Architecture of the Opinion Dynamics Simulator (NetLogo)

Version_20260115-3_Simulator v6_EN.nlogox3d (code excerpt) — 2026-01-17

1) Scientific Purpose and General Principle

The simulator models the diffusion and transformation of bipolar opinions within a population of agents connected by a dynamic social network. It implements two layers of mechanisms: (i) a "direct" layer in which opinion adoption depends on prevalence, opinion distance, influence, and group parameters; (ii) an optional "memes" layer (use-memes?) in which opinion and prevalence emerge from internal stocks of pro/anti representations, with an explicit distinction between quantity (prevalence) and cognitive weight (impact on opinion).

2) Data Structures and State Variables

2.1 Global Variables (globals)

Global variables are used to (a) parameterize the experiment, (b) store simulation counters, (c) control external events and meme injections, and (d) manage CSV export.

- Trial and dynamics counters: iter, try, change, total, inversion, major, fractale, Interactions, %Major.
- Network: links-dead, links-create, meta-links, meta-agents, meta-create.
- Export: csv-export, csv-basename, csv-file, csv-open?.
- Scheduling parameters: tick-event (initialized via event-init), repeat-event, event-pace; inject-tick, inject-pace.
- Injection parameters (memes): inject-prob-max, inject-amount, inject-weight, inject-sign, inject-low_meme…inject-high-prev.
- Meme parameters: meme-max, meme-gain, meme-anti-leak, meme-decay; meme-weight-* (mean/sd/min/max).

2.2 Agent-Specific Variables (turtles-own)

Each agent carries an ideological state (opinion) and cognitive/social attributes:

- opinion ∈ [-1, +1]: position on the bipolar continuum.
- prevalence ∈ [0, 99]: intensity/salience (level of "representations").
- influence ∈ [0, 1]: persuasive capacity.
- agent-type: initial side label (Right side / Left side).
- meme-plus / meme-minus: quantitative stocks (derived prevalence).

- meme-plus-w / meme-minus-w: weighted stocks (derived opinion construction).
- tx-bonus: cumulative transmission bonus (reward mechanism).
- Utility variables: opinion-previous, influence-previous, old-opinion, proposed-opinion; 3D coordinates x3d/y3d/z3d.

3) Initialization Chain

3.1 setup Procedure

setup resets the environment (clear-all), sets prevalence bounds, initializes counters, applies default values to missing parameters, sets tick-event to event-init, and prepares the capacity of meta-influencers to maintain a number of links (meta-links) when vary-influence or meta-ok is active.

Structural points:

- set-background-black: standardizes the 3D background (black patches).
- create: instantiates the population, initializes memes and 3D positioning.
- rapport: initializes output headers (Statistics / Values / File modes).

3.2 create Procedure and Meme Initialization

create generates pop/2 agents on each side (Right/Left). Each agent receives a random initial opinion, prevalence, and influence. Memes are then initialized consistently with (prevalence, opinion) via initial-prevalence-to-memes, with a pro/anti distribution dependent on |opinion|. influenceurs selects the initial meta-influencers, after which the network is created/adjusted via update-networks, followed by recolor-links and apply-link-visibility.

4) Scheduling: Main Loop go

go is the temporal engine. At each tick/iteration, it executes operations in an order that is essential for result interpretation.

Logical order (simplified):

- Increment counters (iter) and update aggregate indicators (Interactions, %Major).
- Open CSV at the first tick (csv-begin) if enabled.
- Schedule/activate external events: auto_event + tick-event; repetition controlled by repeat-event and event-pace.
- Meme injection (if inject-on? / inject-tick / inject-pace configured) via inject-memes.
- Activate meta-links (meta) if meta-ok = true.
- Opinion dynamics: update-opinions.

- Network dynamics: update-networks if network = true; then recolor-links and apply-link-visibility.
- Outputs/measures: statistics, tick, fractal update, optional reset (cumulative), coloring, graphs, CSV export (csv-row).
- Multi-trial management: counter reset and tick-event reset to event-init at each new try.

5) Opinion Transmission — Direct Layer

5.1 Core Diffusion: update-opinions

update-opinions executes the micro-dynamics of adoption. Each agent (receiver) selects a linked neighbor (target) and computes an adoption probability p-adopt. If the random draw is favorable, the agent updates its opinion (subject to meta safeguards), updates counters (total/change), and may trigger secondary mechanisms (reward, influence variation, prevalence modulation, noise).

Main steps (code-level):

- Neighbor selection: let target one-of link-neighbors.
- Prevalence gradient: dprev = max(0, ([prevalence] of target - prevalence)) / max-prevalence.
- "Memetic" distance (proxy): dmem = abs(abs(opinion) - abs([opinion] of target)).
- Base probability: base-prob = dprev * prevalence-weight.
- Polarization penalty: pol-penalty = max(adoption-floor, 1 - polarization-factor * dmem).
- Influence + reward: multiplication by [influence] of target and (1 + [tx-bonus] of target).
- Group impact: group-alignment-effective(self, sign(target.opinion)) weighted via group-impact-weight and group-impact-alpha.
- Draw: if random-float 1 < p-adopt [... adoption ...].

5.2 Meta Veto and Reinforcement Without Inversion

The maybe-set-opinion procedure centralizes the metablock constraint. If metablock is ON and the agent is meta (meta?), a sign change is prevented. In "solution 1," the meta agent may reinforce (increase magnitude) without polarity switching: the opinion keeps its sign and takes max(|old|, |new|).

5.3 Rewards and Influence Dynamics (vary-influence)

After a successful adoption, the model may: (a) increase the emitter's tx-bonus (target) by reward-step up to reward-cap (optionally decaying via reward-decay); (b) increase the receiver's prevalence via reward-prev-delta; (c) adjust influence if vary-influence = true, increasing it toward 1 on success and decreasing it on failure, with automatic promotion to "meta" status when influence reaches 1.

6) "Memes" Layer: Quantitative Prevalence and Weighted Opinion

When use-memes? is ON, adoption no longer directly copies target.opinion. Instead, transmission operates on internal representation stocks (memes) in the receiver, and the state (opinion, prevalence) is recomputed from these stocks.

6.1 Two Stock Levels: Quantity vs Weight

The model explicitly distinguishes:

- Quantity of memes (meme-plus, meme-minus): determines prevalence.
- Weight of memes (meme-plus-w, meme-minus-w): determines opinion polarity and intensity.

6.2 Key Procedures

- init-memes-from-state: initializes stocks (quantities and weights) from (prevalence, opinion) and meme-weight-mean.
- draw-meme-weight: samples a meme weight from meme-weight-mean ± meme-weight-sd, bounded by meme-weight-min/max.
- transmit-memes(emitter): increases receiver stocks according to emitter sign; applies an "anti-leak" to the opposite stock; caps at meme-max.
- recompute-from-memes: (i) computes opinion from weighted stocks; (ii) computes prevalence from quantitative stocks; (iii) applies maybe-set-opinion (meta veto).
- decay-memes: applies forgetting (meme-decay) to quantitative and weighted stocks.

6.3 Structuring Formulas

Derived opinion (weighted):

opinion = (meme-plus-w - meme-minus-w) / (meme-plus-w + meme-minus-w)

Derived prevalence (quantitative):

prevalence = ((meme-plus + meme-minus) / meme-max) * 99

6.4 Controlled Meme Injection

The inject-memes procedure allows artificial addition of memes (quantity + weight) to a subset of agents defined by opinion and prevalence bounds (inject-low_meme…inject-high-prev). Intensity is regulated by inject-amount, exposure probability by inject-prob-max, imposed cognitive weight by inject-weight, and sign by inject-sign (plus/minus).

7) Social Network: Creation, Removal, Bridges, and Meta-Links

7.1 update-networks

- update-networks implements endogenous network dynamics:
- Removal: links whose opinion distance exceeds link-removal-threshold may be removed (probabilistically).
- Creation: links are created either by homophily (pool-homo) or by inter-camp bridges (pool-bridge) with probability bridge-prob, subject to linksup/linksdown constraints.

7.2 Meta and Meta-Link Control

meta (called if meta-ok = true) enforces links toward yellow agents, capped by meta-links. Coloring (recolor-links) and visibility (apply-link-visibility / show-links?) then synchronize representation.

8) External Events: Shocks, auto_event, Repetition

The event procedure applies an exogenous shock to opinion (event_size) and/or prevalence (prev_change) to filtered agents: either via meme_set/to_left or via bounds low_meme–high_meme and low-prev–high-prev. The parameter event-prob-max limits the fraction effectively exposed at each trigger.

- Repetition is handled by the auto_event logic in go:
- If auto_event is ON, the event triggers when iter = tick-event.
- If repeat-event is ON, tick-event is incremented by event-pace.
- At the end of a trial (next try), tick-event is reset to event-init to ensure comparability.

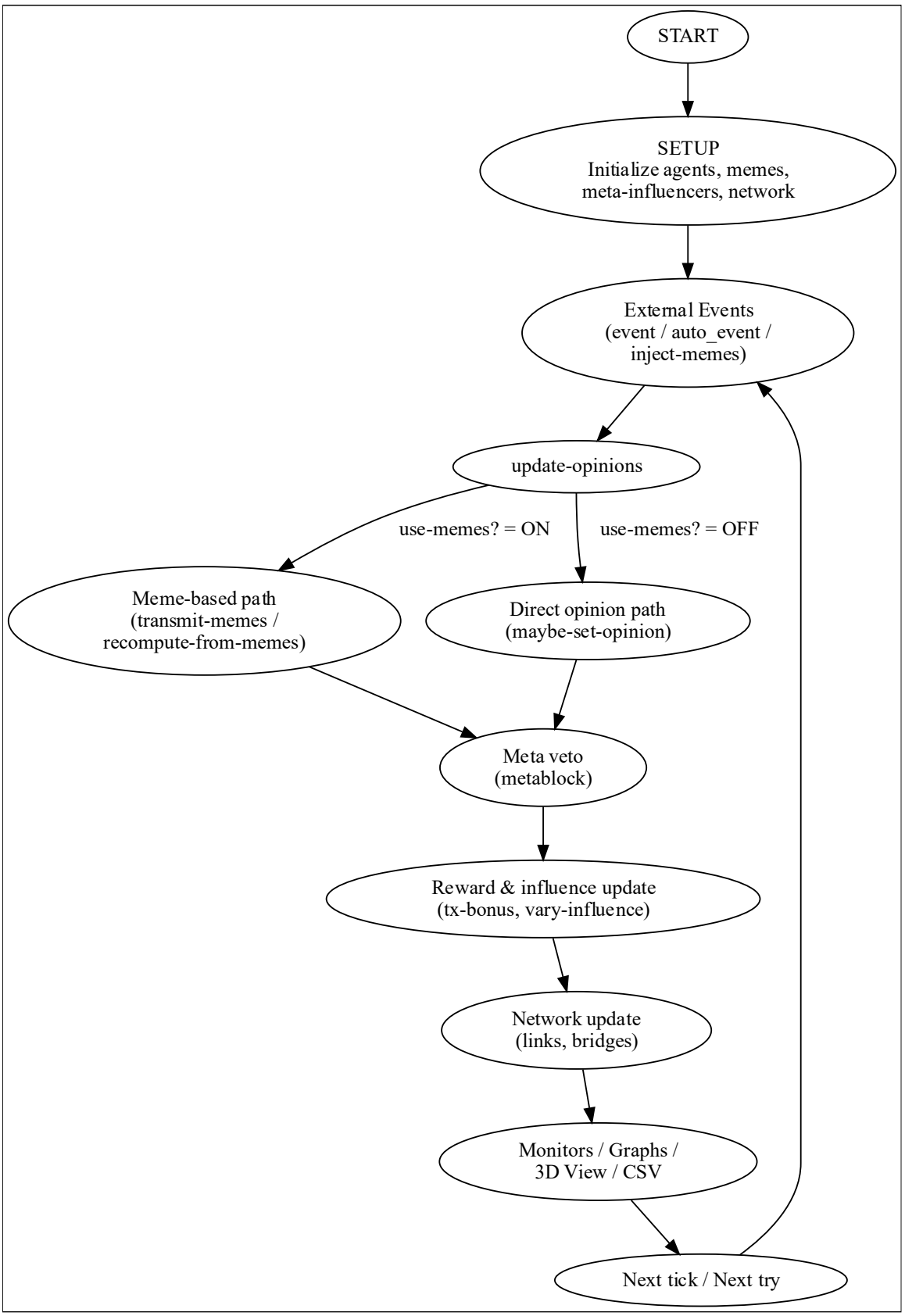
9) Outputs, Measures, and Export

The simulator provides several output channels:

- Output area (Statistics / Values / File) via rapport and compute-statistics.
- Monitors and graphs (notably for meme indicators).
- CSV export per trial (csv-begin / csv-row / csv-end).
- Dedicated reporters (e.g., highly-saturated-agents-pct) provide global measures useful for interpretation.

10) Technical Recommendations (Robustness and Interpretability)

- Ensure event-pace and inject-pace are positive (or rounded) to avoid inconsistent scheduling loops.
- When use-memes? is OFF, call init-memes-from-state only if meme indicators are displayed (avoids perceived inconsistencies).
- For reproducibility, consider exposing a seed (random-seed) in the UI for BehaviorSpace experiments.
- Explicitly document the quantity/weight distinction of memes in the Info tab to avoid “prevalence vs opinion” confusion.


START
SETUP
Initialize agents, memes,
meta-influencers, network
External Events
(event / auto_event /
inject-memes)
update-opinions
use-memes? = ON
use-memes? = OFF
Meme-based path
(transmit-memes /
recompute-from-memes)
Direct opinion path
(maybe-set-opinion)
Meta veto
(metablock)
Reward & influence update
(tx-bonus, vary-influence)
Network update
(links, bridges)
Monitors / Graphs /
3D View / CSV
Next tick / Next try