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 paper 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. A companion paper, *Proof-Carrying Trust: Zero-Knowledge Constraints for EQBSL*, describes how EQBSL updates can be made cryptographically verifiable without turning the system into a public confessional.

Keywords: trust networks; subjective logic; evidence flow; reputation systems; dynamic graphs; hypergraphs; zero-knowledge proofs.

1 Introduction

Trust, in computation, is often treated as though it were weather: a quantity one can simply "measure" and then operate on 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.

A companion paper, [4], builds on the framework here to show how trust updates can be enforced and audited using zero-knowledge proofs, so that systems are not only mathematically honest but cryptographically obliged to remain so.

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

$$b + d + u = 1. \tag{2}$$

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:

$$b = \frac{r}{r + s + K}, \qquad d = \frac{s}{r + s + K}, \qquad u = \frac{K}{r + s + K}. \tag{3}$$

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

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

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 baptised as "truth."

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

$$r_{ij}(t) = \phi_+\big(\boldsymbol{e}_{ij}(t)\big), \qquad s_{ij}(t) = \phi_-\big(\boldsymbol{e}_{ij}(t)\big), \tag{5}$$

where $\phi_+, \phi_- : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ are application-defined (often linear) maps. A common simple choice is a weighted projection:

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

with $\boldsymbol{w}^+, \boldsymbol{w}^- \in \mathbb{R}^m_{\geq 0}$. 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

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

implemented by:

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

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 (hyper)graph be

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

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

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

Let Δ_t denote the collection of new events observed between t and $t+1$. Define a global update operator

$$F : \; (\mathcal{E}_t, \Delta_t) \longmapsto \mathcal{E}_{t+1}, \tag{11}$$

so that in practice:

$$\mathcal{E}_{t+1} = F\big(\mathcal{E}_t, \Delta_t\big). \tag{12}$$

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:

$$\boldsymbol{e}_h(t), \qquad h \subseteq V, \; |h| \geq 2, \tag{13}$$

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

$$\boldsymbol{e}_{ij}(t) \;\mathrel{+}=\; \sum_{h\ni i,j} \alpha_{ijh}\, \Pi_{ij}\big(\boldsymbol{e}_h(t)\big), \tag{14}$$

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:

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

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 Proof-carrying EQBSL (bridge to ZK)

Having a principled trust operator is one thing; forcing systems to actually use it is another. The companion ZK paper [4] takes the state model just described and wraps it in cryptographic obligation.

5.1 State commitments

At time t, let the full EQBSL state be

$$S_t := \big(\mathcal{G}_t, \mathcal{E}_t\big), \tag{16}$$

with $\mathcal{G}_t = (V, E_t)$ and $\mathcal{E}_t = \{\boldsymbol{e}_{ij}(t)\}_{i,j}$. A proof-carrying instantiation publishes a binding commitment

$$C_t = \mathsf{Com}(S_t; r_t), \tag{17}$$

where Com is a standard vector commitment or polynomial commitment scheme, and r_t is the commitment randomness kept by the prover. The commitment C_t is the on-chain or globally visible handle for "the trust state at time t".

5.2 Circuit for the operator F

Define an arithmetic circuit (or R1CS / PLONKish constraint system)

$$C_F : \ (S_t, \Delta_t) \ \longmapsto \ S_{t+1} \tag{18}$$

implementing *exactly* the EQBSL update

$$S_{t+1} = \big(\mathcal{G}_{t+1}, \mathcal{E}_{t+1}\big), \qquad \mathcal{E}_{t+1} = F(\mathcal{E}_t, \Delta_t). \tag{19}$$

Inside the circuit, per-edge evidence vectors are updated, hyperedge contributions are allocated, and—if desired—aggregation functionals $\phi_\pm$ and the opinion mapping (3),(5) are re-evaluated as checks on consistency.

A zero-knowledge proof system (SNARK, STARK, or whatever the age demands) is then used to prove a statement of the form:

> "Given public inputs (C_t, C_{t+1}) and Δ_t, there exist openings S_t, S_{t+1} and randomness (r_t, r_{t+1}) such that $C_t = \mathsf{Com}(S_t; r_t)$, $C_{t+1} = \mathsf{Com}(S_{t+1}; r_{t+1})$, and $S_{t+1} = C_F(S_t, \Delta_t)$."

The verifier need never see the evidence vectors; they see only that the same operator F defined in this paper actually governed the update. The distance between spec and implementation collapses into a proof.

5.3 Circuit for embeddings Γ

If node-level embeddings $\boldsymbol{u}_i(t) = \Gamma(i, \mathcal{E}_t, \mathcal{G}_t)$ drive routing, gating, or ML features, one can define a second circuit

$$C_\Gamma : S_t \longmapsto U_t, \tag{20}$$

where $U_t = \{\boldsymbol{u}_i(t)\}_{i \in V}$, and produce a proof that the published embeddings are indeed the result of applying Γ to the committed state S_t.

The companion ZK paper decomposes these circuits into constraints that mirror the algebra here. The important point for EQBSL is modest: the same symbols $(\mathcal{E}_t, \mathcal{G}_t, F, \Gamma, K, \phi_\pm)$ that define the trust calculus also delimit what must be proven. There is no room for a "shadow" operator that the code secretly uses while the documentation describes another.

6 Interface assumptions

A practical EQBSL system provides:

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

A proof-carrying instantiation further provides:

- commitments C_t to the EQBSL state S_t,
- zero-knowledge proofs π_t that each transition $(C_t, \Delta_t) \mapsto C_{t+1}$ respects F,
- optional proofs that published embeddings U_t match Γ on the committed state.

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

7 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.

The proof-carrying layer sketched here and detailed in [4] does not change the algebra; it constrains the implementation. The system designer may still make poor choices, but they can no longer claim that the code does one thing while the evidence calculus says another. For a domain long dominated by oracular trust scores and unexplained bans, that is already a cultural revolution.

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, and its proof-carrying extension are original constructions by O. 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).

[4] O. C. Hirst. Proof-Carrying Trust: Zero-Knowledge Constraints for EQBSL. Manuscript, 2025.