CALIBRATED CO-EVOLVING SOCIAL SIMULATION FOR MULTI-STEP SENTIMENT FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sentiment forecasting predicts the evolution of population sentiment distributions as events unfold. Most existing methods treat this as single-step prediction. These methods prompt LLMs to role-play users from historical posts, assume fixed social graphs, and score the prediction at a single target time point. These three assumptions do not hold as the event unfolds. We formulate the task as a calibrated, closed-loop social simulation that couples post generation, network evolution and population-level feedback. The population generates its own future inputs. Each LLM agent represents one user. At each step, active agents respond to their neighbors' posts from the previous step, and the expressed stances drive edge formation and dissolution. History-based participation rates capture heterogeneous participation, while stance-decomposed camp summaries provide population context and preserve within-camp differences through agent-specific injection strengths. To calibrate the resulting dynamics, we combine individual-level and population-level objectives over observed multi-step trajectories, together with homophily-constrained opinion updates. Calibration uses only the observation windows and then freezes the parameters. The simulator runs with no access to future real posts or sentiment distributions. On a 2020 US election Twitter benchmark and a self-built event dataset, our method lowers the macro JSD in the free-forecast window by 85% and 75% over the single-step baseline and raises per-agent accuracy there from 43.0% to 55.5%. The simulated networks are consistent with the real graph on polarization, echo chamber, and opinion assortativity.

1 INTRODUCTION

When an event unfolds on social media, people do not form and express their opinions in isolation. They read what others say, respond to those expressions, and change whom they interact with as their views evolve. The sentiment observed in a population is therefore not simply a sequence of independent snapshots, but the outcome of an evolving social process. Forecasting this sentiment over multiple future steps requires modeling not only how sentiment changes, but also the interactions through which future sentiment is produced.

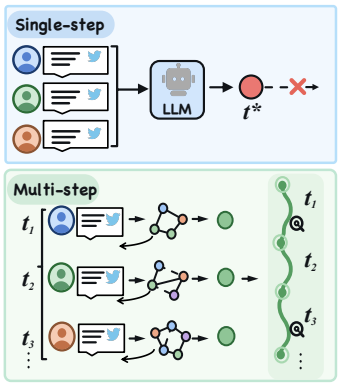

Figure 1: Single-step prediction versus multi-step simulation.

Most LLM-based approaches instead treat this as static single-step prediction, as illustrated in the top panel of Figure 1. Given a user's historical timeline, followee tweets, and demographic information, these methods prompt an LLM to role-play the user and generate a post at a target time point, then compare the aggregated sentiment with the ground truth Man et al. (2025). LLM-based annotation has also been shown to match or exceed fine-tuned classifiers on political text Törnberg (2024); Kuila & Sarkar (2024). This formulation, however, breaks the social process that generates future observations in three ways. Followee content is a snapshot of history rather than a product of the population itself, the social graph remains fixed even as expressed opinions change, and the forecast is made for a single time point rather than an evolving trajectory. As a result,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

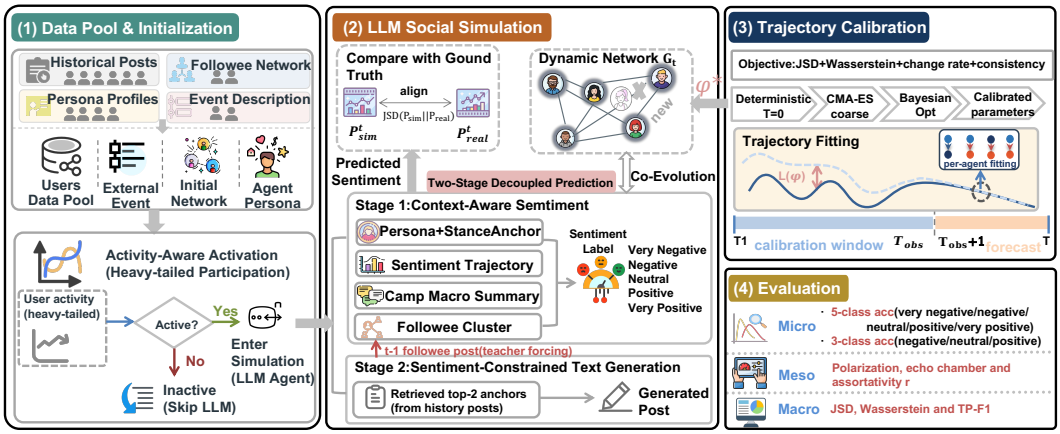

Figure 2: The simulator pipeline, from agent initialization to the evaluation of the forecast.

the inputs to later predictions are treated as given, although in the real system they are themselves outcomes of earlier interactions.

LLM-based social simulation provides a natural framework for restoring these interactions. Generative Agents Park et al. (2023) demonstrated long-horizon simulation with memory streams, OASIS Yang et al. (2024) scaled social simulation to one million agents with dynamic follow relations, AgentSociety Piao et al. (2025) embedded agents in a web environment, and HiSim Mou et al. (2024b) coupled LLM users with deductive bounded-confidence agents over a social movement. These systems reproduce social dynamics, but none of them is calibrated against observed sentiment trajectories Mou et al. (2024a); Wang et al. (2024), and the network mechanisms they include are not driven by the opinions agents express. This leaves a central gap. How can a social simulator be calibrated to the observed evolution of a population and then used to forecast its continuation without relying on future observations?

We address this gap with a calibrated, closed-loop co-evolving social simulation for multi-step sentiment forecasting, as illustrated in Figure 2. The simulated population generates its own future inputs, instead of replaying a frozen record. Each LLM agent represents one user. Its opinion is a continuous stance coordinate inferred from its own text at every step, and a bounded-confidence prior over its neighbors enters the objective as a consistency residual rather than governing the opinion directly. Edges form toward agents of similar opinion and toward well-connected accounts, and they break when two agents diverge beyond the trust radius. Since posting is heavy-tailed, each agent's participation rate drives an exponential activation model, and inactive agents are skipped while their last post stays visible to their followees. Prediction and generation are decoupled into two stages, because a single prompt makes them compete and introduces label bias. The first stage estimates the agent's sentiment from a camp summary of the macro state, visible only to that camp's members and injected at an agent-specific strength set by a relevance score. The second stage generates the post conditioned on that label, anchored to two of the agent's own tweets that fix its stance direction and style. We fit the parameters of this simulation to the whole trajectory, combining a population-level term on the simulated distribution with an individual-level term on each agent's own label. The objective is not differentiable, so we make the forward pass and graph updates deterministic and search with CMA-ES followed by Bayesian optimization. Our main contributions are summarized as follows:

- We propose a multi-step LLM simulation in which the feed, the network, and the macro signal form one calibrated loop, instead of independent per-step predictions.
- We develop a trajectory-level calibration that scores the whole trajectory rather than a single snapshot, at both the population level and the individual level. The simulation forecasts sentiment at any point of the trajectory, not only the next step.
- On a 2020 US Election Twitter benchmark and a self-built event dataset, the calibrated simulation lowers the macro JSD in the free-forecast window by 85% and 75% over the single-step baseline, raises per-agent accuracy in the same window, and transfers to a separate event with different actors and a different horizon.

2 Related Work

LLM-based social simulation. Generative Agents Park et al. (2023) added a memory stream and a reflection step that keep a town of agents coherent over days. OASIS Yang et al. (2024) scaled this to one million agents with changing follow relations. AgentSociety Piao et al. (2025) embedded a population in a city-scale environment with an economy. CAMEL Li et al. (2023) pairs a task-giver with a task-solver. HiSim Mou et al. (2024b) couples language-model core users with deductive bounded-confidence agents over a real social movement. Surveys of this line Chen et al. (2024); Mou et al. (2024a); Wang et al. (2024) report that these systems set their parameters by hand and assess macro behavior qualitatively. We instead fit those parameters to the observed sentiment trajectory and score the simulated distribution at every step.

Opinion dynamics and network co-evolution. Opinion dynamics models update each belief by a fixed rule, either averaging over neighbors DeGroot (1974); Friedkin & Johnsen (1990) or confining influence to a confidence bound Hegselmann & Krause (2002); Castellano et al. (2009). On real platforms, exposure is segregated by opinion Conover et al. (2011); Bakshy et al. (2015). Adaptive network models let ties form and break with the states their endpoints carry Hołyst et al. (2001); Gross & Blasius (2008). Preferential attachment Barabási & Albert (1999) and triadic closure Girvan & Newman (2002) explain the degree distributions and communities that follow. Latent-parameter methods recover an opinion model from social traces with online expectation–maximization Monti et al. (2020). These models keep the opinion a scalar updated by rule, and the network evolves independently of what agents express. We read each agent's opinion from the text it generates and let that reading drive edge formation and dissolution.

Sentiment forecasting from social media. Sentiment forecasting on social media Socher et al. (2013); Devlin et al. (2019); Liu et al. (2019) predicts a label per user per time point, and each step is predicted independently. SINN Okawa & Iwata (2022) recasts opinion dynamics as differential equations fitted to social media traces. Others recover the influence structure from observed activity De et al. (2015). MPR Man et al. (2025) role-plays several perspectives of a user for single-step sentiment prediction. LLM-MC-Affect Lin et al. (2026) estimates an affective trajectory from many stochastic decodings. We instead simulate the population behind the sentiment. The forecast and the social process come from the same run.

3 Method

Setting. A population of N agents $\mathcal{V} = \{1, \ldots, N\}$ evolves over discrete steps $t \in \{0, \ldots, T\}$ on a directed social graph $G_t = (\mathcal{V}, E_t)$, where $(i \to j) \in E_t$ means agent i follows agent j and $\mathcal{N}_i^t$ denotes the followees of i at step t. Each agent i carries a static persona persona_i and a stance score $\text{stance_score}_i \in [-1, 1]$ resolved from it, where $+1$ marks strong support for camp_1 and -1 for camp_2, the two opposing sides of the event under study. At each step agent i generates a post x_i^t through an LLM, and the classification stage of Section 3.2 assigns it a five-class sentiment label $s_i^t \in \{0, \ldots, 4\}$, ordered from very negative to very positive. The simulation is controlled by five interpretable parameters, the trust radius τ, macro-injection strength β, homophily decay γ, preferential-attachment coefficient η, and edge-dissolution sensitivity κ. Each is defined where it enters the mechanism below and fitted so that the simulated trajectory matches the real one (Section 3.5).

3.1 Activity-Aware Participation and Feed

Posting is heavy-tailed, so uniform activation overstates low-frequency users and distorts the feeds of their followees. We estimate each agent's participation rate f_i in posts per day from pre-event tweets:

$$f_i = \frac{n_i^{\text{post}}}{\max(1,\ \text{span}_i)}, \tag{1}$$

where n_i^{post} is the number of tweets in i's history and span_i its length in days, falling back to $n_i^{\text{post}}/30$ when the span is unknown. Agents with no history take a floor rate $f_i = 0.01$. Rates stay per agent, since the simulation must know *which* agents speak at each step. At each step agent i is active with probability $P(\text{active}_i) = 1 - e^{-f_i\,\Delta t}$, where Δt is the real-world span one step covers, expressed

in days so that f_i and Δt share a single unit and the activation law is stated independently of the cadence any particular event is sampled at. The probability is clamped to $[0.05, 0.95]$, and a step-level floor keeps at least 30% of agents active so a step is never dominated by carried-forward labels. Inactive agents skip the LLM entirely. Their last post remains their visible contribution to neighbors' feeds, and their most recent label is carried forward in macro aggregation.

Feed construction. At step t agent i observes the previous-step posts of its followees, $\text{feed}_i^t = \{x_j^{t-1}\}_{j \in \mathcal{N}_i^{t-1}}$, truncated to the 10 most recent entries, with the followees' historical seed tweets standing in at $t = 0$. A purely generated feed compounds error, because an unrepresentative post enters every follower's context and is re-generated downstream.

Teacher-forcing anchor. We therefore expose a run-level anchoring parameter $\alpha \in [0, 1]$, the *teacher-forcing strength*, set at launch, under which the entry contributed by followee j is independently replaced, rather than blended, by one of j's real historical tweets:

$$\text{feed}_{i \leftarrow j}^t = (1 - m_j)\, x_j^{t-1} + m_j\, x_j^{\text{real}}, \tag{2}$$

where $m_j \sim \text{Bernoulli}(\alpha)$ is the replacement indicator and x_j^{real} is a uniformly drawn pre-event tweet of j, so $\alpha = 0$ is fully free-running and larger values anchor the feed more strongly to observed content.

3.2 Stance-Guided Generation Pipeline

(A) Stance-Decomposed Macro Injection. A single global summary would contradict the losing side's supporters at turning points, so we decompose the macro state by camp. Each agent's stance toward the event $\text{stance}_i \in \{\text{pro-camp}_1, \text{pro-camp}_2, \text{neutral}\}$ is inferred from its persona, with a confidence $\text{conf}_i \in [0, 1]$ that falls when the text attests both camps. Writing $\mathcal{C}_j^{t-1} = \{i \mid \text{stance}_i = \text{pro-camp}_j\}$ for the members of camp j in G_{t-1}, the camp-specific distribution averages the class indicators of that camp alone, for $j \in \{1, 2\}$:

$$P_{\text{camp}_j}^{t-1} = \frac{1}{|\mathcal{C}_j^{t-1}|} \sum_{i \in \mathcal{C}_j^{t-1}} \mathbf{e}_{s_i^{t-1}}, \tag{3}$$

where $\mathbf{e}_k \in \{0, 1\}^5$ is the standard basis vector of sentiment class $k \in \{0, \ldots, 4\}$. The pair is verbalized into camp summaries $m_{\text{camp}_1}^{t-1}, m_{\text{camp}_2}^{t-1}$, and agent i receives only $m_{\text{stance}_i}^{t-1}$, tagged by its relevance score (Section 3.3). Agents whose stance is neutral or below the confidence threshold are excluded from both camp distributions. They instead receive the global summary, which is aggregated over the whole population.

(B) Persona Stance Anchoring and Event Neutralization. Each persona includes an explicit stance anchor built from its stance score and restated as a stable identity attribute. When the stance is confidently resolved, the anchor names the camp directly and appends the agent's own self-description, and otherwise the agent keeps that nuanced self-description, a counterweight to macro-level pressure. Stance counts as confidently resolved only when conf_i clears 0.45 and $|\text{stance_score}_i| > 0.3$. Exogenous event descriptions e^t are rewritten from evaluative framing to neutral dated reports that record each side's response, so the description favors neither side and stances can react divergently to the same fact.

(C) Two-Stage Decoupled Prediction. Stage 1 performs classification on a compressed context that retrieves no historical tweet, comprising i's persona persona_i and stance score stance_score_i, the sentiment histogram of its historical tweets baseline_i, its recent sentiment trajectory traj_i^t, the sentiment clusters its followees currently occupy $\text{followee_cluster}_i^t$, the neutralized event report e^t, and the camp summary it is entitled to, $m_{\text{stance}_i}^{t-1}$. The historical signals therefore enter only through baseline_i and traj_i^t, and Stage 1 produces $\hat{s}_i^t$ with an intensity rating. Stage 2 then generates the post conditioned on that prediction as a hard constraint, drawing on two style-exemplar tweets $\text{anchor}_i^1, \text{anchor}_i^2$ retrieved from i's own history and ranked by closeness to its mean polarity, deviation from its modal sentiment, recency, and intensity, in that order of weight (Appendix I), $x_i^t = \text{LLM}_\theta(\text{anchor}_i^1, \text{anchor}_i^2, e^t, \hat{s}_i^t)$. The sentiment label alone is the hard constraint, and we withhold the intensity. The model's safety preference therefore cannot weaken an extreme label, and event overfitting cannot push a neutral agent toward the event's direction.

3.3 Per-Agent Relevance and Injection Strength

The camp summary is injected equally to every camp member, but a member can carry the camp label yet dissent from its dominant sentiment, so uniform injection erases within-camp heterogeneity. We therefore compute a per-agent relevance score against the camp's macro distribution from four features. *Stance consistency* $c_i = \mathrm{clip}_{[0,1]}\big(\frac{1}{2} + \frac{1}{2}\,\mathrm{stance_score}_i \cdot \mathrm{polarity}(\mathrm{camp})\big)$, where camp polarity is its net positive-minus-negative ratio in $[-1, 1]$. *Topic alignment* $a_i = 1 - \min\big(1, \frac{1}{4}|\bar{s}_i - \mathbb{E}_{\mathrm{camp}}[s]|\big)$, where $\bar{s}_i$ is the mean of i's last three sentiment labels and $\mathbb{E}_{\mathrm{camp}}[s] = \sum_{k=0}^{4} k\, P_{\mathrm{camp}}(k)$ the camp's expected sentiment. *Social influence* l_i mixes the degree $\hat{d}_i$ normalized by the maximum degree in G_t with the participation rate $\hat{f}_i$ from Equation 1 normalized by the maximum rate. *Recency* $q_i = \mathbb{1}[\,i \text{ has posted}\,]$. When the camp macro distribution is unavailable the corresponding features default to their neutral value 0.5.

The four features enter the weighted sum $S_i = w_c\, c_i + w_a\, a_i + w_l\, l_i + w_q\, q_i$ with $w_c > w_a > w_l > w_q$, so that stance consistency dominates and social influence and recency only break ties. The weights are hand-set, not fitted. The score is squashed to $[0.076, 0.924]$ by $r_i = \sigma\big(5\,(S_i - 0.5)\big)$, and then tags the camp summary in Stage 1, with $r_i > 0.65$ prefixing it with "`[STRONG SIGNAL -- this closely matches your views]`" and $r_i < 0.35$ with "`[WEAK SIGNAL -- this may not reflect your views]`", while intermediate agents receive the unmodified summary. The same score separately sets the strength at which agent i receives it, $\beta_i = \beta\,(0.3 + 1.5\, r_i)$, where β is the macro-injection strength fitted in Section 3.5. It is attenuated for $0 < \beta_i < 1$ and passed through at full strength for $\beta_i \geq 1$, and at $\beta^* = 0.610$ the full-strength regime is reached only at the top of the relevance range, $r_i \geq 0.893$.

3.4 Opinion-Topology Co-Evolution

The discrete label s_i^t is too coarse to drive smooth opinion dynamics, and an LLM's text-grounded reasoning is not a mechanistic update rule, so we keep the two separate. The *state* is a continuous two-dimensional opinion vector $\mathbf{o}_i^t \in [-1, 1]^2$ inferred at each step from the Stage 1 structured prediction of Section 3.2. It drives the topology co-evolution, and we score the mechanistic prior of Equation 4 against it. For the macro distribution we use the discrete labels s_i^t rather than the opinion vectors. Its first coordinate is the continuous stance scalar of the Stage 1 prediction, and its second is the fixed persona stance $\mathrm{stance_score}_i$ that forms the second component of the FJ anchor $\mathbf{o}_i^0$ in Equation 4. The *mechanistic prediction* $\hat{\mathbf{o}}_i^t$ instead supplies a bounded-confidence prior through an extended Friedkin-Johnsen update:

$$\hat{\mathbf{o}}_i^t = \lambda_i\, \mathbf{o}_i^0 + (1 - \lambda_i)\Big[\sum_{j \in \mathcal{N}_i^{t-1}} W_{ij}^{t-1}\, \mathbf{o}_j^{t-1} + \mu_i\, \psi(e^t)\Big], \tag{4}$$

with stubbornness $\lambda_i \in [0, 1]$ and event susceptibility μ_i both set per agent from its own posting history, the former from sentiment variance and the latter from posting frequency, together with normalized influence weights W_{ij}^{t-1} and the event-to-opinion map $\psi(e^t)$. This defines a reference trajectory. The mechanism-consistency residual $\varepsilon_i^t = \|\mathbf{o}_i^t - \hat{\mathbf{o}}_i^t\|_2$ between the realized and the reference state enters the objective as $\mathcal{L}_{\mathrm{consist}}$ of Section 3.5, so the LLM states are *regularized by* a mechanistic prior rather than required to follow Friedkin-Johnsen dynamics. The influence weights integrate bounded-confidence gating, homophily decay, and preferential influence:

$$W_{ij}^t = \frac{\mathbb{1}[d_{ij}^t < \tau] \cdot \exp(-\gamma\, d_{ij}^t) \cdot (1 + \eta\, \log \deg_j^{\mathrm{in},t})}{\sum_{k \in \mathcal{N}_i^t} \mathbb{1}[d_{ik}^t < \tau] \cdot \exp(-\gamma\, d_{ik}^t) \cdot (1 + \eta \log \deg_k^{\mathrm{in},t})}, \tag{5}$$

where $d_{ij}^t = \|\mathbf{o}_i^t - \mathbf{o}_j^t\|_2$ is the opinion distance, $\deg_j^{\mathrm{in},t}$ the in-degree of j in G_t, and $\mathbb{1}[d < \tau]$ implements bounded confidence, where τ gates which followees influence i at all, γ sets how fast that influence falls with distance, and η sets how strongly a followee's in-degree boosts its influence. The graph G_t evolves concurrently through probabilistic formation:

$$\begin{aligned} \pi_{ij}^t &= \rho_t\, \sigma\big(a_0 - \gamma\, d_{ij}^t + \eta \log \deg_j^{\mathrm{in},t-1} + a_3\, n_{ij}^{\mathrm{cn}}\big), \\ \Pr\big[(i \to j) \in E_t \mid (i \to j) \notin E_{t-1}\big] &= \min\{0.3,\ \pi_{ij}^t\}, \end{aligned} \tag{6}$$

where $a_0 = -2.5$ and $a_3 = 0.3$ are fixed constants, n_{ij}^{cn} is the number of common neighbors of i and j in G_{t-1}, counting both in- and out-neighbors, and ρ_t is a density-dependent scaling factor. The homophily slope and the preferential-attachment coefficient are *shared with the opinion dynamics* and are exactly γ and η from Equation 5. Existing edges are dissolved by the mirror rule, under which an edge beyond the trust radius can be dropped with a probability that rises in $d_{ij}^t - \tau$ with sensitivity κ, the fifth calibrated parameter. Formation decays with the same distance through γ and removal starts only past τ, so the greater the opinion distance, the less likely a tie is to form or to survive. This is the mechanism behind the echo-chamber statistic reported in Section 4.3. With fixed G, scalar opinions, $\tau = \infty$ and no events, Equation 4 reduces to DeGroot consensus, and adding $\lambda_i > 0$, dynamic $\mathbb{1}[d < \tau]$ weights, or topology evolution recovers Friedkin-Johnsen, Hegselmann-Krause, and the co-evolving voter model Clifford & Sudbury (1973) respectively.

3.5 Degeneracy-Aware Trajectory Calibration

Importance reweighting at observation points. Let P_{sim}^t and P_{real}^t be the simulated and the real macro-level sentiment distribution over the five classes. Free-running simulation accumulates drift over long horizons, so at each observed time point $t \in \mathcal{T}_{\text{obs}}$, once P_{real}^t is known, we reweight the agent sentiment samples by

$$w_i^t \propto \frac{P_{\text{real}}^t(s_i^t)}{P_{\text{sim}}^t(s_i^t)}, \tag{7}$$

so that $\sum_i w_i^t \mathbf{e}_{s_i^t}$ matches P_{real}^t. The reweighting constrains the simulation without overwriting agent identities.

Forecast protocol and information boundary. The *post-observation update* applies at $t \in \mathcal{T}_{\text{obs}}$. Equation 7 is applied and its output, not the free-running simulation, conditions the next step, so the trajectory is anchored to P_{real}^t and any error there is calibration fit rather than forecast skill. The scores we report for this regime are measured on the un-reweighted trajectory. The *free forecast* starts at the first window no observation reaches, where the parameters are frozen at ϕ^* and no ground-truth label enters, so the simulated distribution there is a genuine prediction. The model is told that an event occurred and its neutralized description, but not its sentiment consequence, so the forecast is *scenario-conditional* under a known event. For the election run, the T_5 numbers in Section 4.2 come from the free-forecast regime while its T_1–T_4 entries are calibration fit and not comparable to them, and the Ukraine War columns are held-out forecasts.

Explicit optimization objective. We define a trajectory-level alignment loss:

$$\begin{aligned}\mathcal{L}_{\text{macro}}(\phi) = &\underbrace{\frac{1}{|\mathcal{T}_{\text{obs}}|}\sum_{t\in\mathcal{T}_{\text{obs}}} \text{JSD}\big(P_{\text{sim}}^t(\phi)\,\big\|\,P_{\text{real}}^t\big)}_{\text{pointwise distribution alignment}} + \lambda_1 \underbrace{\frac{1}{|\mathcal{T}_{\text{obs}}|}\sum_{t\in\mathcal{T}_{\text{obs}}} W_1\big(P_{\text{sim}}^t(\phi),\ P_{\text{real}}^t\big)}_{\text{trajectory shape alignment}} \\ &+ \lambda_2 \underbrace{\frac{1}{|\mathcal{T}_{\text{obs}}|-1}\sum_{t=1}^{|\mathcal{T}_{\text{obs}}|-1} \big\|\Delta P_{\text{sim}}^t - \Delta P_{\text{real}}^t\big\|_1}_{\text{change-rate alignment}},\end{aligned} \tag{8}$$

where $\Delta P^t = P^t - P^{t-1}$ and λ_1, λ_2 weight the two trajectory terms. The Wasserstein term is evaluated per time point on the five-bin distributions and respects the ordering of the five sentiment classes, $W_1(P, Q) = \sum_{k=0}^{4} \big|\sum_{l\le k}(P_l - Q_l)\big|$. The change-rate term handles the case where the pointwise distributions match but the dynamics do not.

What macro alignment leaves open. The macro term alone does not fix which agent carries which sentiment, since for any target P_{real}^t, the set $\mathcal{S}(P_{\text{real}}^t)$ of individual assignments $\mathbf{s} \in \{0, \ldots, 4\}^N$ whose empirical distribution is exactly P_{real}^t has the size of a multinomial coefficient, enormous for $N = 3000$. Every member yields the same macro divergence yet describes a different population (permuting labels leaves P_{sim}^t exactly equal to P_{real}^t while every individual prediction is wrong), and the change-rate term only partially alleviates this, since it constrains aggregate dynamics rather than individual assignments. We anchor the individual in two ways. The micro term covers the agents that have per-agent labels, and the consistency residual of Section 3.4 covers the dynamics side. Together they select the assignment in $\mathcal{S}(P_{\text{real}}^t)$ that is also consistent with each agent's observed

behavior. We attribute the forecast skill to fixing which agent carries which sentiment rather than to fitting the macro curve on the observed window.

The micro-level fidelity term $\mathcal{L}_{\text{micro}}$ is a per-agent KL divergence. A regularizer maintains parameter interpretability, $\mathcal{L}_{\text{reg}}(\phi) = \rho_1 \mathcal{L}_{\text{consist}} + \rho_2 \|\phi - \phi_0\|_2^2$, where $\mathcal{L}_{\text{consist}}$ averages the mechanism-consistency residual ε_i^t of Section 3.4 over agents and observed time points, and ϕ_0 anchors parameters near their priors. The total objective is $\min_\phi \mathcal{L}(\phi) = \mathcal{L}_{\text{macro}}(\phi) + \lambda_{\text{micro}}\mathcal{L}_{\text{micro}}(\phi) + \mathcal{L}_{\text{reg}}(\phi)$, with fixed rather than calibrated weights $\lambda_1, \lambda_2, \rho_1, \rho_2$ and the prior ϕ_0. $\mathcal{L}$ is non-differentiable in ϕ because the forward simulation is discrete, so we adopt deterministic calibration at temperature zero, using greedy LLM decoding and thresholded graph operations. CMA-ES Hansen (2006) then performs a coarse search over the 5-dimensional parameter space, followed by Bayesian Optimization Snoek et al. (2012) with a Gaussian Process surrogate and Expected Improvement for fine-tuning. Calibration yields $\phi^* = \{\tau = 0.202,\ \beta = 0.610,\ \gamma = 1.071,\ \eta = 0.476,\ \kappa = 1.001\}$.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on two event datasets, and Appendix A lists the exact windows. The public benchmark is the US Election 2020 Tweets dataset published by Hui (2020), with 3,000 agents over five event windows T_1–T_5, one step per window, T_k being step $t = k - 1$, where T_1–T_4 form the calibration window $\mathcal{T}_{\text{obs}}$ and T_5 is the free forecast. The self-built event is the Ukraine War, also with 3,000 agents, over a run of four steps. Neither dataset records follow relations, so we build the interaction graph as a language-matched retweet proxy (Appendix A). That proxy starts the simulation at G_0 and is also the observed network of Table 2(b). For each event we report the window nearest the event as *closest* and a free-forecast window as *after*, T_3 and T_5 for the election and the invasion-anniversary eve and the following week for the Ukraine War, where both Ukraine windows are held out. We fit the parameters once, on the election windows, and transfer them to the second dataset without re-fitting, so its columns in Table 1 are a cross-event transfer rather than a second calibration.

Macro statistic. The simulated distribution counts *agents* and the real trajectory counts *posts*, so a prolific user counts once in one and repeatedly in the other and we treat macro agreement as descriptive rather than exact.

Annotation. Both the agent LLM and the ground-truth annotator are Qwen3-32B, served locally via vLLM. The ground truth is audited against a human re-annotation of a stratified sample and cross-checked against BERT and DeepSeek, and Appendix J reports the agreement of each and the directional bias that keeps BERT-scored rows on a separate scale.

Baselines. We compare against five baselines, B1 Hegselmann-Krause with scalar opinions on a fixed graph, B2 Friedkin-Johnsen with static topology, B3 single-step MPR Man et al. (2025), B4 our pipeline with the macro alignment removed, and B5 our pipeline with a static followee feed. The cross-event comparison adds the voter model Clifford & Sudbury (1973), the DeGroot update DeGroot (1974), and the nonlinear model SLANT+ Kulkarni et al. (2017). Baselines that cannot produce a multi-step trajectory are evaluated at each reported point under the same information boundary. **Metrics.** We evaluate at three levels. At the micro level we report per-agent sentiment accuracy at the *closest* and *after* points, and F1-macro on the election dataset. We report each metric on the five-class scale and on a three-class version that merges the two negative classes and the two positive classes. At the meso level we report polarization, the ratio of opinion standard deviation at T_5 to T_1, echo chamber, the fraction of same-stance neighbors, and opinion assortativity r on opinion polarity rather than degree. At the macro level we report per-time-point Jensen-Shannon divergence, trajectory-level first-order Wasserstein distance, and turning-point detection F1. Calibration and generation are seeded, so each table reports one deterministic run without error bars.

Our method has the lowest macro JSD in every column of Table 1, and on the Ukraine War it also leads every baseline on micro-level accuracy. No Ukraine window takes part in the fit, so the baselines stay near chance there while ours remains accurate. The election dataset separates the two regimes most clearly. The single-step baseline is competitive at the *closest* point but loses that advantage at the forecast point, and its macro JSD moves in the opposite direction to ours across

Table 1: Macro JSD and micro 5-class accuracy. Best per column in bold.

	Election				Ukraine War			
	Closest		After		Closest		After	
Method	JSD↓	5-Acc↑	JSD↓	5-Acc↑	JSD↓	5-Acc↑	JSD↓	5-Acc↑
Voter	0.113	0.436	0.142	0.384	0.337	0.177	0.323	0.187
DeGroot	0.090	0.420	0.112	0.342	0.351	0.188	0.204	0.192
SLANT+	0.149	0.332	0.141	0.345	0.424	0.199	0.411	0.213
B3 Single-step MPR	0.072	**0.583**	0.127	0.430	0.255	0.152	0.260	0.165
Ours	**0.034**	0.577	**0.019**	**0.555**	**0.059**	**0.423**	**0.065**	**0.449**

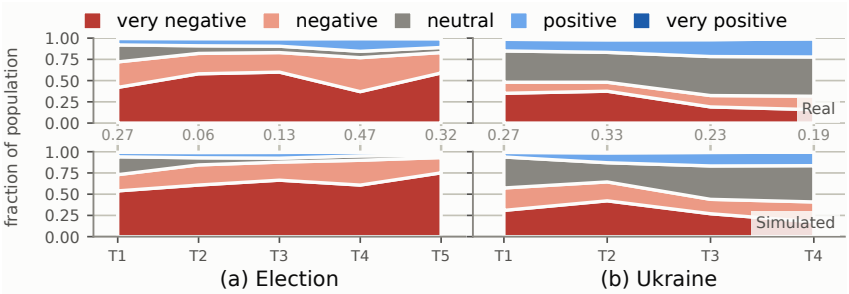

Figure 3: Real and simulated sentiment composition on two event datasets, the 2020 US election (a) over its five windows T_1–T_5 and the Ukraine War event (b) over the four steps of its run T_1–T_4. Within each block the top row is the ground truth, the bottom row our simulation, and the strip between them the per-step L_1 distance.

the two windows. Our advantage therefore holds in the free forecast, which is where a forecasting model should have it, and fitting the calibration window alone does not produce it.

4.2 Macro-Level Trajectory Alignment

The individual baselines appear in full in Table 6, together with the two trajectory-level metrics. No $\mathcal{T}_{\text{obs}}$ entry here, ours included, is measured on the reweighting of Equation 7, so the column compares like with like. Every method runs free at T_5. Our method leads both trajectory-level metrics over the five windows, while removing either ingredient, the macro alignment of B4 or the dynamic followee feed of B5, gives the loss back. No free-running variant reproduces the observed sentiment mix, and neither do the hand-specified dynamics of B1. The two compositions appear side by side in Figure 3. The simulated trajectory stays close to the real one at every sampled point, and their per-step L_1 distance does not widen as the horizon grows.

4.3 Micro-Level and Meso-Level Results

In the first panel we report micro-level sentiment prediction for every method. At T_3 the single-step baseline is already competitive and even edges ours out on 5-class accuracy, since the event is still unfolding and it has full access to the observed content. At the forecast point T_5, where Biden's victory declaration invalidates that assumption, the ordering reverses: our accuracy holds close to its T_3 level while every baseline falls away, we lead on both accuracy metrics and move ahead of the two strongest ablations, so the trajectory objective and the dynamic environment each contribute to the gain. There the single-step baseline keeps both F1 columns, by 0.003 on five classes and 0.027 on three, while its accuracy trails ours by 0.125 and 0.183. In the second panel we compare the simulated network against the real graph. Our method is closest on all three statistics, while B4 over-polarizes and B5 under-polarizes. The meso statistics above all follow from the interaction graph, so we also group the agents by US state, a partition the graph does not determine. The simulation reproduces the sign structure and the overall level to within 0.14, while its region-level ordering stays weaker (Appendix K).

Table 2: **(a)** Per-agent sentiment prediction at *Closest* and *After*, against the same agent-level ground truth, all metrics higher-is-better and best per column in bold. **(b)** Network structure at the forecast point, bold marking the entry closest to the observed value.

(a) Micro-level sentiment.

	Closest				After			
Method	5-Acc	5-F1	3-Acc	3-F1	5-Acc	5-F1	3-Acc	3-F1
B1 HK numerical	0.296	0.149	0.615	0.216	0.186	0.083	0.565	0.201
B2 FJ + static topology	0.207	0.130	0.555	0.211	0.170	0.106	0.517	0.198
B3 Single-step MPR	**0.583**	0.300	0.720	**0.495**	0.430	**0.260**	0.517	**0.417**
B4 No macro alignment	0.479	0.293	0.638	0.385	0.394	0.232	0.545	0.324
B5 Static followee	0.527	0.216	0.671	0.390	0.412	0.221	0.565	0.371
Ours	0.577	**0.331**	**0.745**	0.478	**0.555**	0.257	**0.700**	0.390

(b) Meso-level structure.

Method	Polarization	Echo chamber	Assortativity r
Real network (observed)	1.091	0.689	0.041
B2 FJ + static topology	0.806	0.987	0.164
B4 No macro alignment	1.372	0.835	0.459
B5 Static followee	1.024	0.507	0.223
Ours	**1.057**	**0.581**	**0.079**

Table 3: Ablation of the alignment components and of the annotation pipeline. The separated last row replaces the LLM annotator with BERT for the evaluation ground truth, so it is not on the same accuracy scale as the rows above.

	Closest		After	
Variant	5-Acc	3-Acc	5-Acc	3-Acc
Ours (full)	0.577	0.745	0.555	0.700
− Teacher forcing (α=0)	0.554	0.715	0.424	0.652
− Two-stage decoupled prediction	0.502	0.659	0.464	0.619
− Stance-decomposed macro injection	0.397	0.586	0.226	0.418
− Followee clustering	0.514	0.707	0.435	0.630
BERT annotation pipeline	0.659	0.716	0.634	0.691

4.4 Ablation Study

We ablate the main alignment components and the annotator used for the evaluation ground truth (Table 3). The stance-decomposed macro injection component contributes most on 5-class accuracy at the forecast point, ahead of the teacher-forcing feed anchor of Equation 2 and the Stage 1 followee-cluster context, where the two-stage decoupled prediction component contributes least. No component is redundant. Every component degrades the forecast point more than the calibration point, which suggests that these components keep the simulation on track once it runs free. The BERT-scored row is not a competing method, since replacing the annotator shifts accuracy in the direction of that annotator's documented bias, and the ordering against the single-step baseline is unchanged under either annotator.

5 Conclusion

In this paper, we present a calibrated multi-step social simulation in which the feed is generated rather than recorded, opinions and the network co-evolve, and the calibration scores the whole trajectory. Because the macro curve alone does not fix which agent holds which opinion, we also score how well each agent's opinion matches the opinions of the like-minded agents it follows. On the election dataset we cut the forecast JSD at the *after* point by 85% over the single-step baseline, and by 75% on the Ukraine War. The simulation also reproduces the real network's polarization and echo chambers. We evaluate on two English-language events, and the real statistics count posts while the simulation counts agents. In future work we plan to extend the calibration to other languages and longer horizons.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks, which require disclosure. The simulator is built on a generative model: Qwen3-32B, served locally through vLLM, is the agent model that writes every simulated post, and each agent's stance coordinate is read out of that model's own structured stance prediction for the step, falling back to its annotation of the post it generated. The same model produces our data: the simulation generates the synthetic posts and opinion trajectories that we evaluate, and Qwen3-32B also produces the sentiment labels that serve as ground truth for the election and Ukraine War datasets, with DeepSeek as a cross-check annotator and BERT as a non-generative comparison annotator. We also used generative AI tools to design and get feedback on our experimental methodology and evaluation protocol, to help interpret the results of our comparisons, and to help translate and polish our drafts into English.

We have not used generative AI tools for the remaining tasks that require disclosure. No theoretical model or conceptual framework in this paper was produced by a generative model, and the mathematical claims, the research hypotheses, and the analysis of the annotation sample are our own.

Additionally, we used generative AI tools for tasks with recommended disclosure: we drafted and edited parts of the manuscript, created and modified figures, wrote and edited code for the simulator and for the evaluation scripts, and searched the literature, formatted references, and chose the structure of the paper.

We have reviewed all AI-assisted work and take responsibility for the final content of this work, including the text, the claims, and the artifacts produced with the aid of generative AI. A human expert re-annotated a stratified sample of 100 tweets, and Appendix J reports the per-class recall of every annotator against that sample.

REPRODUCIBILITY STATEMENT

All reported runs are seeded, calibration and generation use greedy decoding at temperature zero, and the graph updates of Appendix G are applied as hard thresholds rather than samples, so each table is a single deterministic run and we report no error bars. Appendix C reports how the free forecast reacts when we move the trust radius and the homophily decay away from their calibrated values. Appendix H lists every constant we fix by hand, Section 3.5 gives the calibrated vector, Appendix F gives the loss terms and the two protocol regimes, and Appendices D and E define the metrics. Calibration searches the five parameters with CMA-ES for at most 20 generations of 8 candidates, followed by at most 20 Bayesian optimization evaluations, so a full calibration stays under 200 simulator rollouts. Appendix A describes how each dataset was built, Appendix B reproduces the prompts of both simulation stages, and Appendix J how the labels were validated.

REFERENCES

Eytan Bakshy, Solomon Messing, and Lada A. Adamic. Exposure to ideologically diverse news and opinion on Facebook. *Science*, 348(6239):1130–1132, 2015. doi: 10.1126/science.aaa1160.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999. doi: 10.1126/science.286.5439.509.

Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, 2009. doi: 10.1103/RevModPhys.81.591.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, Aili Chen, Nianqi Li, Lida Chen, Caiyu Hu, Siye Wu, Scott Ren, Ziquan Fu, and Yanghua Xiao. From persona to personalization: A survey on role-playing language agents. *Transactions on Machine Learning Research (TMLR)*, 2024. Accepted to TMLR 2024.

Peter Clifford and Aidan Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973. doi: 10.1093/biomet/60.3.581.

Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):89–96, 2011. doi: 10.1609/icwsm.v5i1.14126.

Abir De, Isabel Valera, Niloy Ganguly, Sourangshu Bhattacharya, and Manuel Gomez-Rodriguez. Learning and forecasting opinion dynamics in social networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69 (345):118–121, 1974. doi: 10.1080/01621459.1974.10480137.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, volume 1: Long and Short Papers, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423.

Noah E. Friedkin and Eugene C. Johnsen. Social influence and opinions. *The Journal of Mathematical Sociology*, 15(3–4):193–205, 1990. doi: 10.1080/0022250X.1990.9990069.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. doi: 10.1073/pnas.122653799.

Thilo Gross and Bernd Blasius. Adaptive coevolutionary networks: A review. *Journal of the Royal Society Interface*, 5(20):259–271, 2008. doi: 10.1098/rsif.2007.1229.

Nikolaus Hansen. The CMA evolution strategy: A comparing review. In José A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea (eds.), *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, volume 192 of *Studies in Fuzziness and Soft Computing*, pp. 75–102. Springer, Berlin, Heidelberg, 2006. doi: 10.1007/3-540-32494-1_4.

Rainer Hegselmann and Ulrich Krause. Opinion dynamics and bounded confidence: Models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3):2, 2002. URL https://jasss.soc.surrey.ac.uk/5/3/2.html.

Janusz A. Hołyst, Krzysztof Kacperski, and Frank Schweitzer. Social impact models of opinion dynamics. In Dietrich Stauffer (ed.), *Annual Reviews of Computational Physics IX*, pp. 253–273. World Scientific, Singapore, 2001. ISBN 978-981-02-4537-5. doi: 10.1142/9789812811578_0005.

Manch Hui. US election 2020 tweets. Kaggle dataset, 2020. URL https://www.kaggle.com/datasets/manchunhui/us-election-2020-tweets.

Alapan Kuila and Sudeshna Sarkar. Deciphering political entity sentiment in news with large language models: Zero-shot and few-shot strategies. In *Proceedings of the Second Workshop on Natural Language Processing for Political Sciences @ LREC-COLING 2024*, pp. 1–11, 2024.

Bhushan Kulkarni, Sumit Agarwal, Abir De, Sourangshu Bhattacharya, and Niloy Ganguly. SLANT+: A nonlinear model for opinion dynamics in social networks. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 931–936, New Orleans, LA, USA, 2017. IEEE Computer Society. doi: 10.1109/ICDM.2017.117.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*, volume 36, pp. 51991–52008. Curran Associates, Inc., 2023.

Yu-Zheng Lin, Bono Po-Jen Shih, John Paul Martin Encinas, Elizabeth Victoria Achom, Karan Patel, Jesus Horacio Pacheco, Sicong Shao, Jyotikrishna Dass, Soheil Salehi, and Pratik Satam. LLM-MC-affect: LLM-based Monte Carlo modeling of affective trajectories and latent ambiguity for interpersonal dynamic insight. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27984–28000. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1290.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Fanhang Man, Huandong Wang, Jianjie Fang, Zhaoyi Deng, Baining Zhao, Xinlei Chen, and Yong Li. Context-aware sentiment forecasting via LLM-based multi-perspective role-playing agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1: Long Papers, pp. 2687–2703, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.136. URL https://aclanthology.org/2025.acl-long.136/.

Corrado Monti, Gianmarco De Francisci Morales, and Francesco Bonchi. Learning opinion dynamics from social traces. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 764–773, 2020.

Xinyi Mou, Xuanwen Ding, Qi He, Liang Wang, Jingcong Liang, Xinnong Zhang, Libo Sun, Jiayu Lin, Jie Zhou, Xuanjing Huang, and Zhongyu Wei. From individual to society: A survey on social simulation driven by large language model-based agents. *arXiv preprint arXiv:2412.03563*, 2024a.

Xinyi Mou, Zhongyu Wei, and Xuanjing Huang. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024b.

Maya Okawa and Tomoharu Iwata. Predicting opinion dynamics via sociologically-informed neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1306–1316. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539228.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, pp. 1–22, New York, NY, USA, 2023. ACM. doi: 10.1145/3586183.3606763.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. AgentSociety: Large-scale simulation of LLM-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*, 2025.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS 2012)*, volume 25, pp. 2951–2959. Curran Associates, Inc., 2012.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1631–1642. Association for Computational Linguistics, 2013. URL https://aclanthology.org/D13-1170/.

Petter Törnberg. Large language models outperform expert coders and supervised classifiers at annotating political social media messages. *Social Science Computer Review*, 43(6):1181–1195, 2024. doi: 10.1177/08944393241286471.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024. doi: 10.1007/s11704-024-40231-1.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, Prateek Gupta, Shuyue Hu, Zhenfei Yin, Guohao Li, Xu Jia, Lijun Wang, Bernard Ghanem, Huchuan Lu, Chaochao Lu, Wanli Ouyang, Yu Qiao, Philip Torr, and Jing Shao. OASIS: Open agent social interaction simulations with one million agents. *arXiv preprint arXiv:2411.11581*, 2024.

Table 4: Event windows of the two datasets and their role in the protocol.

Dataset	Window	Dates	Role
Election	T_1	Oct. 20–21, 2020	calibration (pre-election baseline)
	T_2	Oct. 23–24, 2020	calibration (reaction to the Oct. 22 debate)
	T_3	Oct. 29–30, 2020	calibration, *closest*
	T_4	Nov. 3–5, 2020	calibration (election day and vote counting)
	T_5	Nov. 7–8, 2020	*after*, covering Biden's victory declaration
Ukraine War	T_1	Aug. 20–30, 2022	input window (pre-counteroffensive stalemate)
	T_2	Sep. 10–26, 2022	input window (Kharkiv counteroffensive)
	T_3	Feb. 23–24, 2023	held out, *closest* (invasion's one-year anniversary)
	T_4	Feb. 26–Mar. 5, 2023	held out, *after* (the week after the anniversary)

A Dataset Windows

The event window each reported time point covers appears in Table 4.

The election dataset is the US Election 2020 Tweets data published by Hui (2020), the same source used by Man et al. (2025), and it covers all five election windows. The Ukraine War event is our own dataset, and we build it from the Ukraine Conflict Twitter Dataset, a public daily collection of posts about the war that covers all four of its windows. We keep the users who posted at least once in every window of Table 4, rank them by follower count, and take the top 3,000, which places the follower threshold at 5,280. Each agent's history is capped at 80 tweets, with a per-window budget of 25, 25, 10 and 20 tweets for the Ukraine T_1, T_2, *closest* and *after* windows, so a heavy poster in one window cannot crowd out their tweets from the others. The source carries no follow edges, so we build the feed as a language-matched proxy. For each agent we first take tweets whose text exactly matches another agent's original tweet, which recovers real retweet chains, and we fill the rest with random tweets in that agent's dominant language, to at most 30 followees per agent. The ground-truth labels of the two held-out windows come from annotating every tweet with the context-free prompt of Appendix J and taking the per-agent mode, with ties broken toward the more negative label.

The election run has five steps, one per window, with $\mathcal{T}_{\text{obs}}$ being T_1–T_4, T_3 reported as *closest* and T_5 as *after*. The Ukraine War run has four steps, T_1 and T_2 input windows and T_3, T_4 held out, reported as the *closest* and *after* columns of Table 1.

B Simulation Prompts

Both stages of Section 3.2 run on fixed hand-written templates, one pair for each dataset, and we reproduce the pair the election run uses, because every number we report depends on what the agents were asked. Only the bracketed fields change from step to step and from agent to agent, and no part of a prompt comes from an optimizer. The Ukraine War run uses its own copy of each template, and only two things change there: the event name in the fixed wording, and the two endpoints of the stance axis, which run from pro-Russia to pro-Ukraine. We therefore do not repeat the templates.

Stage 1. Stage 1 predicts the sentiment, so its system prompt carries the profile, the political identity, the temporal boundary and the role-playing rules. The same prompt withholds three fields that describe how an agent writes rather than what it feels: the language style, the emotional tone and the past tweets.

> You are role-playing a real Twitter user during the 2020 US presidential election.
> You are NOT a news editor, a debate judge, or a generic AI assistant.
>
> **[Your Twitter Profile]**
> Username: {name}
> Location: {address}
> Bio / Persona: {persona}
> Personality traits: {personality_features}
> Values: {values}

[Your Political Identity]
{stance_anchor}

[Temporal Anchoring - Read Carefully]
- The exact current date and event status are provided in each task's "Current Events" and "Temporal boundary" sections.
- React ONLY to what is known at that step. Never use knowledge from after the stated date.
- If the current step is before Election Day, you do not know the winner.
- If the current step is during vote counting, you know Election Day happened but the winner has not been projected yet.
- If the current step is after major networks projected a winner, you may react to that projection and its immediate aftermath.

[Role-Playing Rules]
- Always speak as this specific person. Never break character.
- Your opinions should be consistent with your persona and past tweets.
- You are reacting to what you see on your timeline RIGHT NOW, combined with the current news environment and the overall public mood.
- If new information contradicts your prior beliefs, you may acknowledge it, question it, or push back - as a real person would. Do not pretend you didn't see it.

The user message then supplies the seven signals of Section 3.2 in a fixed order.

{persona_text}

{stance_anchor}

[Current Situation - Your Primary Decision Driver]
Event: {event_context}

Your camp sentiment: {camp_sentiment}.

Your followees are reacting to the event:
{followee_cluster_summary}

[Your Historical Pattern - Context, NOT a Hard Constraint]
Your historical tweets tend to be distributed as: {emotional_baseline}.
This describes your PAST behavior. Today may be DIFFERENT. If the current event and your camp's mood point in a new direction, you should react to TODAY'S reality rather than mechanically repeating your history.

{sentiment_trajectory}{transition_hint}

[Decision Rule]
Weigh ALL signals together - the current event, your camp's mood, your followees' reactions, AND your personal history - to decide what sentiment your next tweet will express RIGHT NOW. Do NOT default to your most common historical sentiment; let the current situation genuinely influence you.

What sentiment will your next tweet express?

Output ONLY (three lines, nothing else):
[PREDICTED SENTIMENT: very_negative/negative/neutral/positive/very_positive]
[PREDICTED STANCE: -1.0 to 1.0]
[PREDICTED INTENSITY: strong/weak]

We fill each placeholder as follows.

- `persona_text`: the username, the location, the bio, the personality traits and the values.
- `stance_anchor`: the anchored political identity built from the agent's stance score.
- `emotional_baseline`: the agent's historical label distribution as percentages, for example *45% very_negative, 30% negative, 20% neutral, 5% positive, 0% very_positive*.

- `sentiment_trajectory`: one line per past window in the form *window: dominant label - reason*.
- `followee_cluster_summary`: up to five lines of the form *3 are furious about election fraud - "post text"*, one line per stance and topic group among the agent's followees, each with one representative post.
- `event_context`: the neutralized event report e^t.
- `camp_sentiment`: the percentage breakdown of the camp summary the agent is entitled to, for example *55% positive, 30% neutral, 15% negative*.
- `transition_hint`: an empty string at every step but the last one, where it names the agent's most recent sentiment and states that the event has shifted while the agent's own stance has not.

We read the reply through a fixed schema with three fields: the sentiment label, the stance as a number in $[-1, 1]$, and the intensity as *strong* or *weak*.

Stage 2. Stage 2 writes the post, so it restores the three fields that Stage 1 withholds and keeps the same role-playing rules. Its system prompt is therefore the Stage 1 system prompt with the language style, the emotional tone and the past tweets added back, and its user message carries the two style anchors, the event report and the predicted label as a hard constraint.

[Your Writing Style - Real Examples From Your Past Tweets]
Example 1 (your typical tone): "{anchor_dominant}"
Example 2 (another side of your voice): "{anchor_contrastive}"

[Current Events]
{event_context}

[CONSTRAINTS - You MUST Follow This]
Your tweet MUST express this sentiment: {required_sentiment}

Task:
Write ONE tweet about the 2020 US presidential election that is CONSISTENT with the constraint above.

Rules:
- The required sentiment is a HARD constraint. Your tweet must clearly express {required_sentiment} emotion.
- Match the writing style shown in the examples above. Use similar word choice, intensity, and emotional register.
- The event context is a neutral factual description. Different people react differently - react as someone with your political stance would.
- Do not output role-playing disclaimers or hashtag-only responses.
- Output ONLY the tweet text, followed by self-annotation metadata.

Output format:
tweet: <your tweet here>
[SENTIMENT: {required_sentiment}]
[STANCE: your stance as a number from -1.0 (pro-Trump) to +1.0 (pro-Biden)]

The required sentiment is the predicted label alone, and we withhold the predicted intensity from this prompt, so the model cannot soften an extreme label while it writes. The reply carries the tweet and two self-annotation tags, and we read the agent's stance coordinate from the Stage 1 prediction, with the `[STANCE]` tag as the fallback.

C Parameter Sensitivity at the Calibrated Point

All results reported in this paper use the single calibrated vector ϕ^*, fitted once on the election windows and transferred to the Ukraine War without re-fitting. No parameter in ϕ is switched to an event-dependent form for the forecast. An event-dependent parameterization, in which the injection strength adapts to the event without a second calibration, is a natural extension that we leave to future work.

Table 5: Sensitivity of the forecast to the trust radius τ and the homophily decay γ, each swept one at a time around ϕ^*. The third column of each panel reports the meso statistic that parameter most directly controls, whose observed target is the real-network row of Table 2(b). Lower JSD is better, and bold marks the calibrated setting.

(a) Trust radius τ.

τ	JSD $T_5 \downarrow$	Polarization
0.1	0.071	0.959
0.2 (calibrated)	**0.019**	**1.057**
0.3	0.081	0.991
0.5	0.109	0.948

(b) Homophily γ.

γ	JSD $T_5 \downarrow$	Echo chamber
0.5	0.103	0.483
1.07 (calibrated)	**0.019**	**0.581**
2.0	0.076	0.490

We sweep two of the five parameters one at a time around ϕ^* in Table 5, the trust radius τ and the homophily decay γ, because each of the two controls a meso statistic we can check against the real interaction graph. All five parameters are set by the same search of Section 3.5, and the remaining three stay at their calibrated values. We re-run the forecast for each setting. The forecast loss is minimized at the calibrated setting in both cases, and those same settings come closest to the observed meso statistics, so neither parameter is a flat direction along which the optimizer could have settled anywhere in its neighbourhood.

D Macro Statistics

The simulated macro-level distribution averages over *agents*, each of the $N = 3000$ agents contributing one label per step:

$$P^t_{\text{sim}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{e}_{s^t_i} \in \Delta^4, \tag{9}$$

where $\mathbf{e}_k$ is the standard basis vector of Section 3.2, Δ^4 is the 5-dimensional probability simplex, $s^t_i \in \{0, \ldots, 4\}$ is the sentiment label of agent i at step t, and inactive agents carry their last label forward, as described in Section 3.1. The real trajectory $\{P^t_{\text{real}}\}$ is instead a frequency over *posts*. For each event window t we collect the tweets W_t falling in that window and annotate each under a context-free prompt with a fixed annotator and author revision, as detailed in Appendix J, giving $P^t_{\text{real}} = |W_t|^{-1} \sum_{x \in W_t} \mathbf{e}_{\ell(x)}$ over the labels $\ell(x)$ of that window's posts. The calibration target of Equations 7 and 8 is this pair of statistics.

Both sides are labelled by the same model, Qwen3-32B, described in Section 4.1, which raises the question of circularity. If the annotator also generated the text, agreement could reflect a shared style rather than a correct forecast. The two sides in fact share no context. A real post is labelled by that context-free prompt, which receives the post text and nothing else. The simulated label s^t_i comes instead from the agent's own classification, conditioned on its persona, its stance score, its historical sentiment baseline and trajectory, the sentiment clusters its followees currently occupy, the event report, and the camp summary it is entitled to, as described in Section 3.2. It never sees a real post from the window being forecast, and the per-agent ground truth is read only by the scoring functions, never by the prompts. Agreement between $\{P^t_{\text{sim}}\}$ and $\{P^t_{\text{real}}\}$ therefore cannot arise from the annotator recognizing its own output.

E Trajectory-Level Metrics

The trajectory-level metrics discussed in Section 4.2 appear in full in Table 6, together with the individual baselines of our comparison.

Trajectory-level W_1. The W_1 column applies the per-window distance of Section 3.5 to each of the five event windows and averages the five values, $W_1 = \frac{1}{5} \sum_{t \in \{T_1, \ldots, T_5\}} W_1\big(P^t_{\text{sim}},\ P^t_{\text{real}}\big)$. It

Table 6: Macro-level trajectory alignment on the 2020 election, with the trajectory-level W_1 and turning-point metrics. JSD $\mathcal{T}_{\text{obs}}$ averages the four calibration points, W_1 and TP-F1 the five event windows, and "–" is undefined for single-time-point methods. Lower is better for JSD and W_1, higher for TP-F1, and best in bold.

Method	JSD $\mathcal{T}_{\text{obs}}$	JSD T_5	W_1	TP-F1
B1 HK numerical	0.220	0.330	0.814	0.312
B2 FJ + static topology	0.139	0.202	0.809	0.342
B3 Single-step MPR	0.104	0.127	–	–
B4 No macro alignment	0.113	0.142	0.903	0.747
B5 Static followee	0.243	0.320	1.093	0.857
Ours	**0.012**	**0.019**	**0.757**	**1.000**

therefore differs from the $\mathcal{T}_{\text{obs}}$ terms of Equation 8 only in the windows it covers, and it is computed on the un-reweighted simulated trajectory for the reason given in Appendix F.

Turning-point F1. Each trajectory is summarized by the mean sentiment of its distribution at each window, $m^t = \sum_{k=0}^{4} k\, P_k^t$ for the windows $t = T_1, \ldots, T_5$ in order. An interior window T_k with $1 < k < 5$ is a turning point when the direction of that series reverses there, $(m^t - m^{t-1})(m^{t+1} - m^t) < 0$, so the turning points of a trajectory are the local extrema of its mean sentiment. Let $\mathcal{R}$ be the turning windows of the real trajectory and $\mathcal{S}$ those of the simulated one. TP-F1 takes $\mathcal{R}$ as the reference set and $\mathcal{S}$ as the prediction and reports the F1 of that match, $2\,|\mathcal{R} \cap \mathcal{S}| / \big(|\mathcal{R}| + |\mathcal{S}|\big)$. A method that produces a single time point has no trajectory to summarize, which is the "–" of its column.

F Loss Details

Calibration target. $\mathcal{L}_{\text{macro}}$ is evaluated on the un-reweighted simulated trajectory, not on the importance-reweighted one of Equation 7, because the reweighting is an observation-time correction applied after a step has been simulated whereas the objective must vary with ϕ. Scoring the reweighted trajectory instead would make all three of its terms vanish identically, the reweighted distribution being equal to P_{real}^t by construction. All divergences use the natural logarithm, so JSD is in nats.

Micro-level fidelity term.

$$\mathcal{L}_{\text{micro}}(\phi) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \sum_{t \in \mathcal{T}_{\text{micro}}} D_{\text{KL}}\big(\mathbf{e}_{s_i^{\text{real},t}} \,\big\|\, \hat{p}_i^t(\phi)\big), \tag{10}$$

where $\hat{p}_i^t(\phi)$ is the five-class softmax that a post-hoc sentence classifier assigns to the post agent i actually generated at step t, not to the Stage 1 distribution, which is discrete by construction, $\mathcal{A}$ is the set of agents for which a per-agent ground-truth label exists, and $\mathcal{T}_{\text{micro}}$ the annotated points it covers.

Protocol regimes. We separate two regimes and report their scores separately. The post-observation update applies at $t \in \mathcal{T}_{\text{obs}}$, where the reweighting of Equation 7 is applied and its output, not the free-running simulation, conditions the next step. The free forecast covers the windows no observation reaches, T_5 for the election run and T_3–T_4 for the Ukraine War run. Parameters are frozen at ϕ^*, the information boundary is frozen at the forecast point so that agents are not reweighted at or beyond it, and no ground-truth label enters the forward pass. The forecast input tells the model that an event occurred and its neutralized description, but not its sentiment consequence, making T_5 a scenario-conditional forecast under a known event rather than an unconditional one.

G Co-Evolution Details

State coordinates. The first coordinate of $\mathbf{o}_i^t$ is taken from the structured stance prediction of Stage 1 as a continuous scalar, bypassing discrete-probability approximations. When no such prediction is available it falls back to the LLM self-annotation score of the generated post, and then to its BERT polarity.

Per-agent coefficients and the event map. The two per-agent coefficients of Equation 4 are read off each agent's history before calibration rather than fitted, so they introduce no free parameter beyond ϕ,

$$\lambda_i = 0.1 + 0.8\,\sigma\big(-\mathrm{Var}(s_i^{\text{hist}})\big), \qquad \mu_i = \tanh(f_i/5), \tag{11}$$

where $\mathrm{Var}(s_i^{\text{hist}})$ is the variance of the five-class BERT labels of the last thirty historical posts of i, so a steady record makes an agent harder to move, and f_i is its historical posting rate in posts per day, so a frequent poster reacts more strongly to an event. The event report e^t carries a polarity $\pi_t \in [-1, 1]$ and an intensity $\iota_t \in [0, 1]$ assigned to its window in the event file, and the map is fixed as

$$\psi(e^t) = \pi_t\,\iota_t\,[1,\ 0.5]^\top, \tag{12}$$

so an event displaces both opinion coordinates toward its polarity, the second at half the rate of the first. Both numbers come from the event file and reach this prior only. They never enter a prompt, so the neutralized report the model reads still gives away no sentiment consequence.

Edge dissolution. An existing edge is removed with probability

$$\pi_{ij}'^t = \rho_t'\,\sigma\big(b_0 + \kappa\,(d_{ij}^t - \tau)\big), \qquad \Pr\big[(i \to j) \notin E_t \mid (i \to j) \in E_{t-1}\big] = \mathbb{1}\big[d_{ij}^t > \tau\big]\ \min\{0.3,\ \pi_{ij}'^t\}, \tag{13}$$

where $b_0 = -2.0$ is a fixed constant. The indicator gates whether a removal is attempted at all, and past the trust radius the probability keeps rising with the divergence $d_{ij}^t - \tau$ at sensitivity κ, from a low base at $d_{ij}^t = \tau$.

Density factors. Both rules are scaled by the graph's distance from its target density 0.05. Formation uses $\rho_t = \min\{1,\ 0.05/\mathrm{density}(G_t)\}$, floored at 0.1, which damps growth once the graph exceeds the target, while dissolution uses $\rho_t' = \max\{1,\ \mathrm{density}(G_t)/0.05\}$, capped at 2, which accelerates pruning in the same regime. Both equal 1 at or below target.

Deterministic $T{=}0$ mode. For calibration both rules are applied as hard thresholds rather than samples. An edge is added when $\pi_{ij}^t > 0.5$ and removed when $\pi_{ij}'^t > 0.5$, in each case before the cap, so that only pairs the probabilistic rule would treat at its maximum rate change state and the graph is reproducible across evaluations.

H Fixed Design Constants

The relevance score of Section 3.3 is governed by hand-set constants rather than fitted parameters. The four feature weights are $w_c = 0.40$, $w_a = 0.25$, $w_l = 0.20$, and $w_q = 0.15$, ordered $w_c > w_a > w_l > w_q$. The raw score is mapped by $\sigma\big(5\,(S_i - 0.5)\big)$, which for a convex combination S_i of features in $[0, 1]$ already lies in $[0.076, 0.924]$, so no agent is treated as fully open or fully closed to its camp summary. Within the social-influence feature the two components are combined as $l_i = 0.6\,\hat{d}_i + 0.4\,\hat{f}_i$. None of these enter the calibrated vector ϕ of Section 3.5, as they are fixed across all runs and exposed as configuration flags. The same policy covers the co-evolution rules of Appendix G: their constants are set once and do not enter ϕ either.

The loss weights are likewise fixed rather than calibrated, at $\lambda_1 = 0.3$, $\lambda_2 = 0.2$, $\lambda_{\text{micro}} = 0.05$, $\rho_1 = 0.1$, and $\rho_2 = 0.05$, with the parameter prior $\phi_0 = \{\tau = 0.3,\ \beta = 0.5,\ \gamma = 1.0,\ \eta = 0.5,\ \kappa = 1.0\}$ in the penalty $\rho_2\|\phi - \phi_0\|_2^2$.

The activation law of Section 3.1 takes Δt per window, as the real-world span that window covers, two to three days for the election windows and two days to about two and a half weeks for the Ukraine War windows, so that f_i stays in posts per day and the activation probability is on the same per-day scale in both runs.

The teacher-forcing strength of Equation 2 takes $\alpha = 0.2$, a fixed launch setting rather than a calibrated parameter.

I STYLE-EXEMPLAR ANCHOR SELECTION

The two style-exemplar anchors of the two-stage decoupled prediction component in Section 3.2 are the top-scoring entries of i's historical tweets:

$$\text{anchor}_i^1, \text{anchor}_i^2 = \underset{x \in \text{history}_i}{\arg\,\text{top-2}} \; 1.25\, g_{\text{stance}}(x) + g_{\text{contrast}}(x) + 0.25\, g_{\text{recency}}(x) + 0.25\, g_{\text{intensity}}(x), \quad (14)$$

where for a tweet x with polarity $\text{pol}(x)$ and label $\ell(x)$, $g_{\text{stance}}(x) = 1 - \min(1, |\text{pol}(x) - \overline{\text{pol}}_i|/2)$ measures closeness to i's mean polarity $\overline{\text{pol}}_i$, $g_{\text{contrast}}(x) = \min(1, |\ell(x) - \text{mode}(\ell_i)|/4)$ rewards deviation from i's modal sentiment, $g_{\text{recency}}(x) = \text{rank}(x)/|\text{history}_i|$ is its normalized position in the history, and $g_{\text{intensity}}(x) = |\text{pol}(x)|$ favours intensity. The weights prioritize stance direction over recency and intensity. Scoring is a stable sort, so ties resolve to the earlier tweet. Selected texts are truncated to the platform character limit and exact duplicates are skipped, so the two anchors are always distinct.

J ANNOTATOR VALIDATION

Every number reported in Section 4.2, Section 4.3 and Section 4.4 is measured against automatically annotated ground truth, so the annotator itself has to be validated first. We validate the annotators on a stratified sample of 100 tweets re-annotated by a human expert, and report their per-class recall in Figure 4. The bias it documents is what motivates the separate reporting of BERT-scored rows.

The annotating prompt is fixed, and it receives the post text and nothing else:

> You are an expert sentiment analyst for social media data about the Russia-Ukraine war.
> Classify the following tweet posted during the Russia-Ukraine war.
>
> Tweet: "{tweet}"
>
> Task:
> 1. SENTIMENT: Rate the emotional tone of this tweet.
> Choose ONE: very_negative, negative, neutral, positive, very_positive
> 2. STANCE: Estimate the stance toward the war on a scale from -1.0 to 1.0.
> -1.0 = strongly pro-Russia / anti-Ukraine
> 0.0 = neutral / unclear / not about the war
> +1.0 = strongly pro-Ukraine / anti-Russia
>
> Output exactly TWO lines, nothing else:
> [SENTIMENT: your_label]
> [STANCE: your_number]

The election dataset uses the same prompt with the first two lines and the stance axis rewritten for the election.

BERT reaches the lowest overall accuracy at 69.0% and its recall stays below 40% on every non-negative class, assigning 10 of the 18 tweets that the human rates most positive in class $+2$ to the most negative class. The error is therefore systematically one-directional rather than random, since positive sentiment is absorbed into the negative class. Class -2 holds 60 of the 100 sampled tweets, so a rule that always answers -2 already scores 60.0%, and BERT's 69.0% sits only 9 percentage points above that. Accuracies measured against BERT annotations are consequently not on the same scale as those measured against LLM annotations. The LLM annotator we use reaches 87.0% overall accuracy, 27 percentage points above the majority-class baseline, and DeepSeek 83.0%. Neither collapses a class to zero recall, whereas BERT misses one class entirely. The bias propagates to any comparison scored against it. Through the BERT annotator the single-step baseline reaches 34.2% 5-class accuracy at T_5 against our method's 63.4%, so the micro-level ordering of Section 4.3 is unchanged under either annotator.

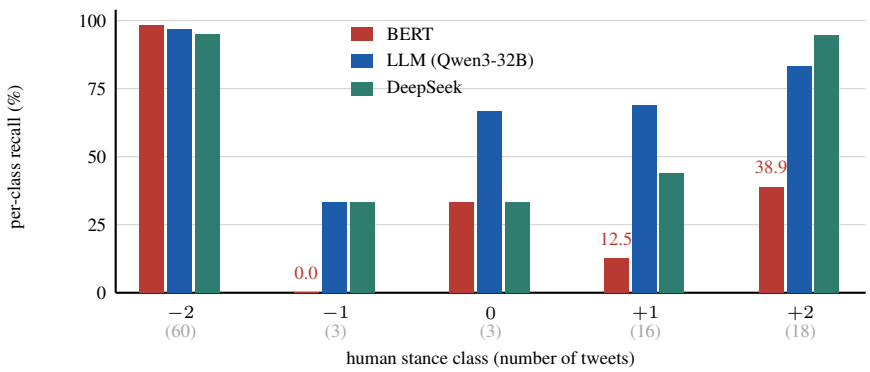

Figure 4: Per-class recall of the three annotators on 100 human-annotated tweets. The two smallest classes contain three tweets each.

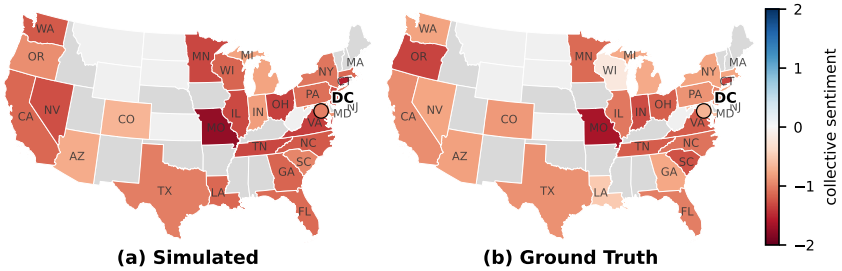

Figure 5: Per-state collective sentiment on the 2020 election dataset, our simulation (a) against the real population (b). States are coloured by the mean sentiment of their users on the five-class scale rescaled to $[-2, +2]$, grey marking states with fewer than five users (Alaska and Hawaii omitted).

K Regional Coarse-Graining

In Figure 5 we apply a second and independent coarse-graining, which the interaction graph does not determine. It matters because a simulation can match network-level aggregates while placing the agents themselves in the wrong regions, and because a region is a coarse-graining an outside observer can check without access to the simulation's own graph.

Three properties of this map are worth separating. First, the sign structure. The real population on Oct. 29–30 is negative in every state, with no state mean above -0.13, and the simulation reproduces this without a single region flipping sign. Second, the level. The two maps differ by 0.14 in their overall mean on the $[-2, +2]$ scale, at -1.07 in the simulation against -0.93 in the ground truth. Third, the spatial ordering, where the agreement is real but weaker. Over a coarse grid covering the mapped area the two agree at a mean absolute deviation of 0.22 and a correlation of $r = 0.47$, with 95% of the cells within 0.5 and none beyond 0.93. We report the correlation rather than only the deviation because it is the honest summary of this panel. The simulation captures which regions are most negative and by how much in level, but its region-level ordering is not a strong claim, which is expected when the ground truth is a per-state mean over a two-day window.

L Limitations and Future Work

Several limitations point to future work. The two datasets cover a narrow range of events and one language, and the simulation counts agents where the real data counts posts. Ground-truth labels remain machine-generated, though author-revised, and the annotator validation of Figure 4 covers only the 2020 election sample. Both datasets follow the same annotation protocol and use the same annotator, so the validation applies to both.