

EQBSL: Against Trust Scores

Evidence, Uncertainty, and Trust Flow in Dynamic Networks

Oliver C. Hirst
Independent Researcher

December 2025

Abstract

Most “trust scores” are numerology with a user interface: confident decimals that conceal their own provenance. Evidence-Based Subjective Logic (EBSL) is a corrective, because it refuses to treat trust as a mystical scalar and instead anchors it in manipulable, auditable evidence. This note presents *EQBSL*, a systems-oriented extension that lifts evidence-based trust into a structured, vectorised, operator-defined form suitable for dynamic graphs and hypergraphs, and for downstream machine learning via stable trust embeddings. The Subjective Logic formalism is credited to Jøsang, and the evidence-flow reformulation to Škorić et al.; EQBSL is proposed here as an explicitly engineered extension built on that intellectual substrate.

Keywords: trust networks; subjective logic; evidence flow; reputation systems; dynamic graphs; hypergraphs.

1 Introduction

Trust, in computation, has a habit of being treated as though it were weather: a quantity one can simply “measure” and then proceed as if the measurement were innocent. In reality it is a ledger of claims, counterclaims, and uncertainty—and any framework that pretends otherwise ends up smuggling assumptions through the back door.

Subjective Logic provides a compact algebra for reasoning under uncertainty, including trust transitivity and opinion fusion [1]. Evidence-Based Subjective Logic (EBSL) strengthens this foundation by centring the calculus on *evidence flow*, addressing structural failure modes that arise when applying classical discounting to general trust networks [2, 3]. One may dislike the conclusions of an evidence ledger, but at least one can interrogate the entries.

This paper proposes *EQBSL* as a pragmatic extension of EBSL for modern systems where: (i) interactions evolve over time and “yesterday” is not merely a poetic metaphor, (ii) evidence is multi-dimensional and context-bound, (iii) multi-party interactions are native (hyperedges rather than forced pairwise fictions), and (iv) downstream models benefit from stable vector embeddings per agent that retain a clear chain of custody back to evidence.

2 Subjective Logic (binomial opinions)

Classical Subjective Logic represents an opinion of agent A about proposition X as a 4-tuple

$$\omega_X^A = (b, d, u, a), \tag{1}$$

where b is belief, d is disbelief, u is uncertainty, and a is the base rate (the prior probability when evidence is uninformative). The opinion satisfies

$$b + d + u = 1. \quad (2)$$

Subjective Logic defines operators for consensus (combining independent opinions), recommendation / trust transitivity, discounting (weighting by trust in the source), and fusion (combining multiple sources into one opinion) [1]. The attraction here is not that it makes doubt disappear, but that it forces doubt to be accounted for.

3 Evidence-Based Subjective Logic (EBSL)

EBSL reframes opinions explicitly in terms of evidence counts [2, 3]. For a binary proposition, interpret:

- r as the amount of positive evidence,
- s as the amount of negative evidence,
- $K > 0$ as a fixed normalisation / prior-weight constant (often denoted W in Subjective Logic literature [1]).

The mapping from evidence to a Subjective Logic opinion is:

$$b = \frac{r}{r + s + K}, \quad d = \frac{s}{r + s + K}, \quad u = \frac{K}{r + s + K}. \quad (3)$$

Key properties:

- **Evidence is additive:** independent observations add evidence (e.g. (r, s) sums). This is the nearest thing to honesty that many reputation systems ever achieve.
- **Uncertainty decays with evidence:** u decreases as $r + s$ grows; confidence must be paid for in evidence, not declared by fiat.
- **Network computation in evidence space:** consensus, propagation, and discounting can be implemented as evidence-flow operations and then mapped back to opinions.

On a trust network:

- Nodes are agents V ,
- edges carry evidence about interactions (successful trade, failed settlement, timely delivery, malicious proposal, and the rest of the human comedy),
- the engine propagates, aggregates, and discounts evidence along the graph to produce trust opinions.

4 EQBSL: a vectorised operator view

EBSL provides a clean basis for trust computation on arbitrary networks [2]. EQBSL extends that basis with explicit structure for time, hyperedges, context, and vector embeddings—because the real world is not obliged to fit neatly into a scalar.

4.1 Definition

Definition (EQBSL). EQBSL lifts evidence-based opinions into a structured, vectorised, operator form over dynamic graphs and hypergraphs, with explicit temporal and contextual semantics.

Concretely, EQBSL introduces:

1. **Evidence tensors (vector evidence).**
2. **A lift from evidence to opinions with explicit aggregation semantics.**
3. **An operator view of trust propagation/update.**
4. **Hypergraph-aware evidence handling.**
5. **Node-level trust embeddings as primary outputs.**

The aim is not rhetorical flourish; it is to make every moving part explicit enough to be tested, audited, and (where necessary) disputed.

4.2 Evidence tensors

Instead of scalar (r, s) per relationship, maintain a richer evidence vector

$$\mathbf{e}_{ij}(t) \in \mathbb{R}^m, \quad (4)$$

for each ordered pair of agents (i, j) at time t . Components can include counts or scores for: successful vs. failed trades, on-time vs. late delivery, governance alignment votes, dispute outcomes, attestation patterns, and time-weighted recency terms. The point is not to inflate dimensionality for its own sake, but to prevent distinct kinds of evidence from being prematurely mashed into a single number and called “truth.”

To recover the scalar EBSL evidence parameters (r, s) needed by (3), define nonnegative aggregation functionals

$$r_{ij}(t) = \phi_+(\mathbf{e}_{ij}(t)), \quad s_{ij}(t) = \phi_-(\mathbf{e}_{ij}(t)), \quad (5)$$

where ϕ_+, ϕ_- are application-defined (often linear) maps. A common simple choice is a weighted projection:

$$r_{ij}(t) = \langle \mathbf{w}^+, \mathbf{e}_{ij}(t) \rangle, \quad s_{ij}(t) = \langle \mathbf{w}^-, \mathbf{e}_{ij}(t) \rangle, \quad (6)$$

with $\mathbf{w}^+, \mathbf{w}^- \in \mathbb{R}_{\geq 0}^m$. These weights are not an embarrassment to be hidden; they are an explicit declaration of what your system counts as evidence.

4.3 Lift to opinions

Define a mapping

$$\Psi : \mathbf{e}_{ij}(t) \mapsto \omega_{ij}(t) = (b_{ij}(t), d_{ij}(t), u_{ij}(t), a_{ij}), \quad (7)$$

implemented by:

$$(b_{ij}(t), d_{ij}(t), u_{ij}(t)) = \left(\frac{r_{ij}(t)}{r_{ij}(t) + s_{ij}(t) + K}, \frac{s_{ij}(t)}{r_{ij}(t) + s_{ij}(t) + K}, \frac{K}{r_{ij}(t) + s_{ij}(t) + K} \right), \quad (8)$$

with base rate a_{ij} chosen per domain (or held constant if not used). What matters is that the lift is specified and reproducible: anyone can retrace the steps from evidence to opinion without consulting a priesthood.

4.4 Operator view of trust propagation

Rather than ad-hoc “iterate until convergence” rules (a phrase that often means “until we stop looking”), EQBSL models trust-state evolution as an operator acting on the global evidence state.

Let the time-indexed graph be

$$\mathcal{G}_t = (V, E_t), \quad (9)$$

and let $\mathcal{E}_t$ denote the collection (tensor field) of all evidence vectors on directed edges:

$$\mathcal{E}_t = \{\mathbf{e}_{ij}(t)\}_{(i,j) \in V \times V}. \quad (10)$$

Define a global update operator

$$F : \mathcal{E}_t \rightarrow \mathcal{E}_{t+1}, \quad (11)$$

so that in practice:

$$\mathcal{E}_{t+1} = F(\mathcal{E}_t, \text{new_events}_t). \quad (12)$$

The operator F encodes: EBSL-style evidence combination rules [2], temporal decay, context-dependent weighting, and (optionally) hyperedge aggregation (next section). The result is a *well-defined* state update step suitable for batch or streaming systems, and mercifully resistant to hand-waving.

4.5 Hypergraph-aware trust

Many interactions are natively multi-party (DAOs, multi-sig execution, group swaps, committees). Forcing these into pairwise edges tends to erase accountability structure and then wonder why the model behaves strangely.

EQBSL represents evidence per hyperedge:

$$\mathbf{e}_h(t), \quad h \subseteq V, |h| \geq 2, \quad (13)$$

and defines a decomposition rule that allocates hyperedge evidence into pairwise or groupwise evidence tensors. One generic form is:

$$\mathbf{e}_{ij}(t) \leftarrow \sum_{h \ni i,j} \alpha_{ijh} \Pi_{ij}(\mathbf{e}_h(t)), \quad (14)$$

where Π_{ij} is a projection / attribution map and α_{ijh} are allocation coefficients (e.g. symmetric, role-weighted, or protocol-specific). This is where protocols confess their actual social structure.

4.6 Node-level trust embeddings

EQBSL outputs stable per-agent embeddings:

$$\mathbf{u}_i(t) = \Gamma(i, \mathcal{E}_t, \mathcal{G}_t) \in \mathbb{R}^{d_u}, \quad (15)$$

where Γ aggregates how the network “feels” about i across evidence/opinions, structural position, and temporal patterns. These embeddings are intended as the main interface for downstream components (e.g. ranking, anomaly detection, clustering, recommendation, or human-facing summarisation layers). The crucial requirement is that they remain tethered to the evidence ledger, so that an embedding can be challenged with more than indignation.

5 Interface assumptions

A practical EQBSL system provides:

- per-agent trust embeddings $\mathbf{u}_i(t)$,
- optional pairwise opinions $\omega_{ij}(t)$ (or evidence tensors $\mathbf{e}_{ij}(t)$),
- access to the underlying (hyper)graph $\mathcal{G}_t$,
- a well-defined update operator F implementing evidence-flow updates.

In short: a state, a mechanism, and an audit trail. Anything less is theatre.

6 Discussion

EQBSL is deliberately framed as a *systems-level* lift: it keeps EBSL’s evidence-centric foundation and extends the representation so trust becomes a first-class, vectorised state that composes with time, hyperedges, and machine learning pipelines. The exact choices of $\phi_{\pm}$, F , and Γ are domain-specific; the virtue is that these choices are exposed as parameters rather than buried as folklore. That makes the system falsifiable: you can perturb evidence, trace effects, and find the fault lines.

Acknowledgements

Subjective Logic is credited to Audun Jøsang [1]. Evidence-Based Subjective Logic and evidence-flow discounting are credited to Škorić and collaborators [2, 3]. EQBSL, as formulated here, is an original extension by Oliver C. Hirst.

References

- [1] Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016. DOI: 10.1007/978-3-319-42337-1.
- [2] Boris Škorić, Sebastiaan J. A. de Hoogh, and Nicola Zannone. Flow-based reputation with uncertainty: evidence-based subjective logic. *International Journal of Information Security*, 15:381–402, 2016. (Published online 20 Aug 2015.) DOI: 10.1007/s10207-015-0298-5.
- [3] Boris Škorić. Flow-based reputation with uncertainty: Evidence-Based Subjective Logic. arXiv:1402.3319, 2014 (revised 2015).