

SSPC: Semantic-Structural Propagation Calibration for Multimodal Fake News Detection

Zejiang He^{1,2†}, Yun Luo^{1,2,3†}, Zhen Huang^{1,2*}, Zhiliang Tian^{1,2*}, Jingyuan Huang^{1,2},
Shanshan Liu^{1,2}, Yanzhou Ma³, Zhaoning Zhang^{1,2}, Dongsheng Li^{1,2}

¹College of Computer Science and Technology, National University of Defense Technology

²National Key Laboratory of Parallel and Distributed Computing

³College of Foreign Languages, National University of Defense Technology

{hezejiang, ly24, huangzhen, tianzhiliang, jingyuanhuang, liushanshan17,
myz827, zhangzhaoning, dsli}@nudt.edu.cn

Abstract

Multimodal fake news detection predicts post veracity from text, images, and inter-post context. Existing content-centric and graph-propagation methods model post evidence and inter-post context, but graph propagation still assumes connected posts provide compatible veracity evidence. In multimodal fake news graphs, edges often represent post relatedness rather than veracity consistency, so propagation can pass misleading evidence across conflicting posts. In this paper, we propose Semantic-Structural Propagation Calibration (SSPC), which uses noisy LLM pseudo-labels to calibrate where and how much propagation should be trusted, rather than using them as direct labels. SSPC refines graph edges from high-confidence pseudo-label agreement, computes a graph-encoded base prediction and a graph-propagated prediction, and learns node-level propagation reliance with a cross-fitted utility gate. Experiments on Twitter, PHEME, and Weibo show state-of-the-art (SOTA) performance over strong multimodal and graph-propagation fake news detection baselines.

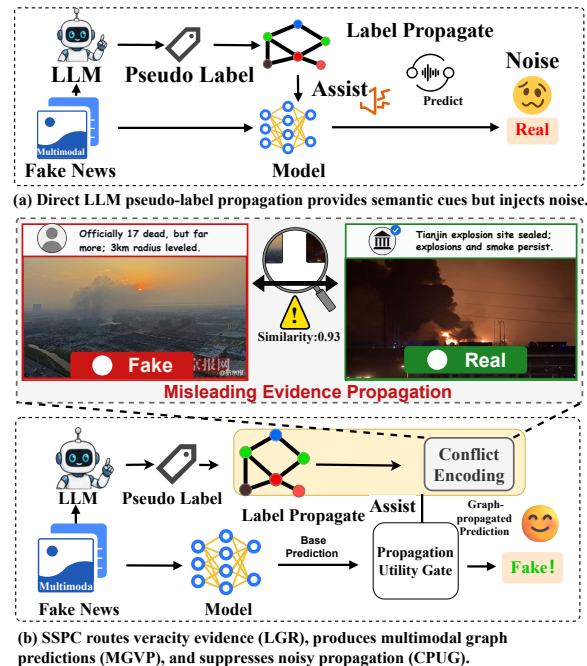


Figure 1: Motivation of SSPC. Noisy pseudo-labels can amplify errors across veracity-conflicting links; SSPC calibrates propagation paths and node reliance.

1 Introduction

Fake news on social media is often conveyed through multimodal posts that combine text, images, and inter-post context. Multimodal fake news detection aims to predict the veracity of each post and supports applications such as misinformation monitoring. Existing methods mainly combine multimodal representation learning with graph-based propagation to capture both post-level evidence and inter-post context.

Early studies mainly adopt content-centric methods, which encode textual, visual, or cross-modal features to detect fake news from the post itself (Wang et al., 2018; Zhou et al., 2020). These methods are effective at capturing post-level content

evidence, but they often miss inter-post context, such as supporting, contradicting, or corrective information from related posts. To address this limitation, recent studies propose graph-based methods that construct social graphs, propagation graphs, or similarity graphs to model inter-post context, aggregate neighboring representations through graph neural networks, and directly propagate predictions or label information over the graph through label propagation (Lu and Li, 2020; Bian et al., 2020; Zhao et al., 2023). Despite these improvements, graph-based propagation remains vulnerable in low-homophily or heterophilous neighborhoods (Pei et al., 2020; Zhu et al., 2020). Message passing and label propagation implicitly assume that connected posts provide compatible veracity evidence, but edges in multimodal fake news graphs often re-

[†] contributed equally to this work

^{*} corresponding authors

flect post relatedness, such as shared images, event co-occurrence, or semantic similarity, rather than veracity agreement. Thus, rumor and debunking posts may be connected despite opposite stances. We call this mismatch **semantic-structural conflict** (illustrated in Figure 1), under which propagation can transfer misleading evidence from veracity-inconsistent neighbors. Appendix Section I provides an empirical conflict analysis across overall, same-image, and same-event edges.

This issue suggests that the key challenge is not merely how to incorporate more neighboring evidence through graph propagation, but how to prevent unreliable neighboring evidence from misleading prediction. In multimodal fake news graphs, edges usually represent shared images, event co-occurrence, or semantic similarity; these links indicate that posts are related, but they do not guarantee consistency in veracity judgment. This leads to two practical questions. First, **which graph edges should support propagation?** Since graph edges may connect semantically conflicting posts, propagating over all edges can transfer misleading label evidence to the target node. Second, **how much should each node rely on propagation?** Some nodes benefit from inter-post context, while others are harmed by noisy propagation, so a uniform propagation strength is suboptimal. Recent work has introduced large language model (LLM) pseudo-labels into label propagation for multimodal fake news detection, using them as auxiliary signals for graph-based prediction (Hu et al., 2025). However, fake news judgments often require event background, platform context, and propagation evidence that an LLM may not observe. Directly using LLM pseudo-labels as training labels can therefore inject noisy and overconfident errors. The core challenge is how to use noisy LLM pseudo-labels as weak propagation-calibration signals to learn both propagation paths and node-level propagation strength without treating them as ground-truth labels.

In this paper, we propose **Semantic-Structural Propagation Calibration** (SSPC), which calibrates graph propagation in multimodal fake news detection by learning a two-level propagation policy. At the edge level, **Veracity-aware Learnable Graph Refinement** (LGR) estimates where veracity evidence should flow by adjusting graph edges according to high-confidence pseudo-label consistency. At the node level, the **Cross-fitted Semantic-Structural Propagation Utility Gate**

estimates how much each node should trust propagated predictions relative to the graph-encoded base prediction. In both levels, LLM pseudo-labels serve as weak calibration signals for propagation reliability, not as direct training labels.

Overall, SSPC shifts the role of graph propagation from unconditional label smoothing to calibrated evidence transfer. It converts noisy LLM pseudo-labels into edge-level routing and node-level reliance estimation signals. On Twitter, PHEME, and Weibo, SSPC achieves SOTA performance over strong multimodal and graph-propagation baselines.

Our contributions are three-fold: (1) We formulate multimodal fake news detection under semantic-structural conflict as a propagation calibration problem, shifting the focus from how to propagate labels to when, where, and how much propagation should be trusted. (2) We propose SSPC, which converts LLM pseudo-labels into weak propagation-calibration signals through edge-level evidence routing and node-level propagation reliance estimation, rather than using them as direct supervision. (3) Experiments show that SSPC achieves SOTA performance across multiple datasets, validating the effectiveness of propagation calibration over direct graph propagation and LLM-assisted propagation.

2 Related Work

2.1 Multimodal Fake News Detection

Multimodal fake news detection predicts post veracity from textual, visual, and contextual cues. Content-centric methods strengthen post-level evidence by learning event-invariant or latent multimodal representations, using pretrained encoders and cross-modal interaction, or modeling ambiguity, causality, contrastive alignment, and optimal transport (Singhal et al., 2019; Zhou et al., 2020; Wu et al., 2021; Chen et al., 2022, 2023; Lu et al., 2023; Shen et al., 2024; He et al., 2025). These methods improve local predictors, but they mainly reason from each post and do not decide when inter-post relations should propagate veracity evidence.

2.2 Graph-based Propagation and Label Propagation

Graph-based fake news detection uses relations among users, publishers, posts, sources, or diffusion paths to provide context beyond local content (Shu et al., 2019; Lu and Li, 2020; Nguyen

et al., 2020; Bian et al., 2020; Dou et al., 2021; Huang et al., 2022). Multimodal graph methods further construct cross-modal or semantic post graphs and propagate labels or evidence over related posts (Zhao et al., 2023; Hu et al., 2025). This line is also related to classic label propagation, local-global consistency, and recent correction/smoothing methods for graph-based prediction (Zhu and Ghahramani, 2002; Zhou et al., 2003; Huang et al., 2021). However, most methods treat graph neighborhoods as evidence to aggregate; SSPC instead learns propagation reliability, deciding which edges should carry veracity evidence and which nodes should reduce reliance on propagated predictions under semantic-structural conflict.

2.3 LLM Pseudo-labels and Weak Propagation Calibration

LLM pseudo-labels extend weak supervision, where data programming and Snorkel combine noisy labeling sources but noisy-label learning warns that incorrect labels can harm generalization (Ratner et al., 2016, 2020; Song et al., 2023). LLMs serve as knowledge advisors and data synthesizers (Tian et al., 2025; Liu et al., 2025; Lu et al., 2026), while also providing annotations or auxiliary signals for social media, language, graph, and fake news tasks, including ChatGPT-based annotation, label-free node classification, RumorLLM, and LLM-assisted multimodal graph propagation (Gilardi et al., 2023; Chen et al., 2024; Lai et al., 2024; Hu et al., 2025). Beyond the tasks above, LLMs also support applications such as graph-organized prompting for low-resource machine translation (Pan et al., 2024) and safety-critical methods (Pan et al., 2025, 2026). Yet fake-news veracity may depend on event background, platform context, source credibility, and propagation evidence unavailable to the LLM, while hallucination or overconfidence can add noise (Ji et al., 2023). SSPC uses pseudo-labels for propagation calibration: unlike methods that inject them into global or node-level propagation (Zhao et al., 2023; Hu et al., 2025), LGR adjusts edge routes and CPUG estimates node reliance on propagated predictions.

3 Method

3.1 Overview

We propose **SSPC**, a Semantic-Structural Propagation Calibration framework for multimodal fake news detection that uses noisy LLM pseudo-

labels as weak propagation-calibration signals rather than training labels. As illustrated in Figure 2, SSPC contains three connected modules organized as a graph-calibration, veracity-prediction, and adaptive-fusion pipeline. (1) The **Veracity-aware Learnable Graph Refinement** module (LGR; § 3.2) implements edge-level routing by transforming high-confidence LLM pseudo-label agreement and disagreement into learnable propagation-path adjustments, producing the refined graph used by MGVP. (2) The **Multimodal Graph Veracity Predictor** module (MGVP; § 3.3) performs veracity prediction on the refined graph through a dual-evidence design, separating the graph-encoded base prediction from the explicitly graph-propagated prediction. (3) The **Cross-fitted Semantic-Structural Propagation Utility Gate** module (CPUG; § 3.4) implements node-level reliance estimation by predicting how much each node should trust graph-propagated predictions relative to the graph-encoded base prediction, and adaptively fuses the two MGVP predictions into the final veracity distribution.

Given multimodal post features, a post graph, labeled training nodes, and noisy LLM pseudo-labels, SSPC first calibrates propagation paths with LGR (§ 3.2), then computes the graph-encoded base prediction and the explicitly graph-propagated prediction with MGVP (§ 3.3), and finally estimates node-level propagation reliance with CPUG (§ 3.4) to output each post’s veracity distribution.

Problem Formulation. Given a post graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ with N nodes, each node v_i has a multimodal feature vector $\mathbf{x}_i \in \mathbb{R}^d$. For labeled training nodes $v_i \in \mathcal{V}_L$, the ground-truth veracity label is $y_i \in \{0, 1\}$. For nodes with LLM pseudo-labels, we denote the pseudo-label and confidence score as $\tilde{y}_i \in \{0, 1\}$ and $c_i \in [0, 1]$; if unavailable, we set $\tilde{y}_i = -1$ and $c_i = 0$. The goal is to learn

$$f_{\Theta} : (\mathbf{X}, \mathcal{G}, \mathbf{Y}_L, \tilde{\mathbf{y}}, \mathbf{c}) \mapsto \mathbf{p}^{final}, \quad (1)$$

where $\mathbf{p}_i^{final} \in [0, 1]^2$ is the final veracity distribution for node v_i .

3.2 Veracity-aware Learnable Graph Refinement (LGR)

To calibrate propagation when post relatedness does not imply veracity consistency, we construct **LGR** as a veracity-aware learnable graph refinement that can adapt evidence routing to edge reliability. Rather than applying a fixed graph re-

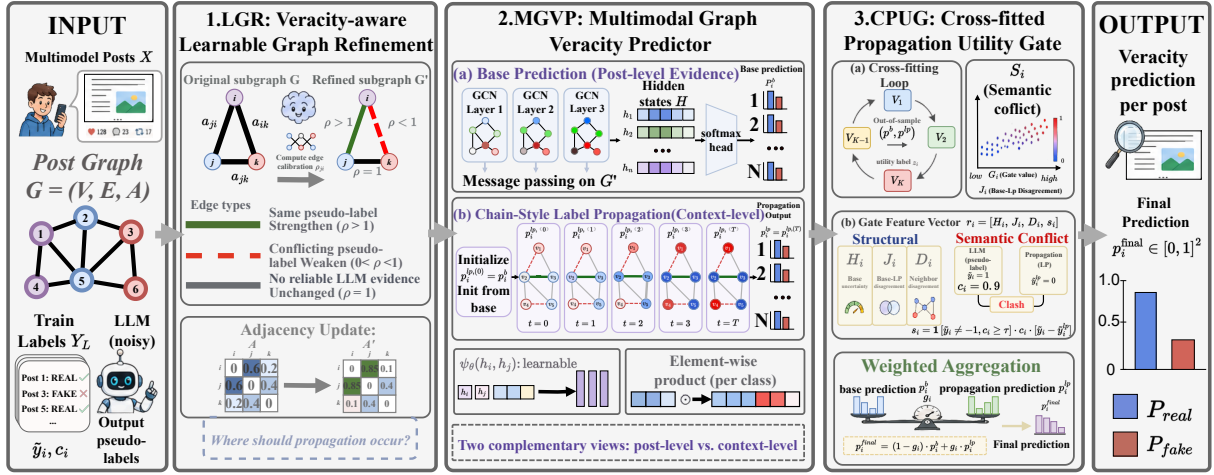


Figure 2: Overview of our method. (a) **LGR**: High-confidence LLM pseudo-label agreement and disagreement are converted into learnable edge-refinement coefficients that strengthen or weaken propagation edges, yielding the refined graph. (b) **MGVP**: MGVP computes a graph-encoded base prediction from post-level evidence and an explicitly graph-propagated prediction via chain-style label propagation over the refined graph. (c) **CPUG**: CPUG estimates per-node propagation reliance from base uncertainty, base-propagation disagreement, neighborhood disagreement, and semantic conflict, and fuses the two predictions into the final veracity prediction.

finement rule, LGR learns how strongly each edge should be adjusted from pseudo-label agreement, disagreement, and confidence, enabling adaptive propagation calibration across datasets and events. The motivation is that a graph edge may indicate shared images, similar events, or related propagation patterns, but such post relatedness does not necessarily imply veracity consistency. Therefore, LLM pseudo-labels should not be used as training labels; instead, they are used to calibrate how graph edges should support propagation.

Given the original graph $\mathcal{G}$ from § 3.1, we first identify nodes with confidence-filtered LLM pseudo-labels that can affect graph refinement:

$$m_i = \mathbb{I}[\tilde{y}_i \neq -1, c_i \geq \tau], \quad (2)$$

where τ is the LLM confidence threshold. For an edge $(v_j, v_i) \in \mathcal{E}$ with original weight a_{ji} , we define two semantic edge indicators:

$$u_{ji}^{same} = m_j m_i \mathbb{I}[\tilde{y}_j = \tilde{y}_i], \quad (3)$$

$$u_{ji}^{diff} = m_j m_i \mathbb{I}[\tilde{y}_j \neq \tilde{y}_i]. \quad (4)$$

Then, to avoid manually fixing the strengthening and weakening factors, LGR learns a differentiable edge-refinement coefficient from semantic consistency and confidence:

$$\begin{aligned} \rho_{ji} = & (1 - u_{ji}^{same} - u_{ji}^{diff}) \\ & + u_{ji}^{same} (1 + \text{softplus}(s_\phi(\mathbf{e}_{ji}))) \\ & + u_{ji}^{diff} \text{sigmoid}(d_\phi(\mathbf{e}_{ji})), \end{aligned} \quad (5)$$

where $\mathbf{e}_{ji} = [a_{ji}, c_j, c_i, u_{ji}^{same}, u_{ji}^{diff}]$ is the edge-calibration feature, and $s_\phi(\cdot)$ and $d_\phi(\cdot)$ are learnable functions. The unnormalized refined weight is then self-looped and incoming-normalized:

$$\begin{aligned} \tilde{a}_{ji} &= a_{ji} \rho_{ji}, \quad \tilde{a}_{ii} = 1, \\ a'_{ji} &= \frac{\tilde{a}_{ji}}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \tilde{a}_{ki}}. \end{aligned} \quad (6)$$

Thus, high-confidence edges with the same LLM pseudo-label can be strengthened, high-confidence edges with different LLM pseudo-labels can be weakened, and other edges keep their original unnormalized weights. The self-loop preserves node-local evidence, while incoming normalization keeps $\sum_{j \in \mathcal{N}(i) \cup \{i\}} a'_{ji} = 1$ and limits degree-driven dominance. The resulting LGR-refined graph is denoted as $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mathbf{A}')$ and is used by MGVP for multimodal graph veracity prediction.

This learnable formulation keeps the conservative behavior of rule-based graph refinement while avoiding fixed global coefficients. It lets the model adjust how LLM pseudo-labels should affect graph-based propagation under the task loss, which is important because LLM confidence and pseudo-label reliability vary across datasets and events.

3.3 Multimodal Graph Veracity Predictor (MGVP)

To compare local content with propagated context, we construct **MGVP** as a multimodal graph veracity predictor that can produce comparable graph-

encoded base predictions and explicitly graph-propagated predictions. The motivation is that some posts contain enough textual or visual cues for local veracity prediction, while others need evidence from related posts, such as supporting reports, contradictions, or corrections. Therefore, SSPC keeps both predictions and lets the propagation utility gate decide how much each node should rely on propagation.

Specifically, MGVP first encodes multimodal node features through graph convolution on the LGR-refined graph:

$$\mathbf{H} = f_{\theta}(\mathbf{X}, \mathcal{G}'), \quad (7)$$

where $\mathbf{h}_i$ denotes the hidden representation of node v_i . The graph-encoded base prediction is obtained from the hidden representation:

$$\mathbf{p}_i^b = \text{softmax}(\mathbf{W}_o \mathbf{h}_i). \quad (8)$$

Then, to incorporate inter-post context, MGVP initializes label propagation with $\mathbf{p}_i^{lp,(0)} = \mathbf{p}_i^b$ and propagates predictions over $\mathcal{G}'$ for T iterations:

$$\begin{aligned} \alpha_{ji} &= \frac{a'_{ji} \psi_{\theta}(\mathbf{h}_i, \mathbf{h}_j)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} a'_{ki} \psi_{\theta}(\mathbf{h}_i, \mathbf{h}_k)}, \\ \mathbf{p}_i^{lp,(t+1)} &= (1 - \lambda) \mathbf{p}_i^b + \lambda \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ji} \mathbf{p}_j^{lp,(t)}. \end{aligned} \quad (9)$$

where $\psi_{\theta}(\mathbf{h}_i, \mathbf{h}_j) = \text{sigmoid}(\text{MLP}_{\theta}([\mathbf{h}_i; \mathbf{h}_j])) \in (0, 1)$ is a scalar pairwise propagation coefficient shared across classes, and $\lambda \in [0, 1]$ controls propagation strength. When $\psi_{\theta} = 1$, the neighbor term reduces to label propagation over $\mathbf{A}'$; the learned ψ_{θ} makes it reliability-aware message passing over class-probability vectors. MGVP outputs

$$(\mathbf{p}_i^b, \mathbf{p}_i^{lp}) = \text{MGVP}(\mathbf{x}_i, \mathcal{G}'), \quad (10)$$

where $\mathbf{p}_i^{lp} = \mathbf{p}_i^{lp,(T)}$.

Keeping both predictions is necessary because propagation is not uniformly useful. The graph-encoded base prediction $\mathbf{p}^b$ is obtained from GCN message passing on the refined graph before explicit propagation and thus already reflects local neighborhood information, while the explicitly graph-propagated prediction $\mathbf{p}^{lp}$ is initialized from $\mathbf{p}^b$ and updated through T label-propagation iterations over the refined graph; the next module learns when the propagation prediction should override or complement the base prediction.

3.4 Cross-fitted Semantic-Structural Propagation Utility Gate (CPUG)

To estimate how much each node should trust propagation, we construct **CPUG** as a cross-fitted semantic-structural propagation utility gate that can learn node-specific reliance on graph-propagated evidence. Instead of simply lowering propagation weights for conflict nodes with a fixed rule after prediction, CPUG integrates semantic-structural conflict into both gate features and gate objective.

Given MGVP predictions $(\mathbf{p}_i^b, \mathbf{p}_i^{lp})$, we first construct utility supervision through cross-fitting. The labeled training nodes are split into K folds:

$$\mathcal{V}_L = \bigcup_{k=1}^K \mathcal{V}_k, \quad \mathcal{V}_k \cap \mathcal{V}_{k'} = \emptyset \quad (k \neq k'). \quad (11)$$

For each fold $\mathcal{V}_k$, SSPC trains LGR and MGVP on $\mathcal{V}_L \setminus \mathcal{V}_k$ and generates out-of-sample predictions $(\mathbf{p}_i^b, \mathbf{p}_i^{lp})$ for $v_i \in \mathcal{V}_k$. Propagation-utility label is

$$z_i = \mathbb{I} \left[\text{CE}(y_i, \mathbf{p}_i^{lp}) + m < \text{CE}(y_i, \mathbf{p}_i^b) \right], \quad (12)$$

where m is a margin requiring propagation to be meaningfully better than the base prediction.

Then, to represent each node for gate learning, CPUG builds a semantic-structural utility feature. The structural part contains base uncertainty, base-propagation disagreement, and neighborhood disagreement:

$$H_i = H(\mathbf{p}_i^b), \quad (13)$$

$$J_i = \text{JS}(\mathbf{p}_i^b, \mathbf{p}_i^{lp}), \quad (14)$$

$$D_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \text{JS}(\mathbf{p}_i^{lp}, \mathbf{p}_j^{lp}). \quad (15)$$

The semantic-structural conflict part compares the LLM pseudo-label with the propagation prediction:

$$\hat{y}_i^{lp} = \arg \max_{c \in \{0,1\}} \mathbf{p}_{i,c}^{lp}, \quad (16)$$

$$s_i = \mathbb{I}[\tilde{y}_i \neq -1, c_i \geq \tau] c_i |\tilde{y}_i - \hat{y}_i^{lp}|. \quad (17)$$

The final gate feature is the four-dimensional vector

$$\mathbf{r}_i = [H_i, J_i, D_i, s_i]. \quad (18)$$

CPUG predicts the propagation gate by

$$g_i = q_{\omega}(\mathbf{r}_i), \quad g_i \in [0, 1]. \quad (19)$$

Finally, to couple semantic-structural conflict with propagation utility learning, CPUG optimizes

$$\mathcal{L}_{gate} = \frac{1}{|\mathcal{V}_L|} \sum_{v_i \in \mathcal{V}_L} [\text{BCE}(g_i, z_i) + \gamma s_i g_i], \quad (20)$$

where γ controls the strength of conflict-aware gate regularization. The first term learns whether propagation improves held-out predictions, while the second term discourages high propagation reliance under semantic-structural conflict. If the LLM pseudo-label is unavailable or low-confidence, then $s_i = 0$ and CPUG falls back to structural propagation-utility features.

The resulting gate controls the final prediction:

$$\mathbf{p}_i^{final} = (1 - g_i)\mathbf{p}_i^b + g_i\mathbf{p}_i^{lp}. \quad (21)$$

This design turns semantic-structural conflict into a learned utility signal instead of using a fixed rule to reduce the gate after prediction.

3.5 Training Objective and Inference

To train the gate on predictions that match the inference setting, we train SSPC in three stages, which first builds held-out propagation-utility labels, then trains the gate, and finally retrain the predictor on all labeled nodes.

Phase A: cross-fitted utility construction. SSPC splits $\mathcal{V}_L$ into K folds. For each fold, it trains LGR and MGVP on $\mathcal{V}_L \setminus \mathcal{V}_k$ with the training objective

$$\mathcal{L}_{pred} = \mathcal{L}_{base} + \mathcal{L}_{lp} + \mu\mathcal{L}_{align}, \quad (22)$$

where $\mathcal{L}_{base}$ and $\mathcal{L}_{lp}$ are cross-entropy losses for $\mathbf{p}^b$ and $\mathbf{p}^{lp}$ on labeled training nodes, respectively. $\mathcal{L}_{align}$ denotes the event-split alignment loss inherited from the FCN-LP backbone, and μ is its weight. It aligns the feature distributions of seen-event and unseen-event samples within each class using MMD: $\mathcal{L}_{align} = \text{MMD}(X_s^r, X_u^r) + \text{MMD}(X_s^f, X_u^f)$, where X_s^r/X_u^r are real-news features from seen/unseen events, X_s^f/X_u^f are fake-news features from seen/unseen events, and $\text{MMD}(P, Q) = \|\text{mean}(P) - \text{mean}(Q)\|_2^2$. The trained fold model produces label-held-out predictions, utility labels z_i , and semantic-structural gate features $\mathbf{r}_i$ for the held-out fold. This controls gold-label leakage in gate supervision while retaining standard transductive access to node features, graph edges, and LLM pseudo-labels.

Phase B: semantic-structural gate training. After collecting all cross-fitted pairs $(\mathbf{r}_i, z_i)$, SSPC trains CPUG with $\mathcal{L}_{gate}$ and freezes the gate parameters ω . Freezing CPUG keeps the gate learned from cross-fitted utility targets and prevents it from being optimized on final in-sample predictions.

Phase C: final MGVP training and inference.

SSPC retrain LGR and MGVP on all labeled training nodes by minimizing $\mathcal{L}_{pred}$. Under the transductive evaluation graph, LGR first produces $\mathcal{G}'$, MGVP then obtains $(\mathbf{p}_i^b, \mathbf{p}_i^{lp})$, and the frozen CPUG computes g_i from $\mathbf{r}_i$ for inference. The final prediction is $\mathbf{p}_i^{final} = (1 - g_i)\mathbf{p}_i^b + g_i\mathbf{p}_i^{lp}$. Throughout all stages, LLM pseudo-labels are used only to refine propagation paths and gate features; they are never treated as training labels. Appendix Section B provides the full algorithm of SSPC.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on three widely used multimodal fake news detection benchmarks: Twitter (Boididou et al., 2015), PHEME (Zubiaga et al., 2017), and Weibo (Jin et al., 2017). Twitter and PHEME are English-language datasets collected from Twitter, while Weibo is a Chinese dataset from Sina Weibo. Following the standard splits adopted by label-propagation-based multimodal fake news detection methods (Zhao et al., 2023; Hu et al., 2025), we report results on the official or commonly used train/test partitions. Appendix Section C reports the split statistics and class distributions.

Baselines. We compare SSPC against three groups of baselines. The first group consists of multimodal detectors without explicit label propagation, including LLM (GPT-4o) (Hurst et al., 2024), EANN (Wang et al., 2018), Spot-Fake (Singhal et al., 2019), MVAE (Khattar et al., 2019), SAFE (Zhou et al., 2020), MCAN (Wu et al., 2021), HMCAN (Qian et al., 2021), and FCN (Zhao et al., 2023). The second group contains label-propagation-based variants built on HMCAN or CLIP (Radford et al., 2021) encoders. The third group comprises LLM-assisted label-propagation methods that directly incorporate LLM pseudo-labels during propagation, including the recent GLPN-LLM (Hu et al., 2025). All label-propagation-based methods follow the same transductive protocol: test-node features may participate in graph construction, and LLM-assisted methods may additionally use test-node pseudo-labels for propagation or calibration, but test labels are withheld. Following the evaluation setting of GLPN-LLM (Hu et al., 2025), GLPN-LLM was re-run with its official code and evaluation protocol, while

Method	Twitter				PHEME				Weibo			
	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$
LLM (GPT-4o)	75.39 $\pm$ 3.32	75.66 $\pm$ 3.44	80.92 $\pm$ 2.83	78.20 $\pm$ 5.66	74.38 $\pm$ 6.68	78.66 $\pm$ 5.31	75.16 $\pm$ 4.32	76.87 $\pm$ 5.65	80.86 $\pm$ 3.11	82.16 $\pm$ 2.86	81.33 $\pm$ 3.66	81.75 $\pm$ 2.95
EANN	71.53 $\pm$ 0.91	71.38 $\pm$ 1.23	63.82 $\pm$ 2.11	68.91 $\pm$ 1.58	70.17 $\pm$ 0.79	71.28 $\pm$ 1.32	67.36 $\pm$ 2.17	69.10 $\pm$ 1.83	79.18 $\pm$ 0.76	80.31 $\pm$ 1.23	78.52 $\pm$ 0.32	79.44 $\pm$ 2.13
SpotFake	77.16 $\pm$ 1.57	75.32 $\pm$ 1.14	87.83 $\pm$ 0.63	85.14 $\pm$ 0.07	81.37 $\pm$ 2.38	79.53 $\pm$ 2.27	81.22 $\pm$ 2.43	79.43 $\pm$ 0.75	86.39 $\pm$ 2.51	86.12 $\pm$ 0.53	87.17 $\pm$ 2.63	83.22 $\pm$ 1.41
MVAE	74.56 $\pm$ 1.58	80.15 $\pm$ 2.69	76.34 $\pm$ 0.83	81.57 $\pm$ 1.98	77.83 $\pm$ 1.27	73.82 $\pm$ 2.05	73.45 $\pm$ 2.62	72.21 $\pm$ 0.54	71.86 $\pm$ 0.25	70.32 $\pm$ 0.69	70.32 $\pm$ 2.84	70.53 $\pm$ 1.60
SAFE	76.66 $\pm$ 3.00	76.32 $\pm$ 1.94	75.41 $\pm$ 2.12	76.37 $\pm$ 2.85	81.25 $\pm$ 1.34	79.22 $\pm$ 2.76	79.11 $\pm$ 1.45	79.69 $\pm$ 2.67	84.91 $\pm$ 2.12	83.81 $\pm$ 1.58	82.19 $\pm$ 1.16	83.01 $\pm$ 1.70
MCAN	80.91 $\pm$ 2.33	82.68 $\pm$ 2.48	76.67 $\pm$ 0.94	82.26 $\pm$ 1.32	80.74 $\pm$ 1.89	79.21 $\pm$ 2.23	79.64 $\pm$ 1.53	80.15 $\pm$ 0.86	86.50 $\pm$ 3.00	88.10 $\pm$ 2.10	84.60 $\pm$ 1.80	86.15 $\pm$ 1.60
HMCAN	83.91 $\pm$ 1.49	81.68 $\pm$ 2.08	84.67 $\pm$ 1.21	82.57 $\pm$ 1.62	86.36 $\pm$ 1.83	83.18 $\pm$ 1.41	83.81 $\pm$ 2.51	83.49 $\pm$ 1.07	86.75 $\pm$ 2.95	88.40 $\pm$ 3.00	84.65 $\pm$ 1.80	87.20 $\pm$ 1.20
FCN	82.86 $\pm$ 1.27	78.64 $\pm$ 1.68	87.39 $\pm$ 0.85	82.78 $\pm$ 0.47	80.36 $\pm$ 1.93	84.43 $\pm$ 1.27	89.12 $\pm$ 0.12	86.71 $\pm$ 1.88	82.92 $\pm$ 0.54	83.17 $\pm$ 1.00	88.45 $\pm$ 2.13	86.74 $\pm$ 0.41
FCN-LP (HMCAN)	84.57 $\pm$ 1.62	83.58 $\pm$ 1.66	85.22 $\pm$ 2.06	84.04 $\pm$ 0.82	87.25 $\pm$ 1.18	84.48 $\pm$ 1.91	84.78 $\pm$ 1.95	84.50 $\pm$ 0.85	87.15 $\pm$ 1.32	88.82 $\pm$ 2.56	86.55 $\pm$ 1.98	88.11 $\pm$ 1.66
FCN-LP (CLIP)	85.32 $\pm$ 2.56	81.52 $\pm$ 2.82	89.32 $\pm$ 0.99	85.24 $\pm$ 1.93	84.68 $\pm$ 0.81	86.32 $\pm$ 1.55	89.85 $\pm$ 1.22	87.97 $\pm$ 0.88	84.47 $\pm$ 1.66	88.41 $\pm$ 0.26	91.18 $\pm$ 0.69	89.78 $\pm$ 0.84
FCN-LP (HMCAN) + LLM	85.30 $\pm$ 1.88	84.70 $\pm$ 2.25	86.30 $\pm$ 2.53	85.10 $\pm$ 0.98	87.35 $\pm$ 1.37	84.55 $\pm$ 2.76	85.85 $\pm$ 2.19	84.60 $\pm$ 1.28	87.55 $\pm$ 1.94	89.30 $\pm$ 2.85	88.60 $\pm$ 2.39	88.75 $\pm$ 1.97
FCN-LP (CLIP) + LLM	<u>85.93</u> $\pm$ 1.83	<u>81.92</u> $\pm$ 2.64	<u>90.44</u> $\pm$ 2.17	<u>85.97</u> $\pm$ 2.72	<u>84.89</u> $\pm$ 2.81	<u>87.80</u> $\pm$ 2.36	<u>90.55</u> $\pm$ 2.28	<u>89.21</u> $\pm$ 2.59	<u>84.88</u> $\pm$ 2.78	<u>88.55</u> $\pm$ 2.43	<u>91.68</u> $\pm$ 1.95	<u>89.85</u> $\pm$ 2.67
GLPN-LLM (HMCAN)	84.24 $\pm$ 1.37	84.77 $\pm$ 0.81	86.77 $\pm$ 2.06	84.13 $\pm$ 0.79	87.40 $\pm$ 0.44	85.26 $\pm$ 1.03	86.00 $\pm$ 0.97	83.17 $\pm$ 1.41	88.99 $\pm$ 1.18	88.18 $\pm$ 0.19	90.54 $\pm$ 0.24	90.70 $\pm$ 0.11
GLPN-LLM (CLIP)	85.65 $\pm$ 2.11	82.04 $\pm$ 2.68	90.78 $\pm$ 2.47	86.13 $\pm$ 2.36	85.82 $\pm$ 2.14	87.39 $\pm$ 2.51	90.15 $\pm$ 2.83	86.68 $\pm$ 2.09	85.29 $\pm$ 1.57	88.82 $\pm$ 0.73	<u>91.88</u> $\pm$ 0.79	90.55 $\pm$ 0.58
SSPC (HMCAN)	85.21 $\pm$ 1.17	86.79 $\pm$ 1.62	90.07 $\pm$ 1.90	<u>86.66</u> $\pm$ 0.53	89.68 $\pm$ 0.16	<u>88.27</u> $\pm$ 0.15	87.63 $\pm$ 0.05	87.74 $\pm$ 0.36	90.54 $\pm$ 0.46	87.51 $\pm$ 0.33	94.67 $\pm$ 0.37	<u>91.59</u> $\pm$ 0.41
SSPC (CLIP)	86.43 $\pm$ 2.54	83.53 $\pm$ 2.98	98.23 $\pm$ 0.94	88.50 $\pm$ 1.56	<u>87.46</u> $\pm$ 0.39	91.40 $\pm$ 0.31	90.97 $\pm$ 0.70	91.18 $\pm$ 0.31	86.83 $\pm$ 0.57	89.32 $\pm$ 1.37	<u>94.08</u> $\pm$ 1.31	91.64 $\pm$ 0.30

Table 1: Overall performance comparison on three benchmark datasets. Best results are **bold** and second-best are underlined. All values are percentages (%), and each metric is averaged independently over five random seeds. Our improvements over the strongest CLIP-based LLM-assisted baseline are statistically significant under the paired two-tailed t -test ($p < 0.05$; Appendix Section K).

Variant	Twitter				PHEME				Weibo			
	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$
SSPC	86.43 $\pm$ 2.54	83.53 $\pm$ 2.98	98.23 $\pm$ 0.94	88.50 $\pm$ 1.56	87.46 $\pm$ 0.39	91.40 $\pm$ 0.31	90.97 $\pm$ 0.70	91.18 $\pm$ 0.31	86.83 $\pm$ 0.57	89.32 $\pm$ 1.37	94.08 $\pm$ 1.31	91.64 $\pm$ 0.30
w/o CPUG	82.58 $\pm$ 2.74	80.44 $\pm$ 3.46	98.07 $\pm$ 3.20	86.54 $\pm$ 1.85	84.98 $\pm$ 0.22	89.20 $\pm$ 1.00	89.81 $\pm$ 1.47	89.50 $\pm$ 0.19	85.41 $\pm$ 0.78	88.18 $\pm$ 1.78	93.50 $\pm$ 1.90	90.77 $\pm$ 0.38
w/o LGR	82.29 $\pm$ 1.96	79.49 $\pm$ 1.72	95.54 $\pm$ 2.36	85.57 $\pm$ 1.51	83.33 $\pm$ 0.62	86.53 $\pm$ 1.35	90.74 $\pm$ 1.47	88.59 $\pm$ 0.25	85.21 $\pm$ 3.66	87.47 $\pm$ 3.99	92.32 $\pm$ 1.65	89.30 $\pm$ 1.77
w/o SCGC	83.50 $\pm$ 3.22	80.58 $\pm$ 3.32	96.68 $\pm$ 0.80	87.38 $\pm$ 2.22	83.99 $\pm$ 0.41	87.81 $\pm$ 0.34	90.05 $\pm$ 0.26	88.91 $\pm$ 0.25	84.96 $\pm$ 3.88	86.21 $\pm$ 2.27	92.71 $\pm$ 1.65	90.71 $\pm$ 1.88

Table 2: Ablation study on three benchmark datasets. Each variant removes one core SSPC component (CLIP encoder). All values are percentages (%).

the other baseline results are cited from the comparison table of the GLPN-LLM paper. Appendix Section D gives the baseline details.

Evaluation Metrics. We report Accuracy, Precision, Recall, and F1. All values are percentages (%). Main comparison results are reported as mean and standard deviation over five random seeds. F1 is macro-averaged over the fake and real news classes, following standard practice in fake news detection (Hu et al., 2025). Implementation details are given in Appendix Section E, and the LLM pseudo-label prompt template is provided in Appendix Section T.

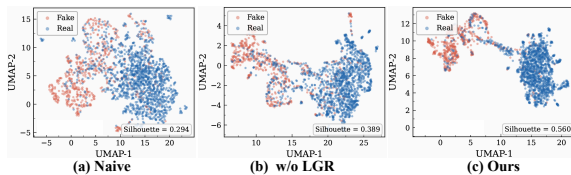


Figure 3: UMAP projections of PHEME test-set embeddings. (a) MLP without graph (Sil. = 0.294); (b) MGVP on original graph G without LGR (Sil. = 0.389); (c) MGVP on LGR-refined graph G' (Sil. = 0.560). LGR calibration substantially reduces class overlap and tightens intra-class clusters.

4.2 Overall Evaluation

Table 1 compares SSPC with multimodal, label-propagation, and LLM-assisted baselines on three benchmarks, where SSPC achieves the best result on all 12 dataset–metric pairs when the HMCAN and CLIP variants are considered together. Against the strongest non-SSPC baseline, SSPC improves F1 by 2.37, 1.97, and 0.94 points on Twitter, PHEME, and Weibo, with the largest metric gains on Twitter Recall (+7.45) and PHEME Precision (+3.60). The only marginal gain is Weibo Precision (+0.02), where strong LLM-assisted baselines are already highly precise; however, SSPC still improves Weibo Recall (+2.20) and F1 (+0.94), indicating broader veracity coverage without sacrificing precision. These results support using LLM pseudo-labels as calibration signals rather than direct propagation labels; statistical significance tests are reported in Appendix Section K, pseudo-label quality and robustness analyses in Appendix Sections F and L, and stricter inductive protocols in Appendix Section N. Appendix Sections Q and R further diagnose the learned edge-weight distribution and veracity awareness of LGR.

4.3 Ablation Study

To demonstrate the effectiveness of each SSPC component, we conduct an ablation study with

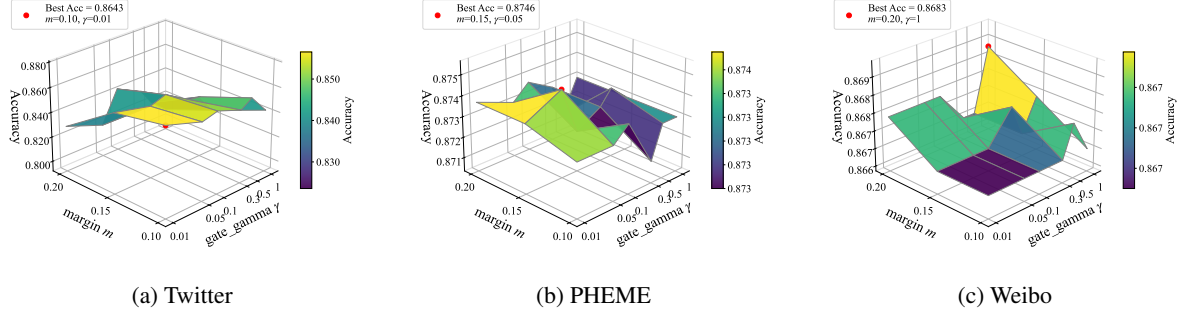


Figure 4: Accuracy surfaces over margin m and coupling weight γ on Twitter, PHEME, and Weibo. Surfaces are flat over a broad central region, indicating robustness to the joint choice of (m, γ) .

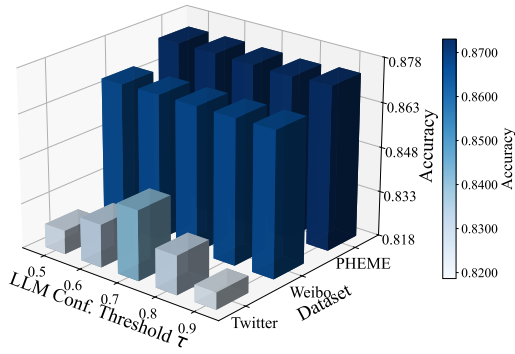


Figure 5: Accuracy versus LLM confidence threshold τ across three datasets. A moderate threshold balances pseudo-label quality and coverage; very low values admit noisy signals, while very high values deplete useful calibration pairs.

three variants: **w/o LGR**: the variant without Veracity-aware Learnable Graph Refinement (§ 3.2); **w/o CPUG**: the variant without Cross-fitted Semantic-Structural Propagation Utility Gate (§ 3.4); **w/o SCGC**: the variant without semantic-structural conflict guidance in CPUG (§ 3.4).

As shown in Table 2, each module is essential and cannot be removed without performance degradation. Specifically, removing LGR causes the largest F1 drops, indicating that veracity-aware edge calibration is crucial for reliable propagation. Removing CPUG also degrades performance, confirming the importance of node-level propagation control, while w/o SCGC shows that semantic-structural conflict guidance helps the gate avoid misleading neighborhoods. Appendix Section M isolates the contribution of this cross-fitted gate protocol against in-sample and no-gate alternatives.

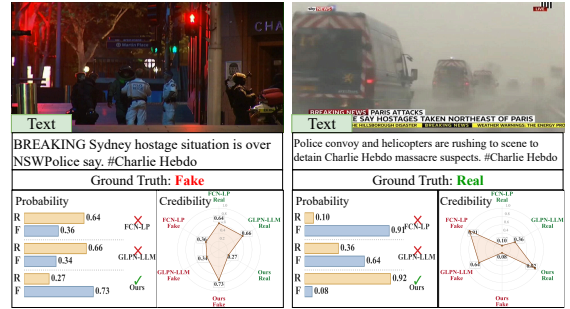


Figure 6: Case study on two PHEME posts. FCN-LP and GLPN-LLM fail in both cases, while SSPP predicts the correct fake/real labels by calibrating edge reliability and propagation reliance.

4.4 LGR Graph Refinement Effectiveness: UMAP Visualization

Figure 3 visualizes why LGR helps: local prediction and original-graph propagation leave PHEME embeddings mixed ($\text{Sil.} = 0.294/0.389$), whereas the LGR-refined graph yields clearer separation ($\text{Sil.} = 0.560$), indicating substantially more veracity-coherent propagation.

4.5 Hyperparameter Sensitivity Analysis

Figures 4 and 5 show that SSPP is stable under moderate choices of the margin m , coupling weight γ , and LLM confidence threshold τ . Accuracy remains flat in broad central regions, while extreme settings either introduce noisy utility or pseudo-label supervision or leave too little calibration signal; we therefore use $\tau = 0.7$ and report additional sweeps, runtime, and complexity in Appendix Sections J, O, and P.

4.6 Case Study

Figure 6 shows two PHEME cases where FCN-LP and GLPN-LLM are pulled to the wrong la-

bel, while SSPC predicts the ground truth with $P_{\text{fake}} = 0.73$ and $P_{\text{real}} = 0.92$. These qualitative examples suggest that edge calibration and node-level propagation control can reduce misleading propagation in mixed-veracity neighborhoods.

5 Conclusion

In summary, we proposed **SSPC** for multimodal fake news detection. SSPC uses LLM pseudo-labels as weak calibration signals rather than direct labels, refining graph edges with LGR and estimating node-level propagation reliance with CPUG. Experiments on Twitter, PHEME, and Weibo show that SSPC achieves SOTA performance under both HMCAN and CLIP encoders. The ablation, case, and homophily analyses further indicate that calibrating both edge reliability and node-level propagation strength helps reduce misleading propagation in mixed-veracity neighborhoods.

Limitations

Like existing graph-based label-propagation methods for multimodal fake news detection, SSPC is evaluated under a transductive graph setting where the post graph is available during inference. This setting follows prior work, but streaming deployment would require feature extraction, LLM pseudo-label generation when semantic calibration is used, and incremental graph construction or periodic graph updates.

SSPC depends on LLM pseudo-labels as weak calibration signals. Confidence filtering and gate learning reduce sensitivity to pseudo-label noise, but future work can combine multiple LLMs, retrieval evidence, or human-verified signals for more reliable calibration. CPUG introduces extra cross-fitting cost during training, although the learned gate is frozen and lightweight at inference time.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62376284, 62306330, and 62406332.

References

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. [Rumor detection on social media with bi-directional graph convolutional networks](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556.

Christina Boididou, Katerina Andreadou, Symeon Papadopoulos, Duc-Tien Dang-Nguyen, Giulia Boato, Michael Riegler, and Yiannis Kompatsiaris. 2015. [Verifying multimedia use at MediaEval 2015](#). In *Working Notes Proceedings of the MediaEval 2015 Workshop*, volume 1436 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Yixuan Chen, Dongsheng Li, Peng Zhang, Jie Sui, Qin Lv, Tun Lu, and Li Shang. 2022. [Cross-modal ambiguity learning for multimodal fake news detection](#). In *Proceedings of the ACM Web Conference 2022*, pages 2897–2905. ACM.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2024. [Label-free node classification on graphs with large language models \(LLMs\)](#). In *The Twelfth International Conference on Learning Representations*.

Ziwei Chen, Linmei Hu, Weixin Li, Yingxia Shao, and Liqiang Nie. 2023. [Causal intervention and counterfactual reasoning for multi-modal fake news detection](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 627–638, Toronto, Canada. Association for Computational Linguistics.

Yingtong Dou, Kai Shu, Congying Xia, Philip S. Yu, and Lichao Sun. 2021. [User preference-aware fake news detection](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2051–2055. ACM.

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

Zejiang He, Jingyuan Huang, Menglong Lu, Zhen Huang, Shanshan Liu, Zhiliang Tian, and Dongsheng Li. 2025. [GCML: Gradient coherence guided meta-learning for cross-domain emerging topic rumor detection](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19148–19162, Suzhou, China. Association for Computational Linguistics.

Shuguo Hu, Jun Hu, and Huaiwen Zhang. 2025. [Synergizing LLMs with global label propagation for multimodal fake news detection](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1440, Vienna, Austria. Association for Computational Linguistics.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R. Benson. 2021. [Combining label propagation and simple models out-performs graph neural networks](#). In *International Conference on Learning Representations*.

Zhen Huang, Zhilong Lv, Xiaoyun Han, Binyang Li, Menglong Lu, and Dongsheng Li. 2022. [Social bot-aware graph neural network for early rumor](#)

- detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6680–6690, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. [GPT-4o system card](#). *arXiv preprint arXiv:2410.21276*.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Computing Surveys*, 55(12):1–38.
- Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. 2017. [Multimodal fusion with recurrent neural networks for rumor detection on microblogs](#). In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 795–816.
- Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. [MVAE: Multimodal variational autoencoder for fake news detection](#). In *The World Wide Web Conference*, pages 2915–2921. ACM.
- Philipp Koehn. 2004. [Statistical significance tests for machine translation evaluation](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- Jianqiao Lai, Xinran Yang, Wenyue Luo, Linjiang Zhou, Langchen Li, Yongqi Wang, and Xiaochuan Shi. 2024. [RumorLLM: A rumor large language model-based fake-news-detection data-augmentation approach](#). *Applied Sciences*, 14(8):3532.
- Shanshan Liu, Menglong Lu, Zhen Huang, Zejiang He, Liu Liu, Zhigang Sun, and Dongsheng Li. 2025. [MONTROSE: LLM-driven Monte Carlo tree search self-refinement for cross-domain rumor detection](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21475–21487, Vienna, Austria. Association for Computational Linguistics.
- Menglong Lu, Zejiang He, Yaohui Guo, Shanshan Liu, Jingyuan Huang, Yunxiang Zhao, Zhiliang Tian, Xiaoran Zhao, Chengcheng Shao, Lin Deng, Dongsheng Li, and Zhen Huang. 2026. [LLM-driven adversarial example synthesis for emerging topic rumor detection on social media](#). *IEEE Transactions on Knowledge and Data Engineering*, 38:2340–2353.
- Menglong Lu, Zhen Huang, Yunxiang Zhao, Zhiliang Tian, Yang Liu, and Dongsheng Li. 2023. [DaMSTF: Domain adversarial learning enhanced meta self-training for domain adaptation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1668, Toronto, Canada. Association for Computational Linguistics.
- Yi-Ju Lu and Cheng-Te Li. 2020. [GCAN: Graph-aware co-attention networks for explainable fake news detection on social media](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online. Association for Computational Linguistics.
- Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. [FANG: Leveraging social context for fake news detection using graph representation](#). In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 1165–1174. ACM.
- Shilong Pan, Zhiliang Tian, Liang Ding, Haoqi Zheng, Zhen Huang, Zhihua Wen, and Dongsheng Li. 2024. [POMP: Probability-driven meta-graph prompter for LLMs in low-resource unsupervised neural machine translation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9976–9992, Bangkok, Thailand. Association for Computational Linguistics.
- Shilong Pan, Zhiliang Tian, Zhen Huang, Wanlong Yu, Zhihua Wen, Xinwang Liu, Kai Lu, Minlie Huang, and Dongsheng Li. 2025. [AGD: Adversarial game defense against jailbreak attacks in large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17391–17406, Vienna, Austria. Association for Computational Linguistics.
- Shilong Pan, Zhiliang Tian, Wanlong Yu, Zhen Huang, Qingyu Qiu, Zihan Chen, Zhonghao Sun, Minlie Huang, and Dongsheng Li. 2026. [WALKSAFE: Risk-aware graph random walk with Bi-GRPO for LLM safety](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32655–32663.
- Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. [Geom-GCN: Geometric graph convolutional networks](#). In *International Conference on Learning Representations*.
- Shengsheng Qian, Jinguang Wang, Jun Hu, Quan Fang, and Changsheng Xu. 2021. [Hierarchical multi-modal contextual attention network for fake news detection](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 153–162.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. [Snorkel: Rapid training data creation with weak supervision](#). *The VLDB Journal*, 29:709–730.
- Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. [Data programming: Creating large training sets, quickly](#). In

Advances in Neural Information Processing Systems, volume 29.

Xiaorong Shen, Maowei Huang, Zheng Hu, Shimin Cai, and Tao Zhou. 2024. [Multimodal fake news detection with contrastive learning and optimal transport](#). *Frontiers in Computer Science*, 6.

Kai Shu, Suhang Wang, and Huan Liu. 2019. [Beyond news contents: The role of social context for fake news detection](#). In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 312–320. ACM.

Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin’ichi Satoh. 2019. [SpotFake: A multi-modal framework for fake news detection](#). In *2019 IEEE Fifth International Conference on Multimedia Big Data*, pages 39–47. IEEE.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2023. [Learning from noisy labels with deep neural networks: A survey](#). *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153.

Zhiliang Tian, Jingyuan Huang, Zejiang He, Zhen Huang, Menglong Lu, Linbo Qiao, Songzhu Mei, Yijie Wang, and Dongsheng Li. 2025. [LLM-based rumor detection via influence guided sample selection and game-based perspective analysis](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28402–28414, Vienna, Austria. Association for Computational Linguistics.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. [EANN: Event adversarial neural networks for multi-modal fake news detection](#). In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 849–857. ACM.

Yang Wu, Pengwei Zhan, Yunjian Zhang, Liming Wang, and Zhen Xu. 2021. [Multimodal fusion with co-attention networks for fake news detection](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2560–2569, Online. Association for Computational Linguistics.

Wanqing Zhao, Yuta Nakashima, Haiyuan Chen, and Noboru Babaguchi. 2023. [Enhancing fake news detection in social media via label propagation on cross-modal tweet graph](#). In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2400–2408.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. [Learning with local and global consistency](#). In *Advances in Neural Information Processing Systems*, volume 16.

Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. [SAFE: Similarity-aware multi-modal fake news detection](#). In

Advances in Knowledge Discovery and Data Mining, volume 12085 of *Lecture Notes in Computer Science*, pages 354–367. Springer.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. [Beyond homophily in graph neural networks: Current limitations and effective designs](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 7793–7804.

Xiaojin Zhu and Zoubin Ghahramani. 2002. [Learning from labeled and unlabeled data with label propagation](#). Technical Report CMU-CALD-02-107, Carnegie Mellon University.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. [Exploiting context for rumour detection in social media](#). In *Social Informatics*, pages 109–123. Springer.

A Appendix Overview

This appendix supports the main paper from three perspectives: reproducibility, mechanism validation, and claim scoping. Section B gives the SSPC training and inference procedure. Sections C–E document datasets, baselines, and implementation settings. Sections F–N examine pseudo-label quality, case and conflict diagnostics, the additional PHEME case study in Section H, sensitivity analyses, statistical significance, pseudo-label robustness, cross-fitting, and stricter inductive diagnostics under restricted test-time information access. Sections O–R discuss detector-side cost and LGR edge-weight diagnostics, and Section S gives reduced-order mechanism analysis. Sections T–W report the prompt template, ethics statement, artifact license, and AI-assistant disclosure.

B Algorithm of SSPC

Algorithm 1 summarizes the full SSPC procedure. The training process contains three stages. First, SSPC obtains out-of-sample base and propagation predictions through cross-fitting, where LGR and MGVP are trained inside each fold. Second, CPUG is trained from semantic-structural gate features and propagation-utility labels. Third, LGR and MGVP are retrained on all labeled nodes, and the frozen CPUG produces the final propagation gate at inference time.

C Dataset Details

We evaluate SSPC on three public multimodal fake news datasets. **Twitter** (Boididou et al., 2015) is an English dataset from the MediaEval Verifying Multimedia Use benchmark. **PHEME** (Zubiaga et al.,

Algorithm 1 Training and Inference Procedure of SSPC

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, node features $\mathbf{X}$, training labels $\mathbf{Y}_L$, LLM pseudo-labels $\tilde{\mathbf{y}}$, confidence scores $\mathbf{c}$, folds K , confidence threshold τ , margin m , gate regularization weight γ

- 1: Split labeled training nodes $\mathcal{V}_L$ into K disjoint folds.
- 2: **for** each fold $\mathcal{V}_k$ **do**
- 3: Train learnable LGR and MGVP on $\mathcal{V}_L \setminus \mathcal{V}_k$.
- 4: Generate out-of-sample predictions $(\mathbf{p}_i^b, \mathbf{p}_i^{lp})$ for $v_i \in \mathcal{V}_k$.
- 5: Build utility labels z_i by comparing held-out base and propagation losses.
- 6: Build semantic-structural gate features $\mathbf{r}_i = [H_i, J_i, D_i, s_i]$.
- 7: **end for**
- 8: Train CPUG on $\{(\mathbf{r}_i, z_i)\}$ with $\mathcal{L}_{gate} = \text{BCE}(g_i, z_i) + \gamma s_i g_i$ and freeze it.
- 9: Retrain learnable LGR and MGVP with all labeled training nodes.
- 10: At inference on the evaluation graph, use LGR to obtain $\mathcal{G}'$ and MGVP to obtain $(\mathbf{p}_i^b, \mathbf{p}_i^{lp})$.
- 11: Compute $\mathbf{r}_i = [H_i, J_i, D_i, s_i]$ and gate value $g_i = \text{CPUG}(\mathbf{r}_i)$.
- 12: Return $\mathbf{p}_i^{final} = (1 - g_i)\mathbf{p}_i^b + g_i\mathbf{p}_i^{lp}$.

Dataset	Split	Total	Real	Fake	Real%	Fake%
Twitter	Train	13,366	5,819	7,547	43.5	56.5
	Test	1,091	468	623	42.9	57.1
PHEME	Train	1,412	996	416	70.5	29.5
	Test	606	432	174	71.3	28.7
Weibo	Train	4,140	3,345	795	80.8	19.2
	Test	1,124	862	262	76.7	23.3

Table 3: Dataset split statistics and class distributions.

2017) is an English rumor-detection dataset collected from five breaking-news events. **Weibo** (Jin et al., 2017) is a Chinese multimodal dataset from Sina Weibo, where fake news items are verified by the platform’s rumor-debunking system. Following prior label-propagation-based multimodal fake news detection protocols (Zhao et al., 2023; Hu et al., 2025), all methods use the same train/test partitions and the same standard transductive graph setting: test-node features may participate in graph construction, and LLM-assisted methods additionally use test-node pseudo-labels for propagation or edge calibration, but test labels are never used during training, calibration, or model selection.

D Baseline Details

We organize baselines into three groups, matching the experimental setup in the main paper.

Multimodal detectors without explicit label propagation. LLM (GPT-4o) (Hurst et al., 2024) directly predicts the veracity label from the post

input without task-specific training. EANN (Wang et al., 2018) learns event-invariant multimodal features with adversarial event discrimination. SpotFake (Singhal et al., 2019) combines pretrained textual and visual encoders for binary veracity classification. MVAE (Khattar et al., 2019) learns a shared multimodal latent representation with a variational autoencoder. SAFE (Zhou et al., 2020) uses cross-modal semantic similarity between image-text evidence. MCAN (Wu et al., 2021) models image-text interactions with co-attention. HMCAN (Qian et al., 2021) uses hierarchical contextual attention and is a strong multimodal encoder for subsequent label-propagation-based methods. FCN (Zhao et al., 2023) constructs a cross-modal tweet graph and uses graph convolution for node-level fake news prediction.

Label-propagation-based methods. The label-propagation-based baselines extend graph-based multimodal detectors by propagating labels or predictions over the cross-modal tweet graph (Zhao et al., 2023). We report **FCN-LP (HMCAN)** and **FCN-LP (CLIP)** to separate the influence of graph-based propagation from the influence of the multimodal encoder.

LLM-assisted label-propagation methods. This group augments label propagation with LLM pseudo-labels. Direct variants are denoted as **FCN-LP (HMCAN) + LLM** and **FCN-LP (CLIP) + LLM**, which directly use LLM pseudo-labels during label propagation. **GLPN-LLM** (Hu et al., 2025) uses LLM pseudo-labels with a global propagation design. SSPC differs from this group by using LLM pseudo-labels as weak propagation-calibration signals rather than as training labels.

All F1 values are macro-averaged over the fake and real news classes, and each metric is computed per run before seed aggregation, so the reported mean F1 is the average of run-level F1 scores rather than being recomputed from the reported mean Precision and mean Recall. For the baselines in Table 1, GLPN-LLM was re-run using its official code and evaluation protocol, while the remaining baseline results — in particular, the Precision/Recall/F1 values of SpotFake and MVAE — are cited from the comparison table of the GLPN-LLM paper.

E Implementation Details

SSPC is implemented in PyTorch and PyTorch Geometric. The backbone follows the FCN-LP implementation: node features are loaded from pre-computed multimodal tweet embeddings, graph edges are loaded from the cross-modal tweet graph, and the graph model contains three GCN layers with hidden dimension 64. We use AdamW with learning rate 1×10^{-4} and weight decay 5×10^{-4} . Unless otherwise specified, the number of label-propagation iterations is 2, dropout is 0.84, and the maximum number of training epochs is 6000. Main comparison results are averaged over five random seeds; auxiliary sensitivity and mechanism analyses follow the configurations specified in their corresponding sections. We performed a grid search and chose the best hyper-parameters; test labels were never used for hyperparameter selection.

Encoder and calibration setup. SSPC is evaluated with both HMCAN and CLIP encoders; CLIP serves as the default for ablation studies. LLM pseudo-labels are generated by GPT-4o and used exclusively as weak propagation-calibration signals, not as training supervision. LGR learns edge weights from high-confidence pseudo-label agreement and disagreement, while CPUG estimates node-level propagation reliance from cross-fitted predictions, structural uncertainty, and semantic-structural conflict.

CPUG settings. CPUG uses 5-fold cross-fitting. For each held-out fold, the utility label is positive only when the propagation prediction improves over the base prediction by margin $m = 0.1$. The propagation gate is a two-layer MLP with dimensions $4 \rightarrow 16 \rightarrow 1$, ReLU activation, and sigmoid output. It is trained for 2000 epochs with the conflict-coupled gate objective and then frozen. The default gate regularization weight is $\gamma = 0.1$.

LGR settings. LGR uses $\tau = 0.7$ as the default LLM confidence threshold, which is examined in the threshold sensitivity analysis. For each edge, it builds semantic edge indicators from high-confidence pseudo-label agreement or disagreement and learns a differentiable edge-refinement coefficient. Edges without high-confidence LLM pseudo-label pairs are left unchanged.

Propagation details. For LGR, the same-pseudo-label coefficient is unbounded before normalization, but the incoming-normalized weights remain

Dataset	Test Nodes	LLM Pseudo-labels	All Acc.	High-conf. Acc.
Twitter	1,091	1,091	75.39	82.40
PHEME	606	606	74.38	79.40
Weibo	1,124	1,124	80.86	85.00

Table 4: LLM pseudo-label availability and accuracy. High-confidence denotes the highest-confidence Top- k samples.

a distribution over each node’s incoming edges, and the self-loop preserves node-local evidence. We empirically inspect learned refinement ratios in Section Q. For MGVP, we use a fixed small number of propagation steps ($T = 2$ by default) for local smoothing rather than convergence to a steady state. The propagation weights α_{ji} are normalized over the incoming neighbors of target node v_i ; for undirected graphs, edges are treated symmetrically, and isolated nodes rely on the self-loop. For CPUG, the main experiments use the default (m, γ) setting specified above, while the sensitivity study shows robustness around this default rather than claiming a globally optimal configuration for every dataset.

Semantic-structural conflict coupling settings. The semantic-structural conflict score is computed only for nodes with available high-confidence LLM pseudo-labels. It is used as the fourth CPUG feature and as the conflict-aware regularization term $\gamma s_i g_i$ in $\mathcal{L}_{gate}$. Therefore, semantic-structural conflict is learned as part of the propagation utility gate rather than applied as a separate fixed calibration step after prediction.

F LLM Pseudo-label Quality

Table 4 reports the quality of the LLM pseudo-labels used in this study. The pseudo-labels are noisy across datasets, and the highest-confidence Top- k subset is more reliable than the full pseudo-label set in these diagnostics. However, LLM pseudo-label quality remains uneven and imperfect across datasets. This observation motivates the conservative design of SSPC: pseudo-labels are not treated as task labels, and their role is limited to learned, local propagation calibration rather than direct supervision.

The dataset-level variation also explains why SSPC applies LLM pseudo-labels conservatively. LGR modifies existing graph edges and only when both endpoints have high-confidence pseudo-labels. CPUG uses disagreement between the pseudo-label and the propagation prediction as a semantic-structural conflict feature and a gate regularization

signal, rather than using pseudo-label agreement as positive class supervision. This design is intended to limit the influence of pseudo-label errors on the detector’s post-level content evidence.

G Additional Ablation Protocols

The main paper reports the removal of LGR, CPUG, and semantic-structural conflict coupling. We further clarify how these variants are constructed.

w/o LGR. This variant disables Veracity-aware Learnable Graph Refinement and keeps original graph edge weights. CPUG is unchanged. It tests whether learning LGR-refined graph is necessary.

w/o CPUG. This variant removes the learned propagation utility gate and therefore cannot estimate whether propagation benefits each node. It tests whether node-level propagation utility is necessary beyond graph refinement.

w/o SCGC. This variant removes the semantic-structural conflict feature s_i from CPUG and disables the $\gamma s_i g_i$ regularization term. It tests whether semantic-structural conflict should affect node-level propagation strength.

Confidence-threshold sensitivity. For threshold sensitivity, we sweep τ over $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ because the confidence scores are dataset-dependent. A lower τ increases coverage but may introduce noisy graph edge updates, while a higher τ makes both LGR and semantic-structural conflict coupling more conservative. This sweep supports using $\tau = 0.7$ as a default in the main experiments, rather than treating it as a calibrated confidence boundary.

Gate-regularization sensitivity. For gate regularization, we sweep γ over $\{0.0, 0.05, 0.1, 0.3, 0.5\}$. Larger γ more strongly discourages propagation under semantic-structural conflict. The default $\gamma = 0.1$ balances conflict avoidance and propagation utility.

H Additional Case Study

Figure 7 provides two additional PHEME cases where the label-propagation baselines are pulled toward the wrong veracity label, while SSPC predicts the ground truth. In the OTTAWASHOOTING case, the ground-truth label is fake, but FCN-LP and GLPN-LLM assign higher real-news probabilities ($P_{\text{real}} = 0.62$ and $P_{\text{real}} = 0.54$, respectively).

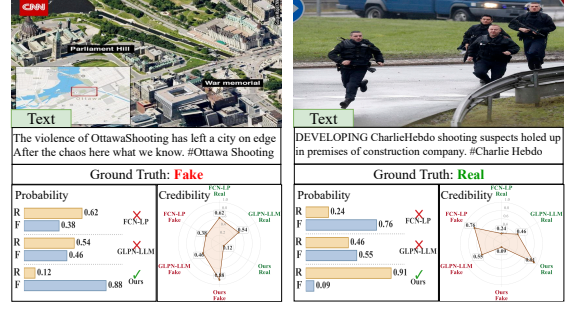


Figure 7: Additional PHEME cases from OttawaShooting and CharlieHebdo. FCN-LP and GLPN-LLM make the wrong veracity prediction in both cases, while SSPC predicts the ground-truth label by calibrating edge reliability via LGR and propagation reliance via CPUG.

SSPC instead predicts fake with $P_{\text{fake}} = 0.88$. In the CHARLIEHEBDO case, the ground-truth label is real, but FCN-LP and GLPN-LLM incorrectly favor fake ($P_{\text{fake}} = 0.76$ and $P_{\text{fake}} = 0.55$, respectively), whereas SSPC predicts real with $P_{\text{real}} = 0.91$.

These examples are qualitative mechanism diagnostics rather than evidence of universal robustness. They illustrate the intended behavior of SSPC in neighborhoods where semantically related posts can carry conflicting veracity signals: LGR reduces misleading propagation across unreliable edges, and CPUG lowers propagation reliance when local semantic-structural evidence is conflicted. This keeps the case analysis aligned with the main claim that SSPC uses noisy pseudo-labels as conservative calibration signals, not as direct supervision.

I Semantic-Structural Conflict Analysis

We test whether semantic-structural conflict is empirically present by measuring node-level label homophily. For each non-isolated node v_i , we compute $h_i = |\{j \in \mathcal{N}(i) : y_j = y_i\}| / |\mathcal{N}(i)|$ under three edge sets: all graph edges $\mathcal{E}$, *same-image* edges with CLIP image-embedding cosine similarity above 0.8, and *same-event* edges with text-embedding cosine similarity above 0.7. We compare these distributions with a per-dataset random baseline obtained by shuffling labels 100 times.

Figure 8 shows that homophily on Twitter, PHEME, and Weibo is consistently above random but far from perfectly veracity-consistent. Even image-similar and event-similar neighborhoods contain a long tail of mixed-veracity nodes, indicating that graph edges encode post relatedness rather than reliable label agreement. This analysis sup-

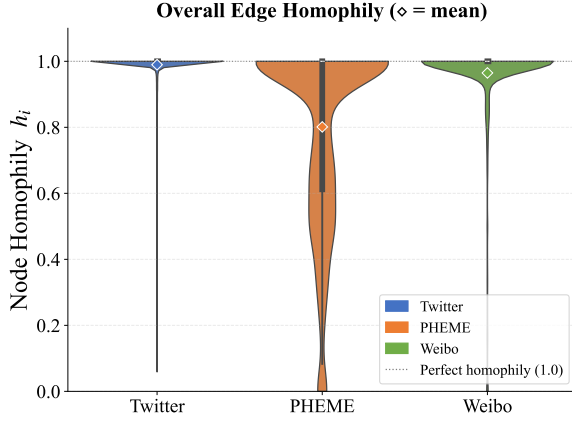


Figure 8: Node-level homophily h_i under three edge types, with per-dataset random-label baselines. Homophily is above random but remains far from perfect veracity consistency across datasets and edge types.

ports SSPC’s propagation calibration: LGR refines noisy edges using high-confidence pseudo-label agreement, while CPUG reduces graph reliance for nodes with unreliable neighborhoods. It is also consistent with Table 2, where removing LGR produces the largest mean F1 degradation across the three datasets; the homophily and UMAP analyses provide complementary mechanism-level evidence for why edge calibration improves propagation.

J Additional Hyperparameter Sensitivity Analysis

We complement the main hyperparameter analysis with the cross-fitting fold number K , the label-propagation iteration number T , and the CPUG hidden dimension d_g , which control gate-supervision granularity, propagation depth, and gate capacity, respectively. These sweeps use the CLIP encoder, report accuracy on all three datasets, and keep other hyperparameters at their defaults.

Joint Sensitivity of K and T . We jointly vary K and T around the default $K=5, T=2$: small K leaves fewer labeled nodes per fold model while large K raises cost and shrinks each held-out fold, and too few T iterations underuse graph context while too many over-smooth or carry errors through mixed-veracity neighborhoods.

Figure 9 shows that accuracy does not improve monotonically with more folds or deeper propagation: PHEME and Weibo peak at the default $K=5, T=2$, while Twitter peaks at $K=7, T=3$ (Best Acc. = 0.8781). Competitive regions concentrate around moderate settings, and the best (K, T)

pair is dataset-dependent, supporting a single cross-dataset default with dataset-specific tuning as an option.

Sensitivity to CPUG Hidden Dimension d_g . We vary the hidden dimension d_g of the CPUG MLP with fixed input features $\mathbf{r}_i = [H_i, J_i, D_i, s_i]$. Since utility labels are cross-fitted and noisy, a very small d_g may underfit interactions among the four features, while an overly large one may overfit fold-specific labels.

Figure 10 shows the default $d_g=16$ is compact and competitive: larger hidden dimensions are not consistently beneficial, and dataset trends are uneven (PHEME and Weibo remain competitive around moderate sizes while Twitter is more sensitive to over-capacity). This supports a small default gate with capacity as a dataset-dependent choice.

K Statistical Significance Tests (T-test)

To test whether the gains of SSPC (CLIP) over the strongest CLIP-based LLM-assisted label-propagation baseline are stable across random seeds, we conduct paired two-tailed t -tests (Koehn, 2004) on Accuracy, Precision, Recall, and F1 across all three benchmarks. The CLIP baseline is selected by F1 in Table 1: GLPN-LLM (CLIP) on Twitter and Weibo, and FCN-LP (CLIP) + LLM on PHEME. Each pair compares the same five random seeds for SSPC and the selected baseline. The null hypothesis is that SSPC and the corresponding baseline have equal expected performance; we reject it at significance level $\alpha = 0.05$.

As shown in Table 5, SSPC (CLIP) achieves consistently higher Accuracy, Precision, Recall, and F1 than the strongest CLIP-based LLM-assisted baseline across all three benchmarks, with p -values below 0.05 in all cases. Under the paired two-tailed t -test, these results indicate statistically significant improvements and make seed variation alone an unlikely explanation for the observed gains. Since the main comparison is primarily framed around F1, the F1 test remains the core significance evidence, while Accuracy, Precision, and Recall provide supporting metric-level evidence.

L Pseudo-label Robustness Analysis

Because LLM pseudo-label quality is uneven across datasets (Section F), we test SSPC’s robustness from two complementary views: perturbing the original pseudo-labels and measuring whether pseudo-label agreement is more reliable at the edge

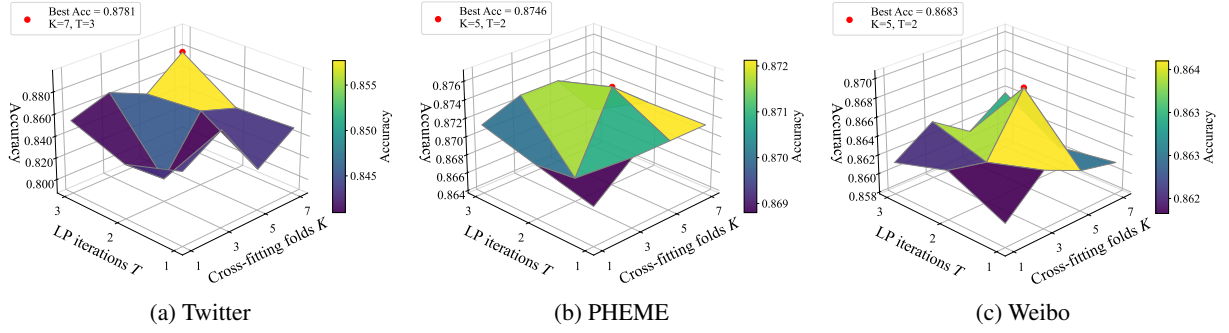


Figure 9: Joint sensitivity of SSPC to cross-fitting fold number K and label-propagation iteration number T . Accuracy is visualized over the (K, T) grid for Twitter, PHEME, and Weibo, with the default configuration $K=5$, $T=2$ indicated.

Dataset	CLIP Baseline	Acc.			Prec.			Rec.			F1		
		Baseline	SSPC	$\Delta \text{Acc.} / p_{\text{acc}}$	Baseline	SSPC	$\Delta \text{Prec.} / p_{\text{prec}}$	Baseline	SSPC	$\Delta \text{Rec.} / p_{\text{rec}}$	Baseline	SSPC	$\Delta \text{F1} / p_{\text{f1}}$
Twitter	GLPN-LLM (CLIP)	85.65 $\pm$ 2.11	86.43 $\pm$ 2.54	+0.78 / 3.04×10^{-2}	82.04 $\pm$ 2.68	83.53 $\pm$ 2.98	+1.49 / 4.11×10^{-2}	90.78 $\pm$ 2.47	98.23 $\pm$ 0.94	+7.45 / 1.84×10^{-2}	86.13 $\pm$ 2.36	88.50 $\pm$ 1.56	+2.37 / 3.82×10^{-2}
PHEME	FCN-LP (CLIP) + LLM	84.89 $\pm$ 2.81	87.46 $\pm$ 0.39	+2.57 / 3.96×10^{-2}	87.80 $\pm$ 2.36	91.40 $\pm$ 0.31	+3.60 / 2.88×10^{-2}	90.55 $\pm$ 2.28	90.97 $\pm$ 0.70	+0.42 / 4.73×10^{-2}	89.21 $\pm$ 2.59	91.18 $\pm$ 0.31	+1.97 / 2.60×10^{-2}
Weibo	GLPN-LLM (CLIP)	85.29 $\pm$ 1.57	86.83 $\pm$ 0.57	+1.54 / 3.58×10^{-2}	88.82 $\pm$ 0.73	89.32 $\pm$ 1.37	+0.50 / 4.62×10^{-2}	91.88 $\pm$ 0.79	94.08 $\pm$ 1.31	+2.20 / 2.91×10^{-2}	90.55 $\pm$ 0.58	91.64 $\pm$ 0.30	+1.09 / 2.73×10^{-2}

Table 5: Statistical significance test (paired two-tailed t -test) comparing SSPC (CLIP) against the strongest CLIP-based LLM-assisted LP baseline over five random seeds. Absolute scores are mean $\pm$ standard deviation from Table 1, and Δ values are mean improvements in percentage points. $p < 0.05$ indicates a statistically significant improvement.

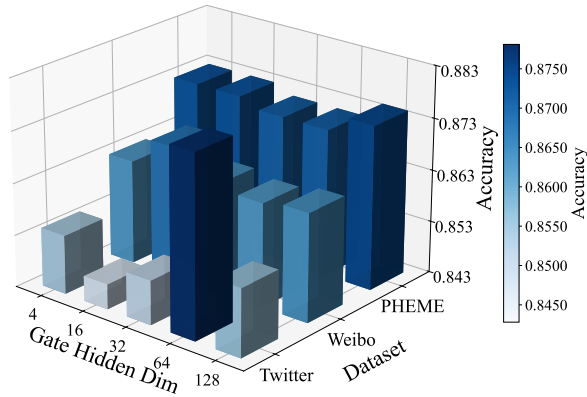


Figure 10: Sensitivity of SSPC to the CPUG hidden dimension d_g . The default gate uses a two-layer MLP with dimensions $4 \rightarrow 16 \rightarrow 1$.

level than at the node level. All variants use the CLIP encoder and the same partitions as the main evaluation; these are diagnostics rather than multi-seed significance claims.

Perturbing pseudo-label structure. We randomly flip a fraction $\rho \in \{0.0, 0.25, 0.50, 1.0\}$ of pseudo-labels while keeping confidence scores unchanged ($\rho=0.0$ is the original setting, $\rho=0.50$ randomizes agreement and disagreement in expectation, and $\rho=1.0$ flips every pseudo-label), isolating the value of pseudo-label agreement from that of simply attaching confidence-weighted binary attributes.

Table 6 shows dataset-dependent sensitivity to

corruption. Relative to $\rho=0.00$, Twitter F1 drops by 1.03–1.11pp across corruption levels, PHEME falls most at $\rho=1.00$ (−2.80pp), and Weibo is stable under moderate corruption (−0.34pp at $\rho=0.25$, −0.42pp at $\rho=0.50$) but deteriorates sharply at $\rho=1.00$ (−12.60pp F1). SSPC thus tolerates moderate pseudo-label noise, while severe corruption exposes dataset-dependent sensitivity.

Edge-level agreement quality. LGR uses pseudo-labels through pairwise agreement and disagreement on high-confidence edges rather than as direct node labels, so we measure whether pseudo-label relations match ground-truth veracity relations (same label when veracity labels agree, different otherwise), against an independent-error baseline $\text{acc}^2 + (1 - \text{acc})^2$.

Table 7 shows that overall edge-level relation accuracy exceeds both node-level pseudo-label accuracy and the independent-error baseline on all three datasets; gains over the baseline are +14.75pp, +30.13pp, and +25.50pp on Twitter, PHEME, and Weibo, and gains over node-level accuracy are +3.34pp, +18.03pp, and +15.00pp. These high scores come mainly from agreement edges, with weak or undefined disagreement accuracy. This supports the narrower conclusion that LGR exploits pairwise relation structure, not that node-level pseudo-labels are fully reliable or that agreement and disagreement cues are equally reliable.

ρ	Twitter				PHEME				Weibo			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
0.00	86.43	83.53	98.23	88.50	87.46	91.40	90.97	91.18	86.83	89.32	94.08	91.64
0.25	84.88	83.00	92.46	87.47	83.99	86.02	92.59	89.19	86.39	89.52	93.16	91.30
0.50	85.06	83.92	91.33	87.47	83.99	85.26	93.75	89.31	86.30	89.69	92.81	91.22
1.00	84.76	82.87	91.20	87.39	82.34	83.23	94.21	88.38	67.26	77.63	80.51	79.04

Table 6: Effect of pseudo-label corruption rate ρ . Higher ρ indicates stronger corruption of the original LLM pseudo-labels while confidence scores are fixed. Values are percentages (%).

Dataset	HC Edges	Agree Acc.	Disagree Acc.	Overall	PL Node Acc.	Indep.
Twitter	9,646	99.97	0.00	85.74	82.40	70.99
PHEME	389	99.47	20.00	97.43	79.40	67.30
Weibo	856	100.00	–	100.00	85.00	74.50

Table 7: Edge-level pseudo-label relation accuracy on high-confidence edges. Values are percentages. “Overall” measures whether pseudo-label agreement/disagreement matches ground-truth veracity agreement/disagreement. “PL Node Acc.” is the node-level accuracy of high-confidence pseudo-labels. “Indep.” is the independent-error baseline. “–” indicates that no high-confidence disagreement edges are available, so disagreement accuracy is undefined.

M Cross-fitting Ablation

Cross-fitting is a core design choice in CPUG because it computes propagation-utility labels from held-out predictions instead of predictions on nodes used for supervised training. To isolate this label-held-out protocol from the gate mechanism itself, we compare three gate training protocols:

- **SSPC (cross-fitted):** the default CPUG with $K=5$ -fold cross-fitting.
- **In-sample gate:** the gate is trained using utility labels derived from in-sample predictions (base and propagation predictors trained on *all* labeled nodes, then evaluated on those same nodes).
- **No gate / w/o CPUG:** the learned gate is removed and propagation is applied uniformly to all nodes, matching the w/o CPUG variant in Table 2.

All variants use the CLIP encoder and LGR-refined graph. Values are percentages averaged over five random seeds. The **Cross-fitted (SSPC)** and **No gate / w/o CPUG** rows are aligned with the corresponding rows in Table 2.

Table 8 shows that the default cross-fitted gate gives the strongest displayed mean on every metric in this ablation. It improves F1 over in-sample gate training by 2.74pp on Twitter, 2.35pp on PHEME, and 0.38pp on Weibo. Compared with removing CPUG, the cross-fitted gate improves F1 by

1.96pp, 1.68pp, and 0.87pp, respectively, matching the component-level conclusion in Table 2. The largest gains appear on Twitter and PHEME, while Weibo shows a smaller but consistent advantage. We therefore interpret this ablation as evidence that cross-fitting provides label-held-out utility labels and useful gate supervision, while treating the smaller Weibo margin as a dataset-level limitation.

N Strict Inductive Evaluation

We further restrict test-time information beyond the standard transductive setting to diagnose whether SSPC’s performance is explained by transductive access to test nodes, test-time pseudo-labels, or test-graph calibration. In the *Fully Inductive Node* setting, test nodes are isolated unseen nodes with no message passing, label propagation, or graph calibration; in the *No Test PL / No Test Graph Calibration* setting, graph connectivity is preserved but test-node pseudo-labels and any LGR/test-graph calibration over test nodes are disabled.

Table 9 shows the two restricted protocols give very similar F1 within each dataset (80.26/80.29 on Twitter, 89.74/89.85 on PHEME, 90.62/90.66 on Weibo), so the results are not primarily explained by test-time pseudo-label leakage or test-graph calibration. Compared with the standard setting, PHEME and Weibo remain close to their standard F1 scores, whereas Twitter drops substantially (from 88.50, Table 1), indicating Twitter depends more on graph context or transductive information. This supports interpreting the experiment as diagnostic evidence against test pseudo-label leakage being the sole explanation, not as evidence that strict inductive inference is uniformly lossless.

O Runtime and Memory Analysis

Table 10 compares detector-side training time, per-epoch time, inference latency, and peak GPU memory for SSPC, FCN-LP, and GLPN-LLM (five-seed averages, CLIP encoder, single RTX 4090

Gate Protocol	Twitter				PHEME				Weibo			
	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$
Cross-fitted (SSPC)	86.43 $\pm$ 2.54	83.53 $\pm$ 2.98	98.23 $\pm$ 0.94	88.50 $\pm$ 1.56	87.46 $\pm$ 0.39	91.40 $\pm$ 0.31	90.97 $\pm$ 0.70	91.18 $\pm$ 0.31	86.83 $\pm$ 0.57	89.32 $\pm$ 1.37	94.08 $\pm$ 1.31	91.64 $\pm$ 0.30
In-sample gate	81.21 $\pm$ 3.37	75.76 $\pm$ 3.05	97.88 $\pm$ 2.33	85.76 $\pm$ 2.34	82.94 $\pm$ 0.22	83.28 $\pm$ 0.62	89.19 $\pm$ 0.82	88.83 $\pm$ 0.11	86.23 $\pm$ 0.36	88.90 $\pm$ 0.68	93.76 $\pm$ 0.48	91.26 $\pm$ 0.19
No gate / w/o CPUG	82.58 $\pm$ 2.74	80.44 $\pm$ 3.46	98.07 $\pm$ 3.20	86.54 $\pm$ 1.85	84.98 $\pm$ 0.22	89.20 $\pm$ 1.00	89.81 $\pm$ 1.47	89.50 $\pm$ 0.19	85.41 $\pm$ 0.78	88.18 $\pm$ 1.78	93.50 $\pm$ 1.90	90.77 $\pm$ 0.38

Table 8: Cross-fitting ablation on gate training protocols. All variants use the CLIP encoder and LGR-refined graph. Values are percentages (%) averaged over five random seeds.

Protocol	Twitter				PHEME				Weibo			
	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	Rec $\uparrow$	F1 $\uparrow$
Fully Inductive Node	72.50	68.00	97.91	80.26	84.49	84.92	95.14	89.74	85.23	88.33	93.04	90.62
No Test PL / No Test Graph Calibration	72.41	67.81	98.39	80.29	84.49	84.21	96.30	89.85	85.23	87.99	93.50	90.66

Table 9: Strict inductive diagnostics under restricted test-time information access. Values are percentages.

D GPU). SSPC’s training time includes all stages ($K=5$ cross-fitting, gate training, final training), and inference latency is measured after features, graph connectivity, and required pseudo-labels are available; feature extraction, graph construction, and pseudo-label generation are excluded as pre-processing costs.

Runtime Comparison. SSPC’s training overhead is dataset-dependent rather than uniformly larger: compared with FCN-LP it is slightly lower on Twitter (7.31 vs. 8.03 min) and PHEME (1.43 vs. 1.63 min) but slightly higher on Weibo (1.49 vs. 1.41 min), and it is much faster than GLPN-LLM on Twitter (7.31 vs. 47.82 min) though slower on PHEME and Weibo. Its per-epoch time stays close to FCN-LP despite the cross-fitting stage.

At inference, SSPC’s latency is below 0.054 ms/sample on all benchmarks; FCN-LP is faster on all three and GLPN-LLM faster on PHEME and Weibo. The supported conclusion is therefore not that SSPC is the fastest detector, but that its absolute per-sample latency remains small under the shared transductive environment.

Memory Overhead. SSPC’s peak GPU memory lies between FCN-LP and GLPN-LLM on all benchmarks, reducing GLPN-LLM-level memory by about 45.2%/40.4%/63.1% on Twitter/PHEME/Weibo while still exceeding FCN-LP. Mechanistically, LGR and CPUG add only lightweight per-edge/per-node operations without large caches, and cross-fitting repeats per-fold training at a similar per-run footprint. As these values depend on implementation, graph scale, and hardware, they are intended for relative comparison under the same environment; they also motivate the complexity analysis below.

P Complexity Analysis

SSPC adds three main sources of computation over the base graph detector.

Cross-fitting cost. CPUG requires K cross-fitting rounds. With $K = 5$, the cost is higher than a single training run, but the propagation gate is trained only once and then frozen. This controls gold-label leakage in gate supervision: each utility label is computed from a predictor whose supervised loss excludes that node’s ground-truth.

Graph-refinement cost. LGR scans graph edges once and computes learnable edge-refinement coefficients from semantic edge features. Its time complexity is $O(|\mathcal{E}|)$ up to a small MLP factor, and its memory overhead is one refined edge-weight vector.

Semantic-conflict gate cost. CPUG computes an argmax label from $\mathbf{p}_i^{lp}$ and a scalar conflict score for each node, then evaluates a small gate MLP. Its time complexity is $O(|\mathcal{V}|)$ and its memory overhead is one scalar conflict score per node. Therefore, after the cross-fitted propagation gate is learned, the additional inference overhead of SSPC is small relative to graph propagation.

Q Analysis of Learned LGR Edge-Weight Distribution

To diagnose how LGR responds to pseudo-label agreement, we compare the distribution of the post-normalization refinement ratio

$$\rho_{ij}^{\text{ratio}} = \frac{a'_{ij}}{a_{ij}}, \quad (23)$$

(larger means the edge retains more weight after refinement) between high-confidence anchor-anchor edges whose endpoints share the same pseudo-label

Method	Twitter				PHEME				Weibo			
	Train (min)	Epoch (s)	Infer (ms)	Peak Mem (GB)	Train (min)	Epoch (s)	Infer (ms)	Peak Mem (GB)	Train (min)	Epoch (s)	Infer (ms)	Peak Mem (GB)
FCN-LP (CLIP)	8.03	0.1605	0.0363	6.86	1.63	0.0325	0.0054	0.032	1.41	0.0282	0.0032	0.059
GLPN-LLM (CLIP)	47.82	0.9563	0.0727	20.20	0.59	0.0117	0.0017	0.063	0.84	0.0168	0.0013	0.196
SSPC (CLIP)	7.31	0.1462	0.0539	11.06	1.43	0.0286	0.0073	0.038	1.49	0.0298	0.0040	0.072

Table 10: Detector-side wall-clock runtime and memory comparison on three benchmarks. “Train” is end-to-end detector-side training wall-clock time in minutes, “Epoch” is average training time per epoch in seconds, “Infer” is detector-side inference latency per test sample in milliseconds after features, graph connectivity, and required pseudo-labels are available, and “Peak Mem” is peak GPU memory usage in GB during training. Results are five-random-seed averages using the CLIP encoder on a single NVIDIA GeForce RTX 4090 D GPU.

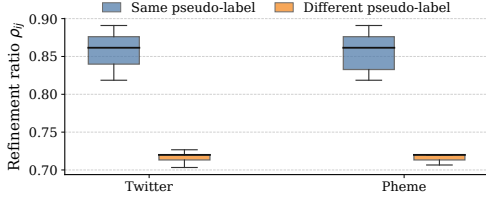


Figure 11: Distribution of learned LGR refinement ratios $\rho_{ij}^{\text{ratio}} = a'_{ij}/a_{ij}$ for same-pseudo-label and different-pseudo-label edges. Same-pseudo-label edges receive consistently higher refinement ratios than different-pseudo-label edges, indicating selective reweighting rather than uniform scaling.

Dataset	Same Edges	Diff. Edges	Same ratio	Diff. ratio	Δ ratio
Twitter	17,415	7,598	0.8569	0.7169	0.1399
PHEME	655	46	0.8537	0.7147	0.1390

Table 11: Summary statistics of learned LGR refinement ratios ρ_{ij}^{ratio} for same-pseudo-label and different-pseudo-label edges.

and those with different pseudo-labels. This is a diagnostic of refined-graph weights, not a claim about true semantic consistency.

Figure 11 and Table 11 show that same-pseudo-label edges receive consistently higher refinement ratios than different-pseudo-label edges on both Twitter (0.8569 vs. 0.7169) and PHEME (0.8537 vs. 0.7147). LGR therefore does not apply uniform scaling but selectively retains more mass on same-pseudo-label edges and downweights different-pseudo-label edges more strongly, evidence that it reacts to pseudo-label agreement (though not proof of true semantic consistency).

R Veracity-Awareness of LGR Learned Edge Weights

The edge-weight diagnostics in Section Q show that LGR responds to LLM pseudo-label relations, but they do not establish whether the learned reweighting aligns with true veracity. On PHEME, we therefore compare the pre-normalization edge multiplier

ρ_{ij} with the ground-truth relation $\mathbb{I}[y_i = y_j]$ over high-confidence anchor edges (LLM confidence $\geq \tau$), reporting the metrics defined in Table 12; here ρ_{ij} is distinct from the post-normalization ratio ρ_{ij}^{ratio} used in Section Q. PHEME is used because its high-confidence edge subset contains both same- and different-pseudo-label edges, which this diagnostic requires.

Setting	τ	n	Prec(str.)	Baseline	Lift	Recall(wk.)	Pearson r	Err-Amp
Fixed LGR ($\alpha=2, \beta=0.1$)	0.50	701	0.985	0.977	+0.008	0.50	0.191***	0.565
	0.80	297	0.993	0.987	+0.007	0.50	0.341***	0.356
Learnable LGR (seed=43)	0.80	297	0.987	0.987	0.000	1.00	0.230***	0.347
Learnable LGR (seed=45)	0.80	297	—	0.987	—	1.00	0.314***	—

*** $p < 0.001$. The displayed Learnable LGR seeds have positive Pearson correlations. Seed=45 has no strengthened-edge precision because no edge meets the strengthened-edge condition under this threshold.

Table 12: LGR edge weight veracity-awareness analysis on PHEME. *Prec(str.)*: true same-label ratio among strengthened edges; *Baseline*: unweighted anchor-edge same-label ratio; *Lift*: *Prec(str.)*–*Baseline*; *Recall(wk.)*: fraction of true different-label edges that are weakened; *Pearson r* : correlation between ρ_{ij} and true label agreement; *Err-Amp*: fraction of strengthened edges where both endpoints carry *wrong* pseudo-labels. “—” indicates that the corresponding strengthened-edge subset is empty or that the metric is not applicable.

Table 12 shows that, for Fixed LGR, raising the confidence threshold from $\tau=0.50$ to $\tau=0.80$ increases the correlation with true label agreement from $r=0.191$ to $r=0.341$ and reduces error amplification from 0.565 to 0.356, indicating that stronger confidence filtering better aligns pseudo-label-driven reweighting with true edge-level agreement. The displayed Learnable LGR seeds also show positive correlations, though seed=43 has zero lift over the high baseline same-label ratio and seed=45 has undefined strengthened-edge precision because no edge meets the strengthened-edge condition. Overall, LGR’s reweighting is partially aligned with true edge-level veracity agreement under high-confidence filtering, but the high baseline same-label ratio, the small edge subset, and incomplete learnable-seed reporting limit this evidence.

We treat this as a mechanism diagnostic rather than a cross-dataset conclusion, a causal proof, or a main significance test.

S Theoretical Analysis

Purpose and Reduced-order Setting. This section gives a mechanism-level explanation of SSPC rather than a generalization theorem. The aim is to isolate the two controls introduced by SSPC: LGR changes where propagation mass flows, and CPUG changes how much a node trusts that propagated mass. To keep the argument transparent, we study a reduced one-step propagation slice around a fixed node and treat the trained representation and softmax layers as locally fixed.

Assumptions and Notation. Assumption 1 (Reduced-order propagation slice). For each node i ,

$$\begin{aligned} q_i(\mathbf{W}) &= \sum_{j \in \mathcal{N}(i)} W_{ji} r_j, \\ r_j &= y_j + \epsilon_j, \\ W_{ji} &\geq 0, \quad \sum_{j \in \mathcal{N}(i)} W_{ji} = 1. \end{aligned} \quad (24)$$

where $y_j \in \{0, 1\}$ is the true label and ϵ_j is the local prediction error. Let

$$m_i = \mathbb{I}[\tilde{y}_i \neq -1, c_i \geq \tau], \quad (25)$$

so $m_i = 1$ marks a high-confidence pseudo-labeled node. Define the same-label and conflictive neighbor sets as $\mathcal{N}_i^+$ and $\mathcal{N}_i^-$:

$$\begin{aligned} \mathcal{N}_i^+ &= \{j \in \mathcal{N}(i) : y_j = y_i\}, \\ \mathcal{N}_i^- &= \{j \in \mathcal{N}(i) : y_j \neq y_i\}. \end{aligned} \quad (26)$$

The theory below suppresses higher-order representation learning, but it captures the effect of moving mass across same-label and different-label neighbors. At the gate level, $s_i \geq 0$ denotes the semantic-structural conflict score and $g_i \in [0, 1]$ denotes the propagation gate. The regularization weight is $\gamma \geq 0$.

Definition 1 (Conflict mass and pairwise relation correctness). The normalized conflict mass is

$$C_i(\mathbf{W}) = \sum_{j \in \mathcal{N}_i^-} W_{ji}. \quad (27)$$

For an edge (v_j, v_i) with $m_j m_i = 1$, we say the pairwise pseudo-label relation is correct when

$$\mathbb{I}[\tilde{y}_j = \tilde{y}_i] = \mathbb{I}[y_j = y_i]. \quad (28)$$

Equivalently, the indicators used by LGR,

$$\begin{aligned} u_{ji}^{same} &= m_j m_i \mathbb{I}[\tilde{y}_j = \tilde{y}_i], \\ u_{ji}^{diff} &= m_j m_i \mathbb{I}[\tilde{y}_j \neq \tilde{y}_i], \end{aligned} \quad (29)$$

match the true same/different relation on that edge. This is the only pseudo-label signal needed by LGR; the absolute correctness of each node-level pseudo-label is not required.

Propagation Error Decomposition. Lemma 1. Under Assumption 1,

$$|q_i(\mathbf{W}) - y_i| \leq \sum_{j \in \mathcal{N}(i)} W_{ji} |\epsilon_j| + C_i(\mathbf{W}). \quad (30)$$

Proof. Substituting $r_j = y_j + \epsilon_j$ gives

$$q_i(\mathbf{W}) - y_i = \sum_{j \in \mathcal{N}(i)} W_{ji} \epsilon_j + \sum_{j \in \mathcal{N}(i)} W_{ji} (y_j - y_i). \quad (31)$$

Applying the triangle inequality yields

$$\begin{aligned} |q_i(\mathbf{W}) - y_i| &\leq \sum_{j \in \mathcal{N}(i)} W_{ji} |\epsilon_j| \\ &\quad + \sum_{j \in \mathcal{N}(i)} W_{ji} |y_j - y_i|. \end{aligned} \quad (32)$$

Because the labels are binary, $|y_j - y_i| = \mathbb{I}[y_j \neq y_i]$, so the second term is exactly $C_i(\mathbf{W})$. This proves the claim.

Pairwise Pseudo-label Calibration. Lemma 2 (Pairwise correctness does not require node-level correctness). If two endpoints are flipped by the same binary flip, for example

$$(\tilde{y}_j, \tilde{y}_i) = (1 - y_j, 1 - y_i), \quad (33)$$

then $\mathbb{I}[\tilde{y}_j = \tilde{y}_i] = \mathbb{I}[y_j = y_i]$ even though $\tilde{y}_j \neq y_j$ and $\tilde{y}_i \neq y_i$. *Proof.* A common binary flip preserves equality and inequality: $y_j = y_i$ if and only if $1 - y_j = 1 - y_i$. Therefore the pairwise same/different relation is unchanged, which is exactly the information consumed by u_{ji}^{same} and u_{ji}^{diff} . More generally, any corruption that preserves pairwise equality on an edge keeps the LGR signal correct even when both node-level pseudo-labels are wrong.

Effect of Conservative LGR Rescaling. Proposition 1 (Conservative LGR reduces normalized conflict mass). Let a_{ji} be the original edge weights and let $a'_{ji} = a_{ji} \rho_{ji}$ be the LGR-refined weights. Assume that on every high-confidence

edge where LGR applies a nontrivial update, the pairwise pseudo-label relation is correct and the learned rescaling is conservative:

$$\begin{aligned} \rho_{ji} &\geq 1 && \text{for } j \in \mathcal{N}_i^+, \\ 0 \leq \rho_{ji} &\leq 1 && \text{for } j \in \mathcal{N}_i^-, \end{aligned} \quad (34)$$

while uncalibrated edges keep $\rho_{ji} = 1$. Let $\mathbf{W}$ and $\mathbf{W}'$ denote the normalized weights induced by $\mathbf{a}$ and $\mathbf{a}'$, respectively. Then $C_i(\mathbf{W}') \leq C_i(\mathbf{W})$. If at least one conflict edge is strictly downweighted and the refined incoming mass is positive, the inequality is strict. *Proof.* Let

$$A = \sum_{j \in \mathcal{N}_i^+} a_{ji}, \quad B = \sum_{j \in \mathcal{N}_i^-} a_{ji}, \quad (35)$$

and define A' and B' analogously with a'_{ji} . By the assumption, $A' \geq A$ and $B' \leq B$. Hence

$$C_i(\mathbf{W}') = \frac{B'}{A' + B'} \leq \frac{B}{A + B} = C_i(\mathbf{W}). \quad (36)$$

If some conflict edge is strictly downweighted and the total incoming mass is nonzero, then $B' < B$ and the inequality is strict, proving the proposition.

Gate Regularization under Semantic-Structural Conflict. Proposition 2 (Conflict-aware gate shrinkage). Let $z_i \sim \text{Bernoulli}(p_i)$ be the cross-fitted utility label, and consider the expected CPUG risk for a fixed node,

$$\begin{aligned} \ell(g; p_i, s_i) &= -p_i \log g - (1 - p_i) \log(1 - g) \\ &\quad + \gamma s_i g. \end{aligned} \quad (37)$$

For any fixed $p_i \in (0, 1)$ and $\gamma > 0$, ℓ has a unique minimizer $g_i^*(s_i)$ for $g \in (0, 1)$, and $g_i^*(s_i)$ is non-increasing in s_i . In particular, larger semantic-structural conflict yields a smaller optimal propagation gate. *Proof.* The first derivative is

$$\frac{\partial \ell}{\partial g} = -\frac{p_i}{g} + \frac{1 - p_i}{1 - g} + \gamma s_i, \quad (38)$$

and the second derivative is

$$\frac{\partial^2 \ell}{\partial g^2} = \frac{p_i}{g^2} + \frac{1 - p_i}{(1 - g)^2} > 0. \quad (39)$$

Therefore ℓ is strictly convex and has a unique minimizer in $(0, 1)$. At the optimum,

$$-\frac{p_i}{g_i^*} + \frac{1 - p_i}{1 - g_i^*} + \gamma s_i = 0. \quad (40)$$

Differentiating this optimality condition with respect to s_i gives

$$\frac{dg_i^*}{ds_i} = -\frac{\gamma}{\frac{p_i}{(g_i^*)^2} + \frac{1 - p_i}{(1 - g_i^*)^2}} < 0. \quad (41)$$

Hence the optimal gate decreases as the conflict score increases. The boundary cases $p_i \in \{0, 1\}$ are even simpler and lead to the same weak monotonicity conclusion.

Relation to SSPC and Scope of the Claim. Corollary 1 (Mechanism-level reduction in propagation exposure). When the pairwise pseudo-label relation is informative on the edges that LGR updates, and when s_i is positively associated with propagation risk, SSPC reduces the effective exposure of node i to conflictive propagation in two ways: Proposition 1 reduces $C_i(\mathbf{W})$, and Proposition 2 reduces g_i for high-conflict nodes. Consequently, the effective conflict term entering Lemma 1 is reduced from the uncalibrated baseline. This conclusion is local and mechanism-level; it is not a universal generalization theorem for arbitrary graphs or arbitrary LLM outputs. *Proof.* From Proposition 1, the refined graph satisfies $C_i(\mathbf{W}') \leq C_i(\mathbf{W})$. From Proposition 2, $g_i^*(s_i) \leq g_i^*(0)$ whenever $s_i \geq 0$. Since both factors are nonnegative,

$$g_i^*(s_i) C_i(\mathbf{W}') \leq g_i^*(0) C_i(\mathbf{W}). \quad (42)$$

Thus SSPC lowers the product of propagation reliance and conflict mass on nodes where the pairwise pseudo-label relation carries signal. The claim is intentionally local because the reduced-order model suppresses representation learning, nonlinear activation, and dataset-specific optimization effects. This proves the corollary.

The empirical diagnostics in Tables 7, 6, and 8 are consistent with this reduced-order story in a limited sense: edge-level agreement can provide signal beyond imperfect node-level pseudo-label accuracy, moderate corruption can be tolerated in some settings, full corruption shows dataset-dependent sensitivity especially on Weibo, and cross-fitting provides label-held-out gate supervision.

T LLM Prompt Template

Table 13 gives the prompt format used to obtain LLM pseudo-labels and confidence scores. The output is parsed into a binary LLM pseudo-label and a confidence score. SSPC does not train directly on these LLM pseudo-labels.

Field	Content
Role	You are a fact-checking assistant for multi-modal social media posts.
Input	Given the post text and its associated image or image description, judge whether the post is real or fake.
Guidelines	Consider whether the text-image pair is consistent, whether the claim is specific, and whether the evidence appears manipulated, out of context, or misleading.
Output	Return “Result: R , Confidence: $C\%$ ”, where $R = 1$ denotes real, $R = 0$ denotes fake, and C is a confidence score from 0 to 100.

Table 13: Prompt template for LLM pseudo-label.

U Ethics Statement

Our work aims to combat misinformation on social media. We use publicly available benchmark datasets that have been widely adopted in prior research. Because these benchmarks are derived from social media, some posts may contain user-generated text, images, usernames or other metadata, and may include offensive, misleading, or distressing content. We use these data only for research evaluation, do not attempt to identify individuals, and report aggregate results rather than reproducing sensitive examples. Any released artifacts will avoid redistributing raw private information beyond what is permitted by the original dataset terms. We acknowledge that automated fake news detection systems may produce false positives, and we recommend that such systems be used as assistive tools rather than sole arbiters of content veracity.

V Artifact License

Author-generated code, configuration files, and experiment scripts will be released with a license file in the public artifact/repository. Public datasets and third-party models or tools used by this work remain subject to their original licenses and terms of use. LLM pseudo-labels and derived experimental files are provided only for research reproducibility and evaluation, and do not change the licensing status of the original datasets.

W AI Assistants Disclosure

AI tools were used solely for language polishing, grammar correction, improvement of English expressions, and correction of code syntax errors.