MGM: Global Understanding of Audience Overlap Graphs for Predicting the Factuality and the Bias of News Media

Anonymous ACL submission

Abstract

In the current era of rapidly growing digital data, evaluating the political bias and factuality of news media outlets has become more important for seeking reliable information online. In this work, we study the classification problem of profiling the news media from the lens of political bias and factuality. Traditional profiling methods, such as Graph Neural Networks (GNNs) and Pre-trained Language Models (PLMs), have shown promising results, but they face notable challenges. GNNs often struggle with media graphs containing disconnected components and insufficient labels, while PLMs focus solely on textual features, causing them to overlook the complex relationships between entities. To address these limitations, we propose MediaGraph-Mind (MGM), an effective solution within a variational Expectation-Maximization (EM) framework. Instead of relying on limited neighboring nodes, MGM leverages features, structural patterns, and label information from globally similar nodes. Such a framework not only enables GNNs to capture long-range dependencies for learning expressive node representations but also enhances PLMs by integrating structural information and therefore improving the performance of both models. Our extensive experiments on a standard dataset demonstrate the effectiveness of our proposed framework.

1 Introduction

The rise of the Internet has offered many opportunities to publish information and to express opinions (Nakov et al., 2021). At the same time, it has accelerated the spread of misinformation and disinformation online (Fairbanks et al., 2018). Initially, efforts focused on verifying individual claims, but it soon became evident that assessing the factuality of the news sources themselves was equally important (Baly et al., 2020b).

Early studies on automatic media profiling relied solely on text characteristics (Battaglia et al., 2018; Pérez-Rosas et al., 2017), which has proven particularly challenging (Baly et al., 2018, 2020b). The complexity is heightened when the text features contain indeterminate noise, leading to classification errors. Moreover, traditional methods struggle to capture the intricate relationships between entities, such as articles, audiences, and media outlets. Graph Neural Networks (GNNs) have emerged as an effective framework to capture and to model these relationships, reducing the dependence on individual text features and improving media profiling (Panayotov et al., 2022; Mehta et al., 2022).

Despite their advantages, GNNs also face challenges due to the graph being disconnected and due to label sparsity. *Disconnected components* hinder GNNs from capturing long-range dependencies, thus limiting their ability to learn expressive node representations (Bodnar et al., 2021; Dai et al., 2021, 2022). To address these issues, various memory-based GNNs have been proposed (Kang, 2021; Zhang et al., 2022; Ma et al., 2022), which capture long-range dependencies using external memory modules that store *global information*, i.e., features and structural patterns throughout the graph. However, these approaches require substantial memory to store the embeddings of all nodes.

To adress these challenges, we propose MGM, a method with a variational Expectation-Maximization (EM) framework that enhances existing GNNs to capture and leverage global information in media graphs. MGM captures both local and global patterns, node features, and labels from global similar nodes for improved performance. Unlike Graph Attention Networks (Veličković et al., 2018), which focus on similar nodes in local neighbors, MGM uses an external memory module to store precomputed node representations, thereby reducing compute costs (Fey et al., 2021) and enabling efficient retrieval of node embeddings. Moreover, MGM reduces memory requirements by focusing on a small set of candidate nodes, guided

by a distribution over the training nodes sampled from a Dirichlet prior (He et al., 2020).

The results show that MGM significantly improves the performance of the baseline GNNs, achieving a 10% improvement in all evaluation measures on the Media Bias/Fact Check (MBFC) [1] data featured in the ACL-2020 (Baly et al., 2020c) and the EMNLP-2018 (Baly et al., 2018) datasets. Despite the lack of rich features in the graph, we enhance the dataset by scraping *Articles* and *Wikipedia* for ACL-2020. Pre-trained language models (PLMs) such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), and DeBERTaV3 (He et al., 2021) are fine-tuned to predict political bias and factuality. Where media data are inaccessible, MGM's representation-based probabilities fill the gap. Moreover, integrating MGM's probabilities with LM's enhances the performance for both tasks. Our contributions can be summarised as follows:

- We introduce MGM, an effective approach that leverages global information to enhance the expressiveness of existing GNNs for reliable news media profiling.
- In MGM, we use external memory to efficiently retrieve globally similar nodes and reduce memory requirements by learning a sparse distribution with a Dirichlet prior.
- We illustrate that MGM consistently outperforms vanilla GNNs for the detection of factuality and political bias across all baselines.
- We validated that integrating the MGM features with the PLMs enhances performance and yields state-of-the-art results.

2 Related Work

2.1 Factuality and Political Bias of Media

Early research on news media profiling focused on textual content analysis (Afroz et al., 2012; Battaglia et al., 2018; Pérez-Rosas et al., 2017; Conroy et al., 2015). To improve the performance, subsequent research added contextual information (Baly et al., 2020c; Hounsel et al., 2020; Castelo et al., 2019; Fairbanks et al., 2018), including the nuances of multimedia production (Huh et al., 2018), the associated infrastructure (Hounsel et al., 2020), and the social context (Baly et al., 2020c). Guo et al. (2022) used BERT (Devlin et al., 2019) to model the linguistic political bias in news articles. Fan et al. (2019) used annotated media from Budak et al. (2016), analyzing articles for political bias using distant supervision. Various methods measured political bias, including analyzing Twitter interactions (An et al., 2011, 2012; Stefanov et al., 2020), often using small English-only datasets (Kulkarni et al., 2018; Potthast et al., 2018; Kiesel et al., 2019; Baly et al., 2020a; Da San Martino et al., 2023; Nakov et al., 2023a,b; Barrón-Cedeño et al., 2023a,b; Azizov et al., 2023; Chen et al., 2018; Fan et al., 2019; Spinde et al., 2022).

The veracity of the news media has been explored using PLMs to estimate the reliability of the source, correlated with the ratings of human experts (Yang and Menczer, 2023). Mehta and Goldwasser (2023) introduced a framework that combines graph-based models, PLMs, and human experience to profile news media, effectively identifying fake news with minimal human input. Early work estimated the reliability of the source based on the stance toward true/false claims using English datasets (Mukherjee and Weikum, 2015; Popat et al., 2017). Recent approaches, such as Baly et al. (2020c), used gold labels and various English sources as features to profile media with PLMs. Although the features in the aforementioned studies are obtained from diverse sources, they neglect the inherent relationships between the media.

To bridge this gap, graphs emerged as a comprehensive and effective framework for representation learning (Mehta et al., 2022). Panayotov et al. (2022) constructed a graph based on the principle of homophily, suggesting that similar media sources attract similar audiences. The framework leveraged the audience overlap of media outlets to build a huge graph that models the interactions between media, and to learn expressive representation for the nodes using GNNs. However, media graphs are characterized by disconnected components and scarce labels. To overcome these limitations, we propose MGM to capture long-range dependencies, and to improve the effective use of information across the entire graph.

2.2 Graph Neural Networks

The current *de facto* design of GNNs follows the message-passing framework (Yang et al., 2022), where they learn node representations by aggregating information from local neighbors. However, media graphs suffer from challenges such

[1] www.mediabiasfactcheck.com

as multiple disconnected components and limited labels, which makes it difficult for GNNs to capture long-range dependencies and to learn informative node representations (Cui et al., 2020; Sun et al., 2020; Bodnar et al., 2021; Dai et al., 2021, 2022). Recent efforts on the integration of external memory modules to store embeddings of all nodes allow GNNs to capture long-range dependencies across graphs (Kang, 2021; Zhang et al., 2022; Ma et al., 2022). However, these methods typically require storing embeddings for all nodes in the graph, which results in high memory costs and low efficiency during testing. Unlike previous approaches, MGM focuses on a small set of candidate nodes, which are more likely to be selected as global similar nodes based on a Dirichlet prior distribution applied to the training nodes (Sethuraman, 1994).

3 Problem Formulation

We instantiate the graph learning task with the node classification problem in a semi-supervised setting (Kipf and Welling, 2016; Veličković et al., 2018), where labels are only available for a small subset of the nodes. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{Y}^l\}$ represent a partially-labeled graph, where $\mathcal{V} = \{v_i\}_{i=1}^N$ is a set of nodes, $\mathcal{E}$ is a set of edges, and N is the total number of nodes. The node features are denoted as $\mathbf{X} \in \mathbb{R}^{N \times F}$, where F is the feature dimension. Since most nodes are unlabeled, $\mathcal{V}$ can be divided into labeled nodes $\mathcal{V}^l$ with labels $\mathbf{Y}^l$, and unlabeled nodes $\mathcal{V}^u$. The labels $\mathbf{Y}^l \in \mathbb{R}^{N_l \times C}$ are in a one-hot form, where N_l and C represent the number of labeled nodes and the number of classes, respectively. The goal of semi-supervised learning is to learn the model parameters θ by maximizing the marginal distribution of the overall labeled nodes, i.e., $p_\theta(\mathbf{Y}^l \mid \mathbf{X}, \mathcal{E}) = \prod_{n \in \mathcal{V}^l} p_\theta(\mathbf{y}_n \mid \mathbf{X}, \mathcal{E})$ on the training graph.

4 Methodology

In this section, we introduce our proposed framework that aims to enhance the performance of GNNs and PLMs for factuality and political bias. GNNs struggle with media graphs containing disconnected components and insufficient labels, while PLMs only use textual features, neglecting the intricate relationships between entities. Thus, we aim to develop a framework that not only enhances the performance of GNNs, but also integrates structural information into the PLMs.

Toward this goal, we propose our novel MGM framework, which uses features, structural patterns, and label information of global similar nodes to enhance the performance of GNNs for factuality and political bias classification. Moreover, we integrate MGM with PLMs to further improve the performance. With such a framework, GNN can effectively overcome the challenges in media graphs. Meanwhile, PLMs can benefit from integrated structural information, leading to enhanced performance for our tasks.

4.1 The MGM Framework

Following (Qu et al., 2019, 2021), we adopt a probabilistic framework for node classification, treating node representations $\mathbf{Z}$ as latent variables determined by a GNN. To improve the performance of the model, we propose to augment the GNNs with information about *global similar nodes*, i.e., nodes in the entire graph that have similar node features and local geometric structures. Specifically, we denote the set of global similar nodes of node n as $\mathbf{t}_n \in \{0, 1\}^{N_l}$, where $\mathbf{t}_{nm} = 1$ indicates that node m is a global node similar to n. Similarly to node representations, we also regard the similar node indicator $\mathbf{t}_n$ as a latent variable. Therefore, the joint probability distribution of global information-enhanced method can be factorized as follows:

$$
\begin{aligned}
&p_\theta(\mathbf{Y}^l, \mathbf{T}, \mathbf{Z} \mid \mathbf{X}, \mathcal{E}) = \qquad (1)\\
&= p_\theta(\mathbf{Z} \mid \mathbf{X}, \mathcal{E}) p_\theta(\mathbf{T} \mid \mathbf{Z}) p_\theta(\mathbf{Y}^l \mid \mathbf{Z}, \mathbf{T}),
\end{aligned}
$$

where $\mathbf{T} = [\mathbf{t}_n]_{n \in \mathcal{V}^l}^\top$ are the global similar nodes of all nodes.

However, finding global similar nodes with node representations requires computing representations for all nodes, which is expensive in terms of space and time (Fey et al., 2021). To alleviate this, we propose to store the embeddings of the labeled nodes in the memory and to use them to find global similar nodes. Consequently, the distribution of $\mathbf{T}$ can be replaced by $p_\theta(\mathbf{T} \mid \hat{\mathbf{Z}})$, where $\hat{\mathbf{Z}}$ is the embeddings of the labeled nodes in the memory, i.e., $\hat{\mathcal{V}}^l$. In this case, we can directly retrieve the representation from memory without computing representations for all nodes, thus making it more efficient to obtain the distribution of global similar nodes for both training and prediction.

To reduce the memory size, we select global similar nodes from a small set of candidate nodes, which are a subset of the training nodes. As a result, only the embeddings of these candidate nodes are stored in memory for prediction. To achieve

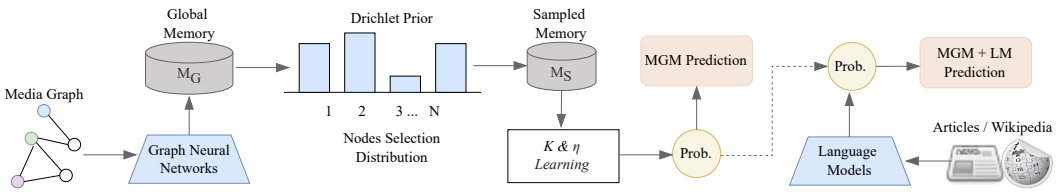

Figure 1: Key components of our proposed approach. GNNs store the representation of the media graphs in an external global memory (M_g). A Dirichlet prior is used to select the distribution of sparse candidate nodes, which are stored in the sampled memory (M_s). The parameters K and η control the number of candidate nodes and their influence, balancing local and global information. Since PLMs miss some of the media representation, they leverage MGM representation-based probabilities for the classification task. Detailed pipeline of the integration MGM with PLMs could be seen in Figure 3 (Appendix E).

this, we assume that $p_\theta(\mathbf{T} \mid \hat{\mathbf{Z}})$ is a sparse distribution, concentrated on a few candidate nodes. Since the candidate set is not known, we introduce a latent variable $\boldsymbol{\omega}$ for each node n, where $\boldsymbol{\omega}_i \in [0,1], s.t. \sum_{i=1}^{N_l} \boldsymbol{\omega}_i = 1$. Here, $\boldsymbol{\omega}_i$ represents the probability that the i-th node in the labeled node is a candidate node. Inspired by (He et al., 2020), we introduce a prior over $\boldsymbol{\omega}$, i.e., $p_\alpha(\boldsymbol{\omega})$ with parameter α. This prior is designed to encourage a sparse distribution over $\boldsymbol{\omega}$. Therefore, the joint distribution of the method is now defined as follows:

$$p_\theta(\mathbf{Y}^l, \mathbf{T}, \mathbf{Z}, \boldsymbol{\omega} \mid \mathbf{X}, \mathcal{E}, \hat{\mathbf{Z}}) = p_\alpha(\boldsymbol{\omega}) \quad (2)$$
$$p_\theta(\mathbf{Z} \mid \mathbf{X}, \mathcal{E})\, p_\theta(\mathbf{T} \mid \boldsymbol{\omega}, \hat{\mathbf{Z}})\, p_\theta(\mathbf{Y}^l \mid \mathbf{T}, \mathbf{Z})\,.$$

Next, we introduce the parameterization of our probabilistic framework.

Prior distribution over $\boldsymbol{\omega}$. We use the Dirichlet distribution as the prior distribution over $\boldsymbol{\omega}$, i.e., $p_\alpha(\boldsymbol{\omega}) \propto \prod_{i=1}^{N} \boldsymbol{\omega}_i^{\alpha_i - 1}$, where α_i is the concentration parameter of the distribution. The concentration parameter α is a positive value and a smaller value of α prefers a sparser distribution over $\boldsymbol{\omega}$ (He et al., 2020). In our experiments, we set $\alpha < 1$ to encourage the sparse nodes distribution.

Prior distribution over node representations Z. We model the prior distribution over node representations as Gaussian distributions (Bojchevski and Günnemann, 2018), which are obtained with GNNs due to their effectiveness in graph learning tasks. Therefore, the prior distribution over $\mathbf{Z}$ is defined as follows:

$$p_\theta(\mathbf{Z} \mid \mathbf{X}, \mathcal{E}) = \mathcal{N}(\mathbf{Z} \mid \mathrm{GNN}_\theta(\mathbf{X}, \mathcal{E}), \sigma_1^2 \mathbf{I}), \quad (3)$$

where σ_1^2 is the learned variance of the prior and GNN_θ is an L-layer GNN with parameter θ.

Prior distribution over T. To obtain global similar nodes of node n, we define a prior distribution over $\mathbf{T}$ as follows:

$$p_\theta(\mathbf{T} \mid \boldsymbol{\omega}, \hat{\mathbf{Z}}) = \mathrm{Mul}(\mathbf{T} \mid K, f_\theta(\boldsymbol{\omega}, \hat{\mathbf{Z}})), \quad (4)$$

where $\mathrm{Mul}(\cdot)$ represents the multinomial distribution, K denotes the predefined number of global similar nodes, and f_θ is designed as a parameterized function that outputs the parameters of the multinomial distribution.

Prediction of label Y. Finally, we use the node representation $\mathbf{Z}$ and information from global similar nodes to predict the label. Specifically, we leverage the labels of global similar nodes and first predict the label based on its representation:

$$p_\theta(\mathbf{Y} \mid \mathbf{Z}) = \mathrm{Cat}(\mathbf{Y} \mid \mathbf{Z})\,, \quad (5)$$

where $p_\theta(\mathbf{Y} \mid \mathbf{Z})$ is formulated as a categorical distribution. Then, we predict the label using the labels of global similar nodes:

$$p_\theta(\mathbf{Y} \mid \mathbf{T}) \propto \sum_{\mathbf{N}, \mathbf{M} \in \hat{\mathcal{V}}^l} \mathbf{T}_{\mathbf{NM}} \cdot \mathbf{Y}_{\mathbf{M}}\,, \quad (6)$$

where $\mathbf{T}_{\mathbf{NM}}$ represents the indices of global similar nodes for the predicted nodes set $\mathbf{N}$. Additionally, $\mathbf{Y}_{\mathbf{M}}$ denotes the one-hot labels of the nodes in $\mathbf{M}$, where $\mathbf{M}$ is the set of global similar nodes. Finally, the predicted label distribution is defined as follows:

$$p_\theta(\mathbf{Y} \mid \mathbf{Z}, \mathbf{T}) = \eta\, p_\theta(\mathbf{Y} \mid \mathbf{Z}) + (1-\eta)\, p_\theta(\mathbf{Y} \mid \mathbf{T})\,, \quad (7)$$

where $\eta \in [0,1]$ is a trade-off hyper-parameter. When $\eta = 1$, our model only uses local representations of nodes for prediction, which degrades to vanilla GNNs. In contrast, when $0 < \eta < 1$, our model predicts the labels of the nodes using information from both local neighbors and global similar nodes.

4.2 Training Process of MGM

Next, we explain how to learn the model parameters θ based on the graph. Ideally, the marginal likelihood should be optimized during training:

$$p_\theta(\mathbf{Y}^l \mid \mathbf{X}, \mathcal{E}, \hat{\mathbf{Z}}) = \tag{8}$$
$$= \int_{\boldsymbol{\omega}} \int_{\mathbf{Z}} \sum_{\mathbf{T}} p_\theta(\mathbf{Y}^l, \mathbf{T}, \mathbf{Z}, \boldsymbol{\omega} \mid \mathbf{X}, \mathcal{E}, \hat{\mathbf{Z}}) \mathrm{d}\mathbf{Z} \mathrm{d}\boldsymbol{\omega} \,.$$

However, the computation of maximizing the marginal likelihood is intractable due to the marginalization of latent variables. Therefore, we develop a variational Expectation-Maximization (EM) algorithm (Qu et al., 2019) to optimize its evidence lower bound (ELBO) instead:

$$\begin{aligned}
&\mathcal{L}_{\mathrm{ELBO}}(\mathbf{Y}^l; \theta, \phi, \alpha, \lambda) = -D_{\mathrm{KL}}(q_\lambda(\boldsymbol{\omega}) \,||\, p_\alpha(\boldsymbol{\omega})) \\
&- D_{\mathrm{KL}}\big[q_\phi(\mathbf{Z} \mid \mathbf{T}, \mathbf{Y}^l) \,||\, p_\theta(\mathbf{Z} \mid \mathbf{X}, \mathcal{E})\big] \\
&- D_{\mathrm{KL}}\big[q_\phi(\mathbf{T} \mid \mathbf{Y}^l) \,||\, p_\theta(\mathbf{T} \mid \boldsymbol{\omega}, \hat{\mathbf{Z}})\big] \\
&+ \mathbb{E}_{q_\phi(\mathbf{T}|\mathbf{Y}^l) q_\phi(\mathbf{Z}|\mathbf{T}, \mathbf{Y}^l)}\big[\log p_\theta(\mathbf{Y}^l \mid \mathbf{T}, \mathbf{Z})\big] \,,
\end{aligned} \tag{9}$$

where $D_{\mathrm{KL}}[\cdot||\cdot]$ is the Kullback-Leibler (KL) divergence, q represents the variational distribution to approximate the model posterior distribution and adheres to the following factorization form:[2]

$$q_\lambda(\boldsymbol{\omega}) q_\phi(\mathbf{T}, \mathbf{Z}, \boldsymbol{\omega} \mid \mathbf{Y}^l) q_\phi(\mathbf{T} \mid \mathbf{Y}^l) q_\phi(\mathbf{Z} \mid \mathbf{T}, \mathbf{Y}^l) \,,$$

where ϕ and λ are variational parameters.

Note that we use the mean-field assumption to approximate the posterior of $\boldsymbol{\omega}$ to simplify the variational distributions. For computational convenience, we assume that the variational distributions of these latent variables have the same distribution form as their prior distributions. Therefore, we define the variational distributions of $\boldsymbol{\omega}$, $\mathbf{T}$ and $\mathbf{Z}$ to be Dirichlet, multinomial, and Gaussian distributions, respectively.

Note that the KL divergence in Equation (9) has a closed-form solution, and we approximate the expectation using a Monte Carlo method by sampling from the variational distributions. In variational EM, the variational parameters ϕ and the model parameters θ are learned alternately. In the E-step, we fix θ and update ϕ by minimizing the KL divergence to approximate the true posteriors. In the M-step, we fix ϕ and update θ by maximizing the expected log-likelihood.

[2]We omit the dependence of variational distributions on node features $\mathbf{X}$, edges $\mathcal{E}$ and memory $\hat{\mathbf{Z}}$ for brevity.

4.3 Prediction Process of MGM

After training, we expect to obtain a sparse distribution $q_\lambda(\boldsymbol{\omega})$, allowing us to select a subset of the candidate nodes. In this case, we can select candidate nodes over a certain probability threshold, thus reducing the memory size and improving the efficiency for prediction. Specifically, we calculate the expected value of $q_\lambda(\boldsymbol{\omega})$ for each node i, which is given by $\mathbb{E}_{q_\lambda(\boldsymbol{\omega})}[\boldsymbol{\omega}_i] = \lambda_i / \sum_{j=1}^{N_l} \lambda_j$, and then we select the top-M nodes that occupy 90% of the probability mass as candidate nodes.

We then leverage the embeddings of the memorized candidate nodes $\hat{\mathbf{Z}}_{\boldsymbol{\omega}}$ and p_θ to predict the labels of the test nodes $\tilde{n}$ based on Equation (7).

4.4 Enhancing PLMs Predictions with MGM

Next, we demonstrate how MGM improves the performance of PLMs by incorporating information from global similar nodes. Given textual features $\mathbf{S}$, such as those from *Articles* and *Wikipedia* pages for the media outlet, we first fine-tune the PLMs using the cross-entropy loss. Then, we concatenate the predicted label distribution from the PLMs with MGM to obtain the final label distribution:

$$p_{\psi,\theta}(\mathbf{Y} \mid \mathbf{S}, \mathbf{Z}, \mathbf{T}) = \tag{10}$$
$$= \mathrm{Softmax}(\oplus(p_\psi(\mathbf{Y} \mid \mathbf{S}), p_\theta(\mathbf{Y} \mid \mathbf{Z}, \mathbf{T}))\mathbf{W} + \mathbf{b}) \,,$$

where ψ are the parameters of the fine-tuned PLMs, $\oplus$ is the concatenation operation, $p_\psi(\mathbf{Y} \mid \mathbf{S})$ is the label distribution predicted by the fine-tuned PLMs, $p_\theta(\mathbf{Y} \mid \mathbf{Z}, \mathbf{T})$ is the label distribution predicted by MGM, which is based on Equation (7), $\mathbf{W}$ and $\mathbf{b}$ are the parameters of the linear classifier. More details are given in Figure 3 and Appendix E.

5 Experiments

5.1 Research Questions

We explore the following research questions:

- (**RQ1**) Can MGM tackle disconnected components and label sparsity in media graphs for factuality and political bias detection tasks?
- (**RQ2**) How do the number of global similar nodes K and the trade-off hyper-parameter η affect the performance of MGM?
- (**RQ3**) How does the memory module affect the performance of MGM?
- (**RQ4**) How does MGM elevate the performance of PLMs when faced with the challenge of missing text in *Wikipedia* or *Articles*?

Model	Fact-2020			Bias-2020		
	Macro-F1	Accuracy	Average Recall	Macro-F1	Accuracy	Average Recall
Majority class	22.93 ± 0.00	52.43 ± 0.00	33.33 ± 0.00	19.18 ± 0.00	40.39 ± 0.00	33.33 ± 0.00
GCN	25.55 ± 0.94	52.55 ± 0.28	34.74 ± 0.49	38.58 ± 5.13	42.90 ± 4.81	41.48 ± 5.11
+ MGM	**43.05 ± 2.03**	**53.37 ± 1.00**	**43.42 ± 1.53**	**42.77 ± 1.09**	**45.23 ± 1.70**	**43.80 ± 3.19**
GAT	33.75 ± 3.12	54.18 ± 0.77	39.26 ± 2.12	41.22 ± 1.79	50.34 ± 0.78	48.06 ± 1.05
+ MGM	**43.63 ± 2.80**	**55.11 ± 1.44**	**43.54 ± 2.71**	**50.41 ± 2.86**	**54.06 ± 1.98**	**51.96 ± 0.79**
GraphSAGE	42.68 ± 2.55	58.02 ± 1.18	45.70 ± 1.25	39.35 ± 1.07	50.00 ± 1.32	49.09 ± 1.06
+ MGM	**46.67 ± 1.58**	**59.00 ± 1.00**	**47.40 ± 1.67**	**46.77 ± 1.82**	**51.04 ± 0.67**	**50.18 ± 0.92**
SGC	22.73 ± 0.07	51.39 ± 0.28	33.10 ± 0.18	35.37 ± 0.60	45.34 ± 0.97	45.80 ± 0.76
+ MGM	**41.28 ± 1.42**	**53.95 ± 0.77**	**41.32 ± 1.22**	**39.11 ± 0.51**	**46.74 ± 0.78**	**47.10 ± 0.74**
DNA	22.75 ± 0.03	**51.74 ± 0.00**	33.33 ± 0.00	24.27 ± 3.02	40.69 ± 0.73	35.03 ± 1.02
+ MGM	**34.04 ± 1.60**	50.81 ± 1.30	**36.56 ± 1.98**	**33.22 ± 1.13**	**42.55 ± 2.50**	**38.59 ± 1.81**
FiLM	43.32 ± 2.25	57.09 ± 0.77	44.46 ± 1.40	39.33 ± 2.76	47.55 ± 1.12	47.85 ± 1.07
+ MGM	**49.68 ± 1.62**	**57.90 ± 2.39**	**49.94 ± 1.68**	**45.33 ± 2.76**	**48.25 ± 2.65**	**48.61 ± 2.84**
FAGCN	24.77 ± 7.52	47.04 ± 3.71	36.12 ± 5.30	19.69 ± 0.65	39.88 ± 0.28	33.71 ± 0.31
+ MGM	**48.77 ± 0.00**	**53.14 ± 1.66**	**49.19 ± 0.00**	**45.02 ± 3.00**	**45.69 ± 2.88**	**45.07 ± 3.00**
GATv2	51.42 ± 2.32	61.13 ± 1.04	55.36 ± 1.74	48.48 ± 1.68	55.11 ± 1.85	53.07 ± 1.75
+ MGM	**54.50 ± 2.55**	**62.72 ± 1.01**	**57.36 ± 1.06**	**52.41 ± 2.85**	**55.46 ± 2.45**	**54.00 ± 2.61**

Table 1: Performance of GNN baselines and their MGM enhanced versions for the factuality and political bias tasks on the ACL-2020 dataset, with the majority class baseline and SVM included as naïve and non-graphical methods. The higher performance is highlighted in **Bold.**

Model	Fact-2020		Bias-2020	
	Macro-F1 †/ §	Average Recall †/ §	Macro-F1 †/ §	Average Recall †/ §
GCN	42.04 ± 1.91 / **43.05 ± 2.04**	41.97 ± 1.77 / **43.42 ± 1.53**	**42.77 ± 1.10** / 42.37 ± 2.09	43.72 ± 3.15 / **43.80 ± 3.19**
GAT	40.63 ± 3.28 / **43.63 ± 2.81**	40.23 ± 2.88 / **43.54 ± 2.71**	**50.41 ± 2.86** / 47.12 ± 1.69	**51.96 ± 0.79** / 49.39 ± 2.02
GraphSAGE	45.11 ± 1.44 / **46.68 ± 1.59**	45.69 ± 1.42 / **47.40 ± 1.67**	44.96 ± 1.72 / **46.78 ± 1.83**	49.78 ± 0.66 / **51.04 ± 0.67**
SGC	**41.29 ± 1.42** / 39.65 ± 2.00	**41.32 ± 1.22** / 40.04 ± 1.50	38.32 ± 1.41 / **39.12 ± 0.51**	46.74 ± 0.56 / **47.10 ± 0.74**
DNA	33.48 ± 3.69 / **34.05 ± 1.60**	35.99 ± 2.24 / **36.56 ± 1.98**	**33.22 ± 1.13** / 32.25 ± 4.14	**38.59 ± 1.81** / 36.63 ± 4.53
FiLM	45.12 ± 3.38 / **49.68 ± 1.62**	45.58 ± 2.78 / **49.94 ± 1.68**	43.98 ± 2.57 / **45.33 ± 2.76**	47.86 ± 1.46 / **48.61 ± 2.84**
FAGCN	**48.77 ± 0.00** / 46.88 ± 2.88	**49.19 ± 0.00** / 48.05 ± 2.65	**45.02 ± 3.00** / 44.36 ± 1.24	**45.07 ± 3.00** / 44.47 ± 1.36
GATv2	54.13 ± 2.93 / **54.50 ± 2.55**	56.82 ± 1.94 / **57.36 ± 1.06**	**52.41 ± 2.85** / 50.44 ± 0.95	**54.00 ± 2.61** / 52.02 ± 1.29

Table 2: Summary of the MGM results detailing the performance variation the use of between using full memory (†) and a reduced (90%) memory allocation (§) for each GNN.

5.2 Dataset

The dataset for media factuality and political bias introduced by Baly et al. (2020c) comprises 859 media sources[3], their domain names, and corresponding gold labels. These labels are sourced from MBFC, a platform supported by independent journalists. Factuality is given on a three-point scale: high, mixed, and low. Political bias is also on a three-point scale: left, center, right. Panayotov et al. (2022) used Alexa Rank[4] to create a graph based on audience overlap, using the 859 media as seed nodes. Media sources that shared the same audience, as determined by Alexa Rank, were connected with an edge. Alexa Rank returned a maximum of five similar media sources for each medium, which could be part of the initial seed nodes or newly identified media. In the resulting graph, the nodes represent the media sources, and the edges represent the precentage of audience overlap between two media. We used these publicly available graph data (the only one of its kind) to train GNNs for the factuality and political bias of the media. More details are given in Appendix A.

5.3 Baselines

For evaluation, we consider two categories of baselines, including GNN-based and PLM-based models. For GNN models, we select eight well-known models, including GCN (Kipf and Welling, 2016), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), SGC (Wu et al., 2019), DNA (Fey, 2019), FiLM (Brockschmidt, 2020), FAGCN (Bo et al., 2021) and GATv2 (Brody et al., 2022). More detail about these GNN baselines are provided in Appendix B. For PLMs, we use four popular encoder models, including BERT, RoBERTa, DistillBERT, and DeBERTaV3. We

[3] https://github.com/ramybaly/News-Media-Reliability

[4] http://www.alexa.com/siteinfo

Model	60% labels	80% labels	100% labels
GAT	37.90 ± 0.41	39.22 ± 0.71	33.75 ± 3.12
+MGM	**40.80 ± 3.83**	**41.99 ± 1.44**	**43.63 ± 2.80**
FiLM	39.89 ± 1.69	38.59 ± 4.30	43.32 ± 2.25
+MGM	**42.93 ± 4.00**	**38.62 ± 2.73**	**49.68 ± 1.62**
FAGCN	24.32 ± 3.18	22.73 ± 0.00	24.77 ± 7.52
+MGM	**34.10 ± 7.36**	**39.89 ± 4.04**	**48.77 ± 0.00**
GATv2	40.54 ± 1.47	42.14 ± 2.76	51.42 ± 2.32
+MGM	**42.98 ± 2.51**	**44.35 ± 2.14**	**54.50 ± 2.55**

Table 3: The impact of different proportions of training labeled data on the performance (Macro-F1) of MGM for the Fact-2020 task.

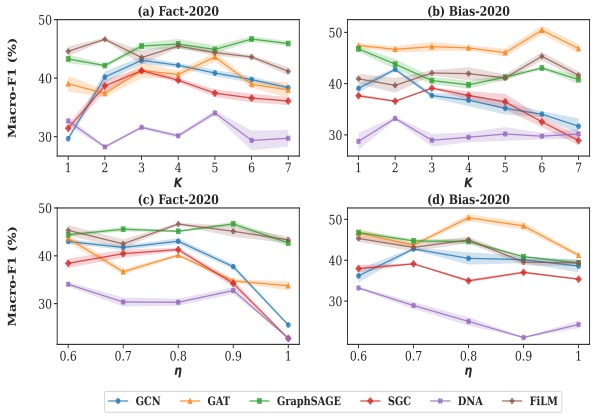

Figure 2: MGM performance across all GNNs for both tasks, evaluated for different values of K (global similar nodes) and η (trade-off hyper-parameter).

compare our results to state-of-the-art results for the factuality and political bias of the news media (Panayotov et al., 2022; Mehta et al., 2022).

6 Discussion

6.1 Overall Performance

To answer **RQ1**, we conduct factuality and political bias classification experiments in a semi-supervised setting. The experimental results reported in Table 1 demonstrate that MGM can improve the performance of existing GNNs in almost all cases. For example, when applied on the Fact-2020 dataset, MGM improves the Macro-F1 performance of GCN, GAT, SGC, and DNA by 17.5%, 9.8%, 18.5%, and 11.4%, respectively. Similarly, for Bias-2020, we can observe that GNNs equipped with MGM consistently outperform the corresponding base models in all evaluation measures.

We conducted a series of experiments using different proportions of training labels to assess the performance of MGM as shown in Table 3. The results indicate a clear trend: as we increase the percentage of training labels, the model's performance improves significantly compared to the baseline. Due to limited data, using a smaller percentage of training labels results in modest improvements over the baseline, constraining the model's ability to generalize well to unseen data. MGM effectively addresses **RQ1** by leveraging global similar nodes in media graphs with disconnected components and label sparsity for factuality and political bias detection. Our evaluation extends to Fact-2018 and depicts MGM's stable performance across different datasets presented in Appendix D.

6.2 Impact of the Number of Global Similar Nodes

Next, we turn to **RQ2** to understand the impact of the number of global similar nodes K. Specifically, we investigate the performance of MGM with different values of K. As shown in Figures 2(a) and 2(b), leveraging a few global similar nodes can improve the performance of the base GNNs. For example, both GCN and SGC exhibit similar patterns, peaking in performance at K=3 on the factuality task. The performance of GNNs enhanced with MGM decreases when K exceeds a certain threshold. This is attributed to the introduction of noise by incorporating excessive information from numerous global similar nodes.

6.3 Impact of the Trade-off Hyper-Parameter

Recall that in Section 4.1, we introduced a hyper-parameter η as a hyper-parameter that influences the predicted label distribution. When $\eta = 1$, MGM only relies on local node representations for prediction, degrading to a vanilla GNN. In contrast, when $\eta < 1$, our model incorporates information from both local neighbors and global similar nodes to predict the node labels. To further investigate the impact of the trade-off hyper-parameter η, we analyze the sensitivity of MGM to its value. The experimental results are shown in Figures 2(c) and 2(d). We find that compared to $\eta = 1$, MGM yields improved performance when $\eta < 1$ in most cases. For example, GCN achieves its best performance when integrated with MGM using an η value of 0.8. Hence, the effectiveness of incorporating information from global similar nodes highlighted in the results validates the **RQ2**.

	Model	Fact-2020						Bias-2020					
		Articles			Wikipedia			Articles			Wikipedia		
		Macro-F1	Accuracy	Avg Recall	Macro-F1	Accuracy	Avg Recall	Macro-F1	Accuracy	Avg Recall	Macro-F1	Accuracy	Avg Recall
STAGE 1	$\text{BERT}_{\text{Base}}$	**38.27**	**63.37**	**39.65**	34.64	59.30	37.98	**65.38**	**68.02**	**64.01**	58.70	62.79	58.69
	$\text{RoBERTa}_{\text{Base}}$	33.55	62.79	37.65	25.29	59.30	33.36	63.34	65.70	62.41	58.73	62.78	58.71
	$\text{DistilBERT}_{\text{Base}}$	35.27	62.21	37.65	25.27	61.01	33.30	65.04	67.44	63.91	58.71	62.85	58.72
	$\text{DeBERTaV3}_{\text{Base}}$	25.28	61.02	33.35	**40.81**	**61.05**	**40.75**	58.72	62.80	58.69	**59.65**	**63.37**	**59.15**
STAGE 2	$\text{BERT}_{\text{MGM}_{\text{GATv2}}}$	**76.18**	**81.98**	**71.86**	73.69	81.40	70.92	83.74	84.30	83.88	**82.25**	**82.56**	**81.40**
	$\text{RoBERTa}_{\text{MGM}_{\text{GATv2}}}$	69.89	80.23	66.85	72.73	79.65	70.52	85.51	86.05	85.38	81.32	81.98	80.71
	$\text{DistilBERT}_{\text{MGM}_{\text{GATv2}}}$	74.55	81.40	71.48	73.03	80.23	70.84	87.20	87.79	86.90	80.68	81.40	80.25
	$\text{DeBERTaV3}_{\text{MGM}_{\text{GATv2}}}$	64.87	77.91	62.97	**74.56**	**81.98**	**73.20**	**87.71**	**88.37**	**87.70**	80.26	80.81	79.28

Table 4: **Stage 1:** Performance of logistic regression (meta-learner) on PLM probabilities with missing media attributed as probabilities (0.0, 0.0, 0.0). **Stage 2:** Performance of the logistic regression (meta-learner) on PLMs probabilities + $\text{MGM}_{\text{GATv2}}$ probabilities for missing media for factuality and political bias of the ACL-2020 dataset.

	Model	Fact-2020			Bias-2020		
		Macro-F1	Accuracy	Avg Recall	Macro-F1	Accuracy	Avg Recall
	Node classification (NC) (Mehta et al., 2022)	68.90	63.72	-	-	-	-
	InfOp Best Model (Mehta et al., 2022)	72.55	66.89	-	-	-	-
	GRENNER (Panayotov et al., 2022)	69.61	74.27	-	91.93	92.08	-
STAGE 3	$\text{DeBERTaV3}_{\text{MGM}_{\text{GATv2}}}$ + $\text{BERT}_{\text{MGM}_{\text{GATv2}}}$	78.43	83.04	75.03	92.64	92.98	92.67
STAGE 4	$\text{DeBERTaV3}_{\text{MGM}_{\text{GATv2}}}$ + $\text{BERT}_{\text{MGM}_{\text{GATv2}}}$ + MGM_{FiLM}	**79.72 ± 0.00**	**84.21 ± 0.00**	**76.54 ± 0.00**	93.04 ± 0.26	**93.45 ± 0.23**	**93.19 ± 0.26**
	$\text{DeBERTaV3}_{\text{MGM}_{\text{GATv2}}}$ + $\text{BERT}_{\text{MGM}_{\text{GATv2}}}$ + $\text{MGM}_{\text{FAGCN}}$	75.69 ± 3.49	81.29 ± 3.09	72.24 ± 3.35	**93.08 ± 0.24**	**93.45 ± 0.23**	93.15 ± 0.34
	$\text{DeBERTaV3}_{\text{MGM}_{\text{GATv2}}}$ + $\text{BERT}_{\text{MGM}_{\text{GATv2}}}$ + $\text{MGM}_{\text{GATv2}}$	77.96 ± 0.30	82.69 ± 0.29	74.84 ± 0.16	92.71 ± 0.46	93.10 ± 0.44	92.72 ± 0.52

Table 5: Previous studies (Mehta et al., 2022; Panayotov et al., 2022) and our best results. **Stage 3:** We concatenate the probabilities of the best PLMs from *Wikipedia* and *Articles* and use logistic regression to make predictions. **Stage 4:** We use probabilities from the Stage 3 model and concatenate with the probabilities of three GNNs (MGM_{FiLM}, $\text{MGM}_{\text{FAGCN}}$ and $\text{MGM}_{\text{GATv2}}$).

6.4 Effectiveness of the Memory Module

Recall that in Section 4.3, MGM leveraged a Dirichlet prior to select a small set of candidate nodes and stored their node embeddings in the sampled memory ($\mathbf{M_S}$) for prediction. To compare the effectiveness of the sampled memory module to the full memory module ($\mathbf{M_G}$), which stores all the training node embeddings, we conducted a performance comparison between the two memory modules. The experimental results are given in Table 2, and they answer **RQ3** that MGM using sampled memory achieves comparable performance to MGM when using full memory. For example, for the GAT model, the performance is higher when using sampled memory compared to using full memory. This suggests that the sampled memory effectively captures sufficient information, allowing MGM to maintain its performance even with limited memory. Experimental results on the Fact-2018 dataset are reported in the Appendix D.

6.5 Impact of Integrating MGM with PLMs

To answer **RQ4**, we integrate the MGM probabilities with those from deep learning models based on textual features, and we observe that this substantially enhances the performance. Initially, with zero probabilities for missing textual features, we achieved accuracies of 68.02% for political bias in *Articles*, 63.37% for *Wikipedia*, 63.37% for factuality in *Articles*, and 61.05% for *Wikipedia*, respectively (see Table 4). Replacing the zero probabilities with the best $\text{MGM}_{\text{GATv2}}$ improved the performance by up to 30%. Further concatenating the best model probabilities in stage three led to additional gains, and in stage four, our models outperformed previous state-of-the-art results in political bias and factuality (Panayotov et al., 2022; Mehta et al., 2022) (see Table 5).

7 Conclusion and Future Work

We introduced MGM, an innovative EM framework that substantially improves the performance of GNNs on media graphs by leveraging global similar nodes. The external memory module of MGM efficiently stores and retrieves node representations, addressing the challenge of test-time inefficiency by selecting global similar nodes from a smaller candidate set based on a sparse node selection distribution. Our experiments demonstrate that the integration of MGM features with PLMs consistently improves over existing baselines and establishes new state-of-the-art results.

In future work, we plan to explore multi-graph fusion, multi-task learning, and ordinal classification for diverse graph structures in media profiling.

Limitations

The graph dataset, originating from ACL-2020 media nodes, was constructed using the Alexa Rank siteinfo tool, which is currently unavailable. Although the graph aids the task by capturing the inherent and hidden relationships between media, building such graphs is complex and resource-intensive. The research largely relies on Western definitions of political bias (left/center/right), which may not accurately capture the nuanced ideological biases present in news outlets from other cultural or political contexts. Moreover, the available graph is limited to the 2020 dataset. We are actively working on constructing graphs for the latest benchmarks, which include a larger number of media sources and updated MBFC rankings. We faced limitations in collecting *Articles* and *Wikipedia* texts from media sources from the ACL-2020 dataset due to the inaccessibility of their websites.

Ethical Statement

Optimizing model architectures to enhance energy efficiency in training and inference operations is crucial for reducing environmental impact. Instead of relying on extensive computational resources to train complex models—which significantly increase carbon emissions, we propose improving model performance with less computational power. The *Articles* from the media pages were compiled in strict compliance with legal and ethical standards. We carefully reviewed the terms of use for all websites to ensure that our data collection processes adhered to them. Our compilation focused solely on publicly available data, avoiding paywalls and subscription models. Transparent data collection methods were designed to minimize the impact on source websites, including limiting the access frequency to prevent resource strain.

References

Sadia Afroz, Michael Brennan, and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE Symposium on Security and Privacy*, pages 461–475. IEEE.

Jisun An, Meeyoung Cha, Krishna Gummadi, Jon Crowcroft, and Daniele Quercia. 2012. Visualizing media bias through Twitter. In *AAAI ICWSM*, volume 6.

Jisun An, Meeyoung Cha, P. Krishna Gummadi, and Jon Crowcroft. 2011. Media landscape in Twitter: A world of new conventions and political diversity. In *AAAI ICWSM*.

Dilshod Azizov, S Liang, and P Nakov. 2023. Frank at checkthat! 2023: Detecting the political bias of news articles and news media. *Working Notes of CLEF*.

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020a. 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)*, pages 4982–4991, Online. Association for Computational Linguistics.

Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. 2018. Predicting factuality of reporting and bias of news media sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3528–3539, Brussels, Belgium. Association for Computational Linguistics.

Ramy Baly, Georgi Karadzhov, Jisun An, Haewoon Kwak, Yoan Dinkov, Ahmed Ali, James Glass, and Preslav Nakov. 2020b. What was written vs. who read it: news media profiling using text analysis and social media context. *arXiv preprint arXiv:2005.04518*.

Ramy Baly, Georgi Karadzhov, Jisun An, Haewoon Kwak, Yoan Dinkov, Ahmed Ali, James Glass, and Preslav Nakov. 2020c. What was written vs. who read it: News media profiling using text analysis and social media context. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3364–3374, Online. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Firoj Alam, Tommaso Caselli, Giovanni Da San Martino, Tamer Elsayed, Andrea Galassi, Fatima Haouari, Federico Ruggeri, Julia Maria Struß, Rabindra Nath Nandi, et al. 2023a. The clef-2023 checkthat! lab: Checkworthiness, subjectivity, political bias, factuality, and authority. In *European Conference on Information Retrieval*, pages 506–517. Springer.

Alberto Barrón-Cedeño, Firoj Alam, Andrea Galassi, Giovanni Da San Martino, Preslav Nakov, Tamer Elsayed, Dilshod Azizov, Tommaso Caselli, Gullal S Cheema, Fatima Haouari, et al. 2023b. Overview of the clef–2023 checkthat! lab on checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 251–275. Springer.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3950–3957.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. 2021. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pages 1026–1037.

Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*.

Marc Brockschmidt. 2020. Gnn-film: Graph neural networks with feature-wise linear modulation. In *International Conference on Machine Learning*, pages 1144–1152. PMLR.

Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? In *International Conference on Learning Representations*.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks?

Ceren Budak, Sharad Goel, and Justin M Rao. 2016. Fair and balanced? quantifying media bias through crowdsourced content analysis. *Public Opinion Quarterly*, 80(S1):250–271.

Sonia Castelo, Thais Almeida, Anas Elghafari, Aécio Santos, Kien Pham, Eduardo Nakamura, and Juliana Freire. 2019. A topic-agnostic approach for identifying fake news pages. In *Companion proceedings of the 2019 World Wide Web conference*, pages 975–980.

Wei-Fan Chen, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. 2018. Learning to flip the bias of news headlines. In *Proceedings of the 11th International conference on natural language generation*, pages 79–88.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations*.

Nadia K Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the association for information science and technology*, 52(1):1–4.

Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 976–985.

Giovanni Da San Martino, Firoj Alam, Maram Hasanain, Rabindra Nath Nandi, Dilshod Azizov, and Preslav Nakov. 2023. Overview of the CLEF-2023 CheckThat! lab task 3 on political bias of news articles and news media. In *Working Notes of CLEF 2023–Conference and Labs of the Evaluation Forum*, CLEF '2023, Thessaloniki, Greece.

Enyan Dai, Charu Aggarwal, and Suhang Wang. 2021. Nrgnn: Learning a label noise resistant graph neural network on sparsely and noisily labeled graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 227–236.

Enyan Dai, Wei Jin, Hui Liu, and Suhang Wang. 2022. Towards robust graph neural networks for noisy graphs with sparse labels. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 181–191.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

James Fairbanks, Natalie Fitch, Nathan Knauf, and Erica Briscoe. 2018. Credibility assessment in the news: do we need to read. In *Proc. of the MIS2 Workshop held in conjuction with 11th Int'l Conf. on Web Search and Data Mining*, pages 799–800. ACM.

Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. 2019. In plain sight: Media bias through the lens of factual reporting. *arXiv preprint arXiv:1909.02670*.

Matthias Fey. 2019. Just jump: Dynamic neighborhood aggregation in graph neural networks. *arXiv preprint arXiv:1904.04849*.

Matthias Fey, Jan E Lenssen, Frank Weichert, and Jure Leskovec. 2021. Gnnautoscale: Scalable and expressive graph neural networks via historical embeddings. In *International Conference on Machine Learning*, pages 3294–3304.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric. In *ICLR 2019 (RLGM Workshop)*.

Xiaobo Guo, Weicheng Ma, and Soroush Vosoughi. 2022. Measuring media bias via masked language modeling. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 1404–1408.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30.

Junxian He, Taylor Berg-Kirkpatrick, and Graham Neubig. 2020. Learning sparse prototypes for text generation. *Advances in Neural Information Processing Systems*, 33:14724–14735.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Austin Hounsel, Jordan Holland, Ben Kaiser, Kevin Borgolte, Nick Feamster, and Jonathan Mayer. 2020. Identifying disinformation websites using infrastructure features. In *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*.

Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. 2018. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117.

Seokho Kang. 2021. K-nearest neighbor learning with graph neural networks. *Mathematics*, 9(8):830.

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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Vivek Kulkarni, Junting Ye, Steve Skiena, and William Yang Wang. 2018. Multi-view models for political ideology detection of news articles. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3518–3527, Brussels, Belgium. Association for Computational Linguistics.

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*.

Guixiang Ma, Vy Vo, Theodore Willke, and Nesreen K Ahmed. 2022. Memory-augmented graph neural networks: A neuroscience perspective. *arXiv preprint arXiv:2209.10818*.

Nikhil Mehta and Dan Goldwasser. 2023. An interactive framework for profiling news media sources. *arXiv preprint arXiv:2309.07384*.

Nikhil Mehta, María Leonor Pacheco, and Dan Goldwasser. 2022. Tackling fake news detection by continually improving social context representations using graph neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1363–1380.

Subhabrata Mukherjee and Gerhard Weikum. 2015. Leveraging joint interactions for credibility analysis in news communities. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 353–362. ACM.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814.

Preslav Nakov, Firoj Alam, Giovanni Da San Martino, Maram Hasanain, Rabindra Nath Nandi, Dilshod Azizov, and Panayot Panayotov. 2023a. Overview of the CLEF-2023 CheckThat! lab task 4 on factuality of reporting of news media. In *Working Notes of CLEF 2023–Conference and Labs of the Evaluation Forum*, CLEF '2023, Thessaloniki, Greece.

Preslav Nakov, Firoj Alam, Giovanni Da San Martino, Maram Hasanain, RN Nandi, D Azizov, and P Panayotov. 2023b. Overview of the clef-2023 checkthat! lab task 4 on factuality of reporting of news media. *Working Notes of CLEF*.

Preslav Nakov, Husrev Taha Sencar, Jisun An, and Haewoon Kwak. 2021. A survey on predicting the factuality and the bias of news media. *arXiv preprint arXiv:2103.12506*.

Panayot Panayotov, Utsav Shukla, Husrev Taha Sencar, Mohamed Nabeel, and Preslav Nakov. 2022. GREENER: Graph neural networks for news media profiling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, EMNLP '22, Abu Dhabi, UAE.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the Web and social media. In *WWW Companion*, pages 1003–1012.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 231–240, Melbourne, Australia. Association for Computational Linguistics.

Meng Qu, Yoshua Bengio, and Jian Tang. 2019. Gmnn: Graph markov neural networks. In *International Conference on Machine Learning*, pages 5241–5250.

Meng Qu, Huiyu Cai, and Jian Tang. 2021. Neural structured prediction for inductive node classification. In *International Conference on Learning Representations*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jayaram Sethuraman. 1994. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650.

Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, and Akiko Aizawa. 2022. Neural media bias detection using distant supervision with BABE–bias annotations by experts. *arXiv preprint arXiv:2209.14557*.

Peter Stefanov, Kareem Darwish, Atanas Atanasov, and Preslav Nakov. 2020. Predicting the topical stance and political leaning of media using tweets. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 527–537, Online. Association for Computational Linguistics.

Ke Sun, Zhouchen Lin, and Zhanxing Zhu. 2020. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5892–5899.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, pages 6861–6871.

Kai-Cheng Yang and Filippo Menczer. 2023. Large language models can rate news outlet credibility. *arXiv preprint arXiv:2304.00228*.

Mingqi Yang, Renjian Wang, Yanming Shen, Heng Qi, and Baocai Yin. 2022. Breaking the expression bottleneck of graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*.

Ningyu Zhang, Xin Xie, Xiang Chen, Shumin Deng, Chuanqi Tan, Fei Huang, Xu Cheng, and Huajun Chen. 2022. Reasoning through memorization: Nearest neighbor knowledge graph embeddings. *arXiv preprint arXiv:2201.05575*.

Appendix

A GNN Data & Task Statistics

Table 6 describes the statistics of the graph data. Factuality is given on a three-point scale: high, mixed, and low. Political bias is also on a three-point scale: left, center, right. Panayotov et al. (2022) used the Alexa[5] (down temporarily) to create a graph based on audience overlap, using 859 media from ACL-2020 (Baly et al., 2020c) as seed nodes. Media sources that shared the same audience, as determined by Alexa, were connected with an edge, provided they met a specific score threshold. Alexa was set to return a maximum of five similar media sources for each medium; these could be part of the initial seed nodes or newly identified media. The primary graph constructed using ACL-2020 dataset media as seed nodes is designated as *level-0*. In this graph, the nodes represent the media sources that publish news or information, and the edges represent the audience overlap for a pair of nodes. The procedure was reiterated five times, leading to the formation of five distinct graph levels. With every subsequent iteration, the graph expanded, encompassing media sources previously identified by Alexa. This iterative expansion resulted in a progressive increase in both the number of nodes and edges at each level.

Upon analyzing the constructed graphs, we observed several disconnected components, each signifying a unique subnetwork of nodes. Naturally, as the graph levels increased, the number of these components decreased. This can be attributed to the fact that an increase in nodes offers more opportunities for components to merge. We opt for graph level 3 to train GNNs, as detailed in Table 6: it represents the most granular level publicly accessible with fewer disconnected components for both factual and bias tasks. The Alexa tool also generated features for each node in the graph, which we treat as node attributes while training the GNNs. These features include site rank, total sites linked in, bounce rate, and the daily time users spend on the site. These features are the numeric values that are described and normalized in the study (Panayotov et al., 2022). We refer to the GNN training tasks as *Fact-2020* and *Bias-2020* for the factuality and political bias tasks, respectively, since both tasks are derived using ACL-2020. As graph-based data become increasingly accessible, we focus exclusively on the graph and its inherent features, promoting an approach tailored to such structures. In contrast, (Panayotov et al., 2022) operates in a supervised setting and uses specialized textual features (e.g., Articles, Wikipedia, Twitter, and YouTube) that are not publicly available. The proposed MGM addresses the unique challenges of the media graph, offering solutions to the research questions described in the designated section 5.1.

[5] http://www.alexa.com/siteinfo

Property	Specification
Nodes	67,350
Edges	200,481
Features	5
Discon. comp.	44
Avg. nodes / comp.	1,500
labeled Nodes	859 (1%)
Unlabeled Nodes	66,492 (99%)
Tasks	Fact-2020, Bias-2020
Factuality task dist.	high (162), mix (249), low (453)
Political Bias task dist.	left (243), center (272), right (349)
Training Split	687 (80% of 1%)
Test Split	172 (20% of 1%)

Table 6: Statistics about the level-3 graph constructed from ACL-2020 (Panayotov et al., 2022).

Table 7 describes the statistics of the level-3 graph constructed from EMNLP -2018 (Panayotov et al., 2022) media in the same way explained in Section 5.2. The EMNLP-2018 dataset comprises 1,066 news outlets, rated on a 3-point scale for factuality (*high, mixed, low*) and a 7-point scale for political bias (*extreme-left, left, center-left, center, center-right, right, and extreme-right*) (Baly et al., 2018). A subsequent analysis (Baly et al., 2020c) identified that the labels *center-left* and *center-right* serve as vague intermediate categories, leading to their exclusion. Furthermore, to minimize subjectivity in the annotator decisions, the *extreme-left* and *extreme-right* categories were amalgamated into the *left* and *right* categories, respectively. This adjustment resulted in a simplified 3-point political bias scale (*left, center, right*) and reduced the dataset to 864 outlets as shown in Table 6, published in ACL-2020, which we consider as our main dataset in section 5.2.

B Baselines

This section summarizes the baseline GNN models that we use as the backbone for our proposed MGM framework to enhance their learning capabilities in the presence of sparcity challenges.

GCN (Kipf and Welling, 2016): GCN simplifies the convolution operation to alleviate the problem

Property	Specification
Nodes	78429
Edges	232530
Features	5
Discon. comp.	88
Avg. nodes / comp.	911
labeled Nodes	1066 (1.35%)
Unlabeled Nodes	77363 (98.65%)
Tasks	Fact-2018
Factuality task dist.	high (265), mixed (268), low (542)
Training Split	852 (80% of 1.35%)
Test Split	214 (20% of 1.35%)

Table 7: Statistics about the level-3 graph constructed from EMNLP-2018.

Property	Specification
Tasks	Fact-2020, Bias-2020
Factuality task dist.	high (295), mix (119), low (58)
Political Bias task dist.	left (152), center (181), right (139)
Training Split	387
Test Split	85

Table 8: Statistics about *Articles* and *Wikipedia* collected from ACL-2020 (Panayotov et al., 2022) dataset.

Hyper-parameter	BERT	RoBERTa	DistilBERT	DeBERTaV3
Batch size	80	100	120	80
Max length	512	512	512	512
Epochs	3	4	5	5
Learning rate	2e-5	2e-5	2e-5	2e-5

Table 9: Experimental setup for PLMs

of overfitting and introduces a renormalization trick to solve the vanishing gradient problem. We set the number of hidden neurons to 16, and the number of layers to 2. ReLU (Nair and Hinton, 2010) is used as the activation function. We do not dropout between GNN layers.

SGC (Wu et al., 2019): SGC shows that the graph convolution in GNNs is actually Laplacian smoothing, which smooths the feature matrix so that nearby nodes have similar hidden representations. SGC removes the weight matrices and nonlinearities between layers. In our experiments, we set the number of hidden neurons to 256, the number of layers to 2, and the number of hops to 2. We do not dropout between GNN layers.

GraphSAGE (Hamilton et al., 2017): GraphSAGE learns embeddings of nodes in the network by sampling and aggregating features from nodes' local neighborhoods. GraphSAGE has different variants based on different feature aggregators, and we adopt GraphSAGE with a mean-based aggregator as our baseline. In our experiments, we set the number of hidden neurons to 64, and the number of layers to 2. ELU (Clevert et al., 2016) is used as the activation function. We do not dropout between GNN layers.

GAT (Veličković et al., 2018): GAT incorporates the attention mechanism into the propagation step, allowing each node to compute its hidden states by attending to its neighbors using self-attention and multi-head attention strategies. we set the number of hidden neurons to 128 per attention head and the number of layers to 3. The number of heads for each layer is set to $4, 4$ and 6. ELU (Clevert et al., 2016) is used as the activation function. We do not dropout between GNN layers.

DNA (Fey, 2019): DNA leverages the jumping knowledge network to enhance the performance of GNNs. This approach enables selective and node-adaptive aggregation of neighboring embeddings, even when they have different localities within the graph. We set the number of hidden neurons to 128, the number of heads to 8, and the number of layers to 4. ReLU (Nair and Hinton, 2010) is used as the activation function. We set the dropout rate to 0.5 between GNN layers.

FiLM (Brockschmidt, 2020): FiLM learns embeddings of nodes in the network by training a linear message function that is conditioned on the features of the neighboring nodes. This allows FiLM to effectively capture and incorporate contextual information from the neighbors into the node embeddings. We set the number of hidden neurons to 320 and the number of layers to 4. We set the dropout rate to 0.1 between GNN layers.

FAGCN (Bo et al., 2021): FAGCN adopts a self-gating attention mechanism to learn the proportion of low-frequency and high-frequency signals. By adaptively modelling the frequency signals, FAGCN achieves enhanced expressive performance in capturing graph structure and features. We set the number of hidden neurons to 16, and the number of layers to 4. We set the dropout rate to 0.5 between GNN layers. **GATv2Conv** (Brody et al., 2021): GATv2 introduces a dynamic graph attention variant that reorders the internal operations, resulting in a significantly higher level of expressiveness compared to GAT. We set the number of hidden neurons to 64 per attention head and the number of layers to 3. ELU (Clevert et al., 2016) is used as the activation function. We do not dropout between GNN layers.

Model	Macro-F1 †/ §	Average Recall †/ §
GCN	**47.20 ± 1.54** / 46.52 ± 1.52	**48.13 ± 1.19** / 47.60 ± 1.16
GAT	**54.99 ± 4.14** / 53.65 ± 2.79	**57.15 ± 4.05** / 55.85 ± 2.27
GraphSage	46.54 ± 1.65 / **47.86 ± 1.38**	49.09 ± 0.87 / **50.91 ± 1.07**
SGC	44.60 ± 2.41 / **45.16 ± 2.29**	45.82 ± 0.73 / **46.03 ± 1.80**
DNA	**34.93 ± 3.95** / 33.88 ± 1.52	**36.71 ± 3.83** / 35.35 ± 1.68
FiLM	51.06 ± 2.24 / **51.47 ± 2.47**	51.35 ± 2.16 / **52.13 ± 2.01**

Table 10: Summary of the MGM results detailing performance variations between using full memory (†) and a reduced 90% memory allocation (§) for each GNN across Fact-2018 task. The best performance per base model is marked in boldface.

C Experimental Settings

As mentioned in Section 4.2, MGM is trained using the variational EM, which iteratively maximizes the ELBO and the expectation of log-likelihood function through an E-step and an M-step. To optimize the model, we use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. The early stopping strategy is implemented with patience of 10 epochs. In each experiment, we train MGM for 50 iterations to obtain the results. In order to encourage a sparse node selection distribution, we set the Dirichlet hyper-parameter α to 0.1. The hyper-parameter K, which determines the number of global similar nodes, is selected from the range $[1, 7]$ through a tuning process. Its value is optimized to achieve the best performance in the validation set for the node classification task. Similarly, the trade-off hyper-parameter η, which strikes a balance between the utilization of local representations and the information from global similar nodes, is chosen from the range $[0.6, 1]$ and is tuned to obtain the optimal performance in the validation set for the node classification task. The model is trained for 5 epochs using different random seeds and mean ± standard deviation is reported. We use the GNN module implementations provided by PyTorch Geometric[6] (Fey and Lenssen, 2019).

Evaluation Measures We evaluate our frameworks using mean of three key measures: *Macro-F1, Accuracy,* and *Average Recall. Macro-F1* balances precision and recall for each class, ideal for imbalanced datasets. *Accuracy* measures overall correctness, while *Average Recall* highlights the model's sensitivity to different classes. For GNNs experiment, we used an Nvidia 2080 Ti GPU, and for PLMs experiment, we used an NVIDIA A6000 48GB GPU.

[6] https://github.com/pyg-team/pytorch_geometric/tree/master/examples

Model	Macro-F1	Average Recall
Majority class	22.47 ± 0.00	33.33 ± 0.00
SVM	41.78 ± 0.00	48.89 ± 0.00
GCN	48.63 ± 2.19	48.16 ± 2.49
+ MGM	**49.21 ± 1.54**	**51.13 ± 1.19**
GAT	46.63 ± 3.53	52.25 ± 4.20
+ MGM	**54.99 ± 4.14**	**57.15 ± 4.05**
GraphSAGE	41.77 ± 0.22	48.65 ± 0.22
+ MGM	**47.86 ± 1.38**	**50.91 ± 1.07**
SGC	41.06 ± 0.35	44.91 ± 0.43
+ MGM	**45.16 ± 2.29**	**46.03 ± 1.80**
DNA	28.24 ± 1.23	33.26 ± 1.03
+ MGM	**34.93 ± 3.95**	**36.71 ± 3.83**
FiLM	46.75 ± 0.79	50.92 ± 1.36
+ MGM	**51.47 ± 2.47**	**52.13 ± 2.01**

Table 11: Performance of GNN baselines and their MGM enhanced versions on the Fact task of EMNLP-2018, with the majority class baseline and SVM included as naive and non-graphical methods. The highest performance is highlighted in bold.

D Results on Dataset EMNLP-2018

This section presents the results of the MGM model on the factuality task from the EMNLP-2018 dataset, supplementing our evaluation of MGM's adaptability across various datasets and tasks. Table 11 shows that MGM is able to enhance GNNs for news media graphs. All the GNNs baselines are consistently improved with MGM. Given the reasons mentioned in Appendix A, we did not conduct experiments on the political bias task of EMNLP-2018.

The experimental results in Table 10 demonstrate that MGM using sampled memory achieves performance comparable to MGM with full memory. Using sampled memory, all baselines perform competitively compared to the use of full memory. This suggests that sampled memory effectively captures crucial information while mitigating computational and storage overhead compared to full memory.

E Collecting Articles & Wikipedia

*Articles.*The article collection involves the following steps: (i) We obtained media sources from the ACL-2020 dataset. (ii) During the article link parsing, we parsed front-page article links from these

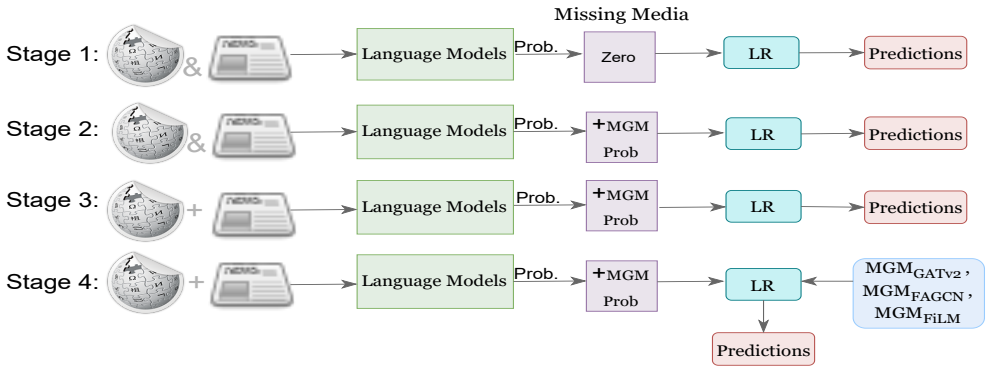

Figure 3: The pipeline of Integrating MGM with PLMs. **Stage 1:** We use logistic regression (meta-learner) to make predictions on probabilities obtained from PLMs on 472 media sources. For the remaining media sources, we assign $[0.0, 0.0, 0.0]$ probabilities. **Stage 2:** We use the probabilities produced by PLMs and, for the missing ones, we integrate probabilities from the best GNN MGM$_{GATv2}$. Logistic regression is then used to make predictions. **Stage 3:** We concatenate the probabilities of the best PLMs on *Wikipedia* and *Articles* and use logistic regression to make predictions. **Stage 4:**We use the probabilities obtained from Stage 3, which involve concatenating these probabilities with those generated by three GNNs (MGM$_{FiLM}$, MGM$_{FAGCN}$, and MGM$_{GATv2}$) across five different run seeds. Subsequently, logistic regression is employed to make predictions, and the scores are calculated using standard deviation.

media sources based on criteria of selecting only internal links with more than 65 characters and excluding menu button links. (iii) In the article collection stage, we use the selected article links to retrieve the titles and full text of the articles, using scripts and manual testing to ensure effective text extraction, with up to 30 news articles per media. (iv) Finally, the post-processing stage involved formatting the collected data in the JSON format. Moreover, we specifically targeted sections that focused on political, economic, and social issues sections.

Wikipedia. We started by searching for the media outlet name on the Internet to find the *Wikipedia* link. We ensured the link directed to a *Wikipedia* page specifically about the media outlet. We then retrieved the text from the *Wikipedia* page using its consistent HTML format. Finally, the post-processing stage involved formatting the collected data in the required JSON format.

In total, from 859 media sources, we have collected data from 472 media sources with *Articles* and *Wikipedia*. Table 8 provides detailed statistics. Moreover, Figure 3 provides our detailed pipeline for integrating MGM with PLMs.