrates that PAR achieves state-of-the-art performance on both datasets. The fact that PAR outperforms MAN and KCD, two methods that also take political actors into account, shows that PAR learns high-quality representations that results in performance gains.

4.2 Roll Call Vote Prediction

Roll call vote prediction aims to predict whether a legislator will vote in favor or against a specific legislation. Ideal points ([Gerrish and Blei, 2011](#)), ideal vectors ([Kraft et al., 2016](#)), and [Mou et al. \(2021\)](#)

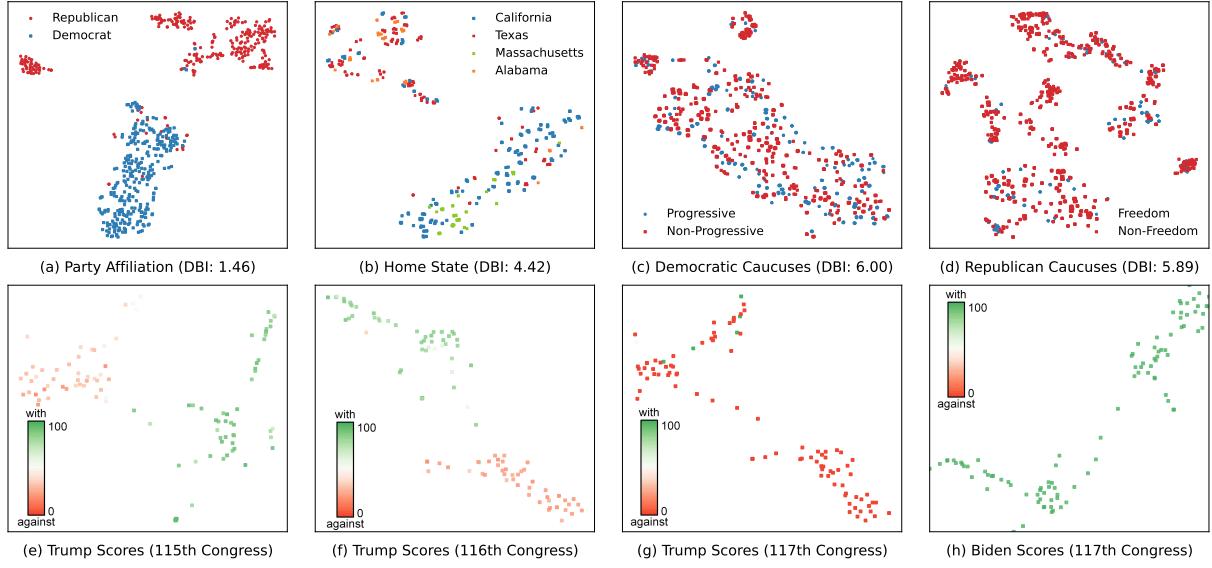


Figure 3: Using t-sne to visualize learned representations of political actors. DBI denotes the Davies-Bouldin Index.

have been proposed to model ideological preferences. We examine whether PAR is more effective than these models in roll call vote prediction.

4.2.1 Datasets

We adopt the datasets and settings proposed in Mou et al. (2021) to evaluate PAR and competitive baselines. Specifically, for the random setting, voting records in the 114th and 115th congresses are randomly split into 6:2:2 for training, validation, and testing. For the time-based setting, the voting records in the 114th congress is split into 8:2 for training and validation, while the voting records in the 115th congress is used for testing.

4.2.2 Baselines

We compare PAR with competitive baselines:

- **majority** predicts all votes as *yea*.
- **ideal-point-wf** and **ideal-point-tfidf** (Gerrish and Blei, 2011) adopt word frequency and TFIDF of legislation text as features and leverage the ideal point model for vote prediction.
- **ideal-vector** (Kraft et al., 2016) learns distributed representations with legislator and bill text.
- **CNN** (Kornilova et al., 2018) encodes bill text with CNNs. On top of that, **CNN+meta** introduces metadata of bill sponsorship to the process.

- **LSTM+GCN** (Yang et al., 2021) leverages LSTM and GCN to jointly update representations of legislations and legislators.
- **Vote** (Mou et al., 2021) proposes to align statements on social networks with voting records.
- **RoBERTa** and **TransE** use RoBERTa (Liu et al., 2019) encoding of legislator description on Wikipedia or TransE (Bordes et al., 2013) embeddings for legislator representation learning. They are then concatenated with RoBERTa encoded bill text for vote prediction.
- **PAR** concatenates political actor representations learned with PAR and RoBERTa encoded legislator text for roll call vote prediction.

4.2.3 Results

We evaluate PAR and competitive baselines on roll call vote prediction and present model performance in Table 2. PAR achieves state-of-the-art performance, outperforming existing baselines that model political actors in different ways. As a result, PAR learns high-quality representations of political actors that provide political knowledge and augment the vote prediction process.

4.3 Political Findings of PAR

PAR learns political actor representations with social context and expert knowledge, which has been proven effective in political perspective detection and roll call vote prediction. We further examine

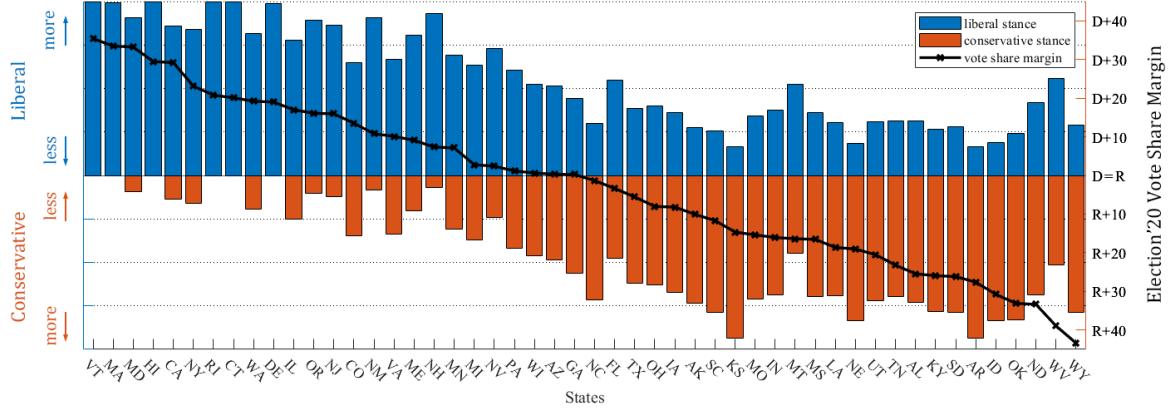


Figure 4: State stances predicted by PAR compared to vote shares in the 2020 U.S. presidential election.

whether PAR provides new insights into the political behavior of legislators, states, and governors.

4.3.1 Legislators and PAR

To examine whether legislator representations learned with PAR align well with different social and political factors, we adopt t-sne (Maaten and Hinton, 2008) to illustrate our learned representations of political actors in Figure 3.

Social context. Figure 3 (a) and (b) illustrate the correlation between learned representations and social context factors such as party affiliation and home state. It is illustrated that legislators from the same party and the same home state tends to be similar in the representation space. In addition, we calculate the DBI scores (Davies and Bouldin, 1979) to quantitatively analyze the collocation. In conclusion, the learned representations successfully reflect these social context information.

Congressional caucus. Figure 3 (c) and (d) demonstrate the correlation between learned representations and major congressional caucuses in both parties. The progressive caucus in the democratic party and the freedom caucus in the republican party are often viewed as more ideological wings of the party. Both the illustration and the DBI scores suggest little collocation among different caucuses. As a result, PAR reveals that despite widening ideological gaps between party factions (Cohen et al., 2016), inter-party differences still outweigh intra-party differences in contemporary U.S. politics.

Voting records. Figure 3 (e), (f), (g) and (h) illustrate how often does a legislator vote for or against the sitting president. We retrieve this information

from FiveThirtyEight⁵ and illustrate voting records with color gradients. "Trump scores" and "Biden scores" indicate what percentage out of all votes did a legislator vote with or against the official stance of the president. As a result, our learned representations of legislators in the 115th and 116th congress correlate well with their voting records, while the 117th congress might have not hold enough votes for an accurate categorization⁶.

4.3.2 States and PAR

Political commentary often uses "blue state", "red state", and "swing state" to describe the ideological preference of states in the U.S. We examine the ideological scores of states learned by PAR (l_i and c_i in Equation (11)) and compare them with results in the 2020 U.S. presidential election in Figure 4. It is illustrated that PAR stance predictions highly correlate with the 2020 presidential election results. In addition, PAR reveals that Pennsylvania and North Carolina, two traditional swing states, are actually more partisan than expected. PAR also suggests that Georgia is the most electorally competitive state in the United States and we will continue to monitor this conclusion in future elections.

4.3.3 Governors and PAR

Political experts typically study and evaluate the stances of presidents and federal legislators, while state-level officials such as governors are also essential in governance and policy making (Beyle, 1988). PAR complements the understanding of state-level politics by learning ideological scores for governors. Specifically, we use l_i and c_i in

⁵<https://fivethirtyeight.com/>

⁶Voting records of the 117th congress is collected in June 2021, only five months into its term.

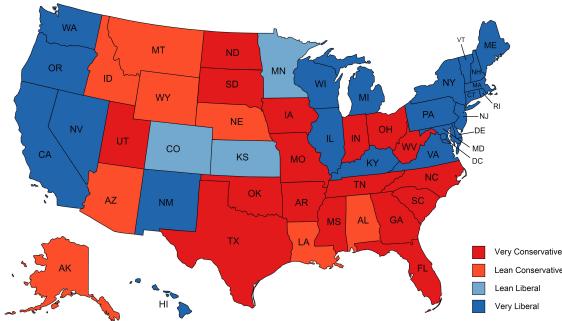


Figure 5: Governor stances predicted by PAR.

Equation (11) to evaluate the ideological position of governors and present PAR’s predictions of governor stances in all 50 U.S. states in Figure 5⁷. It is no surprise that governors from partisan strongholds such as California and Utah hold firm stances. Conventional wisdom often assumes that in order to win “swing states” or electorally challenging races, one has to compromise their ideological stances and lean to the center. However, PAR reveals exceptions to this rule, such as Andy Beshear (D-KY) and Ron DeSantis (R-FL), which is also suggested in political commentary (Jr., 2020; Graham, 2022).

5 Conclusion

In this paper, we present PAR, a framework to learn representations of political actors with social context and expert knowledge. Specifically, we retrieve social context information from Wikipedia and expert knowledge from political think tanks, construct a HIN to model legislators and learn representations with gated R-GCNs and three training objectives. Extensive experiments demonstrate that PAR advances the state-of-the-art in political perspective detection and roll call vote prediction. PAR further provides novel insights into political actors through its representation learning process.

6 Limitations

- PAR proposes to learn representations for political actors with the help of social context and expert knowledge. Though it achieved great performance on several tasks and provided interesting political insights, it might reinforce political stereotypes, such as people from “red states” are often assumed to be more conservative. We leave for future work how to mitigate the potential bias in political actor representations.

⁷This figure is created with the help of mapchart.net.

- In this paper, we focus on political actors in the United States instead of other countries. However, our proposed framework could be easily extended to other scenarios by leveraging the Wikipedia pages of political actors for social context and political think tanks in other countries for expert knowledge.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2020AAA0108800), National Nature Science Foundation of China (No. 62192781, No. 62272374, No. 61872287, No. 62250009, No. 62137002), Innovative Research Group of the National Natural Science Foundation of China (61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Project of China Knowledge Center for Engineering Science and Technology and Project of Chinese academy of engineering “The Online and Offline Mixed Educational Service System for ‘The Belt and Road’ Training in MOOC China”.

We would like to thank the reviewers and area chair for their constructive feedback. We would also like to thank all LUD lab members for our collaborative research environment.

References

- Kenan Alkiek, Bohan Zhang, and David Jurgens. 2022. Classification without (proper) representation: Political heterogeneity in social media and its implications for classification and behavioral analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland.
- Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885.
- Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.
- David Bamman, Jacob Eisenstein, and Tyler Schnoebel. 2012. Gender in twitter: Styles, stances, and social networks.
- Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. 2015. Tweeting from left to right: Is online political communication more

- than an echo chamber? *Psychological science*, 26(10):1531–1542.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Thad L Beyle. 1988. The governor as innovator in the federal system. *Publius: The Journal of Federalism*, 18(3):131–152.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- Xavier Bresson and Thomas Laurent. 2017. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*.
- Joshua Clinton, Simon Jackman, and Douglas Rivers. 2004. The statistical analysis of roll call data. *American Political Science Review*, 98(2):355–370.
- Marty Cohen, David Karol, Hans Noel, and John Zaller. 2016. Party versus faction in the reformed presidential nominating system. *PS: Political Science & Politics*, 49(4):701–708.
- Kareem Darwish, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. 2020. Unsupervised user stance detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 141–152.
- David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.
- Maryam Davoodi, Eric Waltenburg, and Dan Goldwasser. 2022. Modeling U.S. 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)*, Dublin, Ireland.
- Erenay Dayanik, Andre Blessing, Nico Blokker, Sebastian Haunss, Jonas Kuhn, Gabriella Lapesa, and Sebastian Pado. 2022. Improving neural political statement classification with class hierarchical information. In *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota.
- Michael Evans, Wayne McIntosh, Jimmy Lin, and Cynthia Cates. 2007. Recounting the courts? applying automated content analysis to enhance empirical legal research. *Journal of Empirical Legal Studies*, 4(4):1007–1039.
- William Falcon and The PyTorch Lightning team. 2019. [PyTorch Lightning](#).
- Shangbin Feng, Zilong Chen, Wenqian Zhang, Qingyao Li, Qinghua Zheng, Xiaojun Chang, and Minnan Luo. 2021. Kgap: Knowledge graph augmented political perspective detection in news media. *arXiv preprint arXiv:2108.03861*.
- Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.
- Michael Freedmen. 2006. Ideology and political theory. *Journal of Political Ideologies*, 11(1):3–22.
- Sean M Gerrish and David M Blei. 2011. Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*.
- David A. Graham. 2022. How a purple state got a bright red sheen. *The Atlantic*. Accessed: 2022-05-24.
- Yupeng Gu, Yizhou Sun, Ning Jiang, Bingyu Wang, and Ting Chen. 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*, pages 183–192.
- William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035.
- Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the sixth international joint conference on natural language processing*, pages 1348–1356.
- Benjamin D Horne, Sara Khedr, and Sibel Adali. 2018. Sampling the news producers: A large news and feature data set for the study of the complex media landscape. In *Twelfth International AAAI Conference on Web and Social Media*.
- Kathleen Hall Jamieson and Joseph N Cappella. 2008. *Echo chamber: Rush Limbaugh and the conservative media establishment*. Oxford University Press.
- Ye Jiang, Johann Petrak, Xingyi Song, Kalina Bontcheva, and Diana Maynard. 2019. Team bertha von suttner at semeval-2019 task 4: Hyperpartisan news detection using elmo sentence representation convolutional network. In *Proceedings of the*

- 13th International Workshop on Semantic Evaluation, pages 840–844.
- Kristen Johnson and Dan Goldwasser. 2016. Identifying stance by analyzing political discourse on twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 66–75.
- Perry Bacon Jr. 2020. Analysis: The surprising boldness of andy beshear. *WFPL*. Accessed: 2022-05-24.
- Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4725–4735.
- Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839.
- Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Anastassia Kornilova, Daniel Argyle, and Vladimir Eidelman. 2018. Party matters: Enhancing legislative embeddings with author attributes for vote prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–515.
- Peter Kraft, Hirsh Jain, and Alexander M Rush. 2016. An embedding model for predicting roll-call votes. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2066–2070.
- Dilek Küçük and Fazli Can. 2018. Stance detection on tweets: An svm-based approach. *arXiv preprint arXiv:1803.08910*.
- Chang Li and Dan Goldwasser. 2019. Encoding social information with graph convolutional networks for political perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2594–2604.
- Chang Li and Dan Goldwasser. 2021. Using social and linguistic information to adapt pretrained representations for political perspective identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4569–4579.
- Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. P-stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yujian Liu, Xinliang Frederick Zhang, David Weigman, Nicholas Beauchamp, and Lu Wang. 2022. POLITICS: Pretraining with same-story article comparison for ideology prediction and stance detection. In *Findings of the Association for Computational Linguistics: NAACL 2022*, Seattle, United States.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.
- Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1–23.
- Xinyi Mou, Zhongyu Wei, Lei Chen, Shangyi Ning, Yancheng He, Changjian Jiang, and Xuan-Jing Huang. 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)*, pages 1236–1246.
- Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana.
- Anushka Prakash and Harish Tayyar Madabushi. 2020. Incorporating count-based features into pre-trained models for improved stance detection. In *Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, Barcelona, Spain (Online).
- Rajkumar Pujari and Dan Goldwasser. 2021. Understanding politics via contextualized discourse processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic.
- Indira Sen, Fabian Flöck, and Claudia Wagner. 2020. On the reliability and validity of detecting approval of political actors in tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.
- Yunsheng Shi, Zhengjie Huang, Wenjin Wang, Hui Zhong, Shikun Feng, and Yu Sun. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.
- Barea Sinno, Bernardo Oviedo, Katherine Atwell, Malthe Alikhani, and Junyi Jessy Li. 2022. Political ideology and polarization: A multi-dimensional approach. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States.
- Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018a. Stance detection with hierarchical attention network. In *Proceedings of the 27th international conference on computational linguistics*, pages 2399–2409.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018b. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.
- Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia. Association for Computational Linguistics.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.
- Danae Sánchez Villegas, Saeid Mokaram, and Nikolaos Aletras. 2021. Analyzing online political advertisements. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3669–3680.
- Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 186–196.
- Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. 2016. pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 384–388.
- John Wilkerson and Andreu Casas. 2017. Large-scale computerized text analysis in political science: Opportunities and challenges. *Annual Review of Political Science*, 20:529–544.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrette Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online.
- Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jian-feng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.
- Yuqiao Yang, Xiaoqiang Lin, Geng Lin, Zengfeng Huang, Changjian Jiang, and Zhongyu Wei. 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*, pages 1424–1430.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Wenqian Zhang, Shangbin Feng, Zilong Chen, Zhenyu Lei, Jundong Li, and Minnan Luo. 2022. KCD: Knowledge walks and textual cues enhanced political perspective detection in news media. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States.

Method	Text	Graph	Acc	MaF	MiF
Linear BoW	✓		68.49	40.00	68.53
Bias Features	✓		47.26	20.08	47.10
Glove	✓		52.05	26.94	52.01
RoBERTa	✓		71.92	49.70	71.87
LongFormer	✓		68.49	42.27	68.56
GCN	✓	✓	74.66	54.16	74.46
GAT	✓	✓	78.08	55.82	78.17
GraphSAGE	✓	✓	75.34	51.39	75.43
TransformerConv	✓	✓	77.40	55.63	77.48
ResGatedConv	✓	✓	76.03	54.31	75.97
Ours	✓	✓	80.82	60.37	80.89

Table 3: Our model’s performance on the expert knowledge prediction task compared to various text and graph analysis baselines. Acc, MaF and MiF denote accuracy, macro and micro-averaged F1-score.

A Expert Knowledge Prediction

We propose to train PAR with three objectives, the first one L_1 being expert knowledge prediction, where the model learns to predict how liberal or conservative a given legislator is. To examine whether the PAR architecture successfully conducts expert knowledge prediction, we compare PAR with various text (Pedregosa et al., 2011; Horne et al., 2018; Pennington et al., 2014; Liu et al., 2019; Beltagy et al., 2020) and graph (Kipf and Welling, 2016; Veličković et al., 2018; Hamilton et al., 2017; Shi et al., 2020; Bresson and Laurent, 2017) analysis baselines. Table 3 presents model performance on the expert knowledge task. It is demonstrated that PAR achieves the best performance compared to various text and graph analysis baselines, suggesting our learned representations successfully reflect expert knowledge from political think tanks. Apart from that, graph-based models generally outperform text-based methods, which suggests that the constructed HIN is essential in modeling political actors.

B Ablation Study

PAR aims to learn representations of political actors with social context and expert knowledge as well as three training objectives. We conduct ablation study to examine their effect in the representation learning process. We report model performance on the expert knowledge prediction task and follow the same dataset splits as in Table 3.

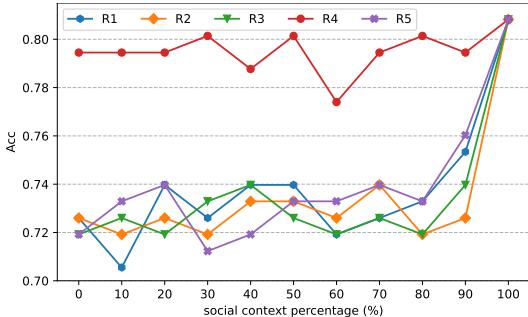


Figure 6: Ablation study of social context information. $R1$ to $R5$ follows that in Section 3.2.2.

B.1 Social Context

We use five types of heterogeneous relations $R1$ to $R5$ to connect different entities based on their social context. We randomly and gradually remove five types of social context edges in the constructed HIN and report model performance in Figure 6. It is illustrated that all relations but $R4$ (Time in Office) significantly contributes to the overall performance. Besides, Figure 6 illustrates a great gap between 90% and 100% edges, suggesting the importance of a complete HIN structure.

B.2 Expert Knowledge

We learn legislator representations with the help of two political think tanks: AFL-CIO and Heritage Action. We retrieve their evaluation of political actors and construct the expert knowledge objective for training. To examine the effect of expert knowledge in our proposed approach, we gradually remove expert knowledge labels in L_1 and report model performance in Figure 7. It is illustrated that our performance drops with partial expert knowledge from either AFL-CIO or Heritage Action, which indicates that expert knowledge is essential in the representation learning process.

B.3 Training Objectives

We propose to train our framework with three objectives. To examine their effect, we train our method with different combinations of L_1 , L_2 and L_3 and report performance in Table 4. Our model performs best with all three training objectives, proving the effectiveness of our loss function design. We further study the influence of loss weights λ_1 , λ_2 , λ_3 and λ_4 . We fix $\lambda_1 = 1$ and $\lambda_4 = 10^{-5}$, present model performance under different settings of loss weights for auxiliary tasks λ_2 and λ_3 in Figure 8. It is illustrated that $0.2 \leq \lambda_2 \leq 0.3$ and

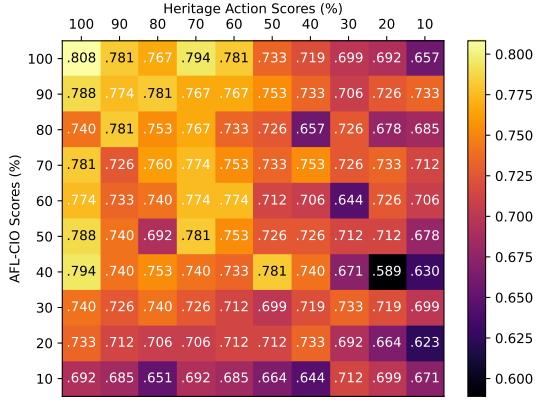


Figure 7: Ablation study of expert knowledge. We report accuracy on the expert knowledge prediction task.

Loss Function(s)	Acc	MaF	MiF
L_1 only	78.08	56.43	78.17
L_1 and L_2	79.45	55.97	79.52
L_1 and L_3	76.03	51.91	76.11
L_1, L_2 , and L_3	80.82	60.37	80.89

Table 4: Ablation study of three training objectives.

$0.01 \leq \lambda_3 \leq 0.1$ would generally lead to an effective balance of three different training objectives.

B.4 Graph Learning

We adopted five relations $R1$ to $R5$ to connect eight types of entities $N1$ to $N8$, so that the graph is heterogeneous. To examine whether the graph heterogeneity contributes to model performance, we substitute gated R-GCNs with homogeneous GNNs such as GCN, GAT, and GraphSAGE. Table 5 shows mixed results, where PAR performs best with gated R-GCNs while R-GCN does not outperform GraphSAGE.

C Error Analysis

We manually examined the part of the results in the roll call vote prediction task. Among the (legislator, bill) pairs where PAR made a wrong prediction, it is often the case that the legislator has voted across party lines. This suggests that more information about these legislators are required to achieve more fine-grained analysis on these borderline cases.

D Reproducibility Details

In this section, we provide additional details to facilitate reproducing our results and findings. We submit data and code as supplementary material

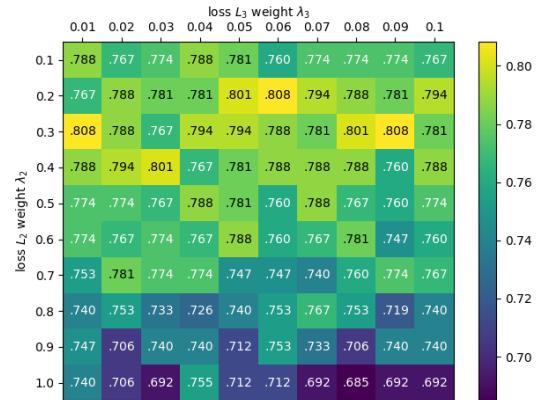


Figure 8: Model accuracy with different loss weights.

GNN operator	Het.	Acc	MaF	MiF
GCN	✗	76.03	58.31	78.08
GAT	✗	77.40	59.01	78.85
SAGE	✗	78.77	58.04	78.81
R-GCN	✓	78.08	55.61	78.15
Gated R-GCN	✓	80.82	60.37	80.89

Table 5: Model performance with different GNNs. We adopt gated R-GCN and achieves the best performance. Het. denotes whether it supports heterogeneous graphs.

and commit to make them publicly available upon acceptance to facilitate reproduction.

D.1 Hyperparameters

We present the hyperparameter settings of our proposed approach in Table 6. These hyperparameters are manually tuned. We follow these settings throughout the paper unless stated otherwise.

D.2 Implementation

We use pytorch (Paszke et al., 2019), pytorch lightning (Falcon and The PyTorch Lightning team, 2019), torch geometric (Fey and Lenssen, 2019), and the transformers library (Wolf et al., 2020) for implementation. All implemented codes are available as supplementary material.

D.3 Computation

Our proposed approach has a total of 2.0M learnable parameters with hyperparameters in Table 6. We train it on a Titan X GPU with 12GB memory. It takes approximately 0.6 GPU hours for training with hyperparameters in Table 6.

Hyperparameter	Value	Hyperparameter	Value
RoBERTa size	768	GNN size	512
optimizer	Adam	learning rate	$1e - 3$
batch size	64	max epochs	100
L	2	ϕ	ReLU
Q	-0.1	#negative sample	2
λ_1	0.01	λ_2	0.2
λ_3	1	λ_4	$1e - 5$

Table 6: Hyperparameters of our proposed approach.

E Experiment Details

E.1 Political Perspective Detection

To examine whether our learned representations of political actors would benefit perspective analysis, we replace TransE in Feng et al. (2021) with our learned representations. We use the GRGCN setting in Feng et al. (2021) as model backbone. We maintain the same evaluation settings to ensure a fair comparison and highlight the effectiveness of our learned representations compared to TransE.

E.2 Roll Call Vote Prediction

We make our best effort to maintain the same experiment settings as Mou et al. (2021) while there might be minor differences. For the *random* setting, we conduct roll call vote prediction for legislators in the 114th and 115th congress. We follow the same 6:2:2 split. For the *time-based* setting, we use the 114th congress as training and validation set and the 115th congress as test set.

E.3 Expert Knowledge Prediction

We collect expert knowledge about legislators from two political think tanks, which assign a continuous score s from 0 to 1 to indicate how well a political actor aligns with their agenda. We construct a classification task from expert knowledge by creating five labels: strongly favor ($0.9 \leq s \leq 1$), favor ($0.75 \leq s \leq 0.9$), neutral ($0.25 \leq s \leq 0.75$), oppose ($0.1 \leq s \leq 0.25$), and strongly oppose ($0 \leq s \leq 0.1$). In this way, we adapt from expert knowledge to derive liberal and conservative labels for legislators and thus $D = 5$ for Equation 12. We use 7:2:1 to partition them into training, validation and test sets. We calculate evaluation metrics on the liberal and conservative set separately, and present the harmonic mean of metrics. In this way, the presented results accurately and comprehensively reflect how our approach and baselines perform on both political think tanks. For text-based baselines,

we encode Wikipedia summaries of entities with these methods and predict their stances with two fully connected layers. For graph-based baselines, we train them with the constructed HIN and the expert knowledge training objective.

E.4 Figure 4

As detailed in Section E.3, each entity in PAR’s HIN has two set of scores: liberal scores (l_0, l_1, l_2, l_3, l_4) and conservative scores (c_0, c_1, c_2, c_3, c_4), denoting the probability that the entity is strongly against, against, neutral, favor, or strongly favor liberal/conservative values. We use $\sum_{i=0}^4 i \cdot l_i$ and $\sum_{i=0}^4 i \cdot c_i$ to obtain continuous values of state stances and let them be the height of blue and red bars in Figure 4.

E.5 Figure 5

We use the liberal scores of governors learned by PAR (l_0, l_1, l_2, l_3, l_4) to infer their ideological stances. Specifically, we use "very conservative", "lean conservative", "lean liberal", and "very liberal" to denote that $\text{argmax}(l_0, l_1, l_2, l_3, l_4) = 0, 1, 3, 4$ respectively. In addition, there are no governor that has $\text{argmax}(l_0, l_1, l_2, l_3, l_4) = 2$ so "neutral" is omitted from the figure.