ξPLN: A Correct and Complete Foundation for Probabilistic Logic Networks
`(DRAFT)`

Zar Codex CLI (OpenAI GPT-5.2)

February 4, 2026

Abstract

Probabilistic Logic Networks (PLN) couples logical structure with uncertain inference. In practice, however, many PLN presentations mix distinct layers: (i) a probabilistic semantic model, (ii) an evidence/weight plumbing model, and (iii) operational truth-value views (strength, confidence, interval bounds). This leads to "semantic drift": algebraic operators are treated as if they were the logic itself, and link-level inference attempts to propagate correlation-sensitive quantities without carrying the information required to do so correctly.

We present ξPLN, a complete PLN foundation designed to be (a) mathematically correct and (b) extensible to tractable sublayers. The core move is to make the complete layer a *posterior-state calculus*: the proof object is a revisable world-model posterior state, and all link/event truth values are obtained by querying that state. We give a reference complete instance as a Dirichlet posterior over complete worlds (`JointEvidence`), prove that evidence extraction commutes with revision, and formalize a no-go theorem showing that complete link evidence cannot, in general, be computed from local per-link evidence alone. We then explain how Bayesian-network-style sublayers arise as restricted but tractable world-model classes, keeping PLN's operational rules as compiled tactics with explicit assumptions.

1 Motivation and Context

PLN, as described by Goertzel, Iklé et al. [6], represents uncertain judgments using truth values such as *strength* (a probability-like estimate) and *confidence* (a reliability-like quantity). Modern *nuPLN* work (Nil Geisweiller, draft and v1) makes the key step explicit: a complete grounding requires a global probabilistic semantics—a world model with a joint distribution over possible worlds and a Bayesian update story. We treat νPLN as the probabilistic reference layer and extend it in ξPLN by (i) making posterior states the proof objects, (ii) separating evidence algebra from probability views, and (iii) connecting the complete layer to tractable Bayesian-network sublayers and Heyting-valued foundations [1].

Our experience formalizing PLN in Lean is that correctness failures almost always come from mixing layers:

- treating operational truth-value operators (interval bounds, confidence heuristics) as if they were the logic; or
- trying to compute complete posteriors for derived links from local link truth values without carrying the correlation information needed to do so.

ξPLN is a response: it isolates a *complete* core calculus and makes all fast PLN rules live as derived/compiled approximations relative to declared model classes.

2 Three Semantic Layers (and Why They Must Not Be Conflated)

We distinguish three layers.

2.1 Layer 1: World-model posterior states (complete semantics)

The complete layer carries a posterior state in some explicit class of world models. Conceptually: *a distribution or structured distribution* that can answer any query by conditioning/marginalization (or by exact inference within the chosen model class).

In the Lean prototype, this layer is represented by `JointEvidence`:

$$\texttt{JointEvidence}(n) := \texttt{Fin}(2^n) \to \mathbb{R}_{\geq 0}^{\infty}$$

interpreted as Dirichlet pseudo-counts over the 2^n complete worlds for n propositional atoms. Revision is pointwise addition of pseudo-counts.

2.2 Layer 2: Evidence algebra (quantale / Heyting structure)

PLN uses an evidence carrier `Evidence` as a compositional algebra of observations. In our formalization, `Evidence` is the 2D count object (n^+, n^-) with:

- parallel aggregation $\oplus$ ("revision"): add counts componentwise;
- order (information ordering): coordinatewise $\leq$;
- rich lattice structure (complete lattice; Heyting operators exist).

This layer is *not* itself the complete probabilistic semantics; it is an algebraic semantics for evidence aggregation and logical structure. It is particularly important for FO/HO extensions, where truth values are naturally Heyting-valued (cf. the SatisfyingSet construction in the codebase).

2.3 Layer 3: Operational views (strength, weight, confidence, bounds)

Strength/weight/confidence/intervals are *views* extracted from evidence. They are indispensable operationally, but they should not be used as if they were complete semantic state. ξPLN adopts the mantra:

> Probability is what you query; evidence is what you carry.

3 ξPLN Core: World-model Revision + Query

The complete core calculus is intentionally simple: it is the calculus of revising posterior states plus a query interface.

3.1 World-model interface

In Lean, we isolate a minimal interface `WorldModel` (posterior state + query projections):

```
inductive PLNQuery (Atom : Type*) where
  | prop : Atom -> PLNQuery Atom
  | link : Atom -> Atom -> PLNQuery Atom

class WorldModel (State : Type*) (Query : Type*) [EvidenceType State]
    where
  evidence : State -> Query -> Evidence
  evidence_add : forall W1 W2 q, evidence (W1 + W2) q = evidence W1 q +
       evidence W2 q
```

Here `EvidenceType` means: revision is an additive commutative monoid on `State`. Standard PLN event/link queries are represented by `PLNQuery Atom`; all truth-value quantities are derived by mapping extracted `Evidence` to strengths/weights. See `Mettapedia/Logic/PLNWorldModel.lean`.

3.2 Reference complete instance: Dirichlet over worlds

For finite propositional scope (atoms `Fin n`), we instantiate `WorldModel` with the Dirichlet world-table `JointEvidence`:

- `evidence(E, prop(A))` sums world counts where an atom is `true` vs `false`;
- `evidence(E, link(A,B))` sums world counts where `A` is true and `B` true vs false.

Crucially, extraction commutes with revision:

> revising joint evidence and then extracting a query is equal to extracting and revising at the query-evidence level.

Formally, the Lean prototype proves:

$$\text{evidence}(E_1 + E_2, q) = \text{evidence}(E_1, q) \oplus \text{evidence}(E_2, q)$$

for all queries q. See `Mettapedia/Logic/PLNJointEvidence.lean`.

3.3 Probability views

From `Evidence` we obtain posterior-mean probabilities (improper-prior strength):

$$P(A) = \frac{\#(A)}{\#(\top)}, \qquad P(B \mid A) = \frac{\#(A \wedge B)}{\#(A)}$$

where $\#(\cdot)$ is a world-count sum in `JointEvidence`. The Lean file `Mettapedia/Logic/PLNJointEvidenceProbability.lean` proves the exact ratio forms for these views.

4 A Sequent-Calculus View: World Models and Links

We present ξPLN as a sequent-style system with two contexts:

- an *evidence context* Γ, a finite multiset of posterior fragments (elements of a world-model state type `State`); and
- a *side-condition context* Σ containing structural assumptions about the chosen world-model class (e.g. a BN DAG, positivity, d-separation facts).

4.1 Judgments

We use two judgment forms.

1. **World-model judgment:** $\Sigma; \Gamma \vdash_{\text{wm}} W$ meaning "from the evidence pieces in Γ, we can construct the revised posterior state W".
2. **Query judgment:** $\Sigma; \Gamma \vdash q \Downarrow e$ meaning "querying the revised posterior state derived from Γ yields evidence e for query q".

In the Lean prototype, the world-model judgment is deterministic: if `State` is an additive commutative monoid, then the constructed posterior is simply the revision sum $W := \sum \Gamma$. The query judgment is then obtained by extraction via `WorldModel`:

$$e := \text{evidence}(W, q).$$

4.2 Core rules

Let `State` be an `EvidenceType` (revision $+$ with unit 0), and let `WorldModel` provide evidence $:$ `State` $\to$ `Query` $\to$ `Evidence`. The complete core calculus is:

$$\frac{}{\Sigma; \varnothing \vdash_{\text{wm}} 0}\ (\text{WM-Unit}) \qquad \frac{}{\Sigma; \{W\} \vdash_{\text{wm}} W}\ (\text{WM-Ev}) \qquad \frac{\Sigma; \Gamma \vdash_{\text{wm}} W \quad \Sigma; \Delta \vdash_{\text{wm}} W'}{\Sigma; \Gamma \uplus \Delta \vdash_{\text{wm}} (W + W')}\ (\text{WM-Rev})$$

$$\frac{\Sigma; \Gamma \vdash_{\text{wm}} W}{\Sigma; \Gamma \vdash q \Downarrow \text{evidence}(W, q)}\ (\text{Q-Extract})$$

The key algebraic law is commutation of extraction with revision:

$$\text{evidence}(W_1 + W_2, q) = \text{evidence}(W_1, q) \oplus \text{evidence}(W_2, q),$$

which makes the following *query-revision* rule admissible:

$$\frac{\Sigma; \Gamma \vdash q \Downarrow e \quad \Sigma; \Delta \vdash q \Downarrow e'}{\Sigma; \Gamma \uplus \Delta \vdash q \Downarrow (e \oplus e')}\ (\text{Q-Rev})$$

Links are queries. In this view, a PLN "link" $A \Rightarrow B$ is not an object-level connective; it is a query `link(A,B)` against the revised world model. The calculus does not propagate links directly; it revises world models and answers link queries.

4.3 Derived link calculus: classic PLN rules as admissible rewrites

Classic PLN rules (deduction/abduction/induction) become *compiled tactics* or *admissible query rewrites* relative to a world-model class with explicit side conditions Σ. Intuitively, they are ways to answer some link queries using other link/event queries, without pretending to compute complete link evidence in full generality.

Example: deduction strength admissibility (BN case). Let $\mathsf{strength}(W, q)$ denote the posterior-mean view extracted from evidence:

$$\mathsf{strength}(W, q) := \text{toStrength}(\text{evidence}(W, q)).$$

For a BN world-model class, if Σ entails the screening-off equalities needed by the general PLN deduction theorem (conditional independence of C from A given B and given $\neg B$, plus positivity), then one may rewrite a hard query $\mathtt{link}(A, C)$ into smaller ones:

$$\mathsf{strength}(W, \mathtt{link}(A, C)) = \text{plnDeductionStrength}\Big(\mathsf{strength}(W, \mathtt{link}(A, B)), \mathsf{strength}(W, \mathtt{link}(B, C)), \mathsf{strength}(W, \mathtt{p}$$

Lean status. `Mettapedia/Logic/PLNBayesNetFastRules.lean` proves this admissibility in the simplest nontrivial BN instance, the chain $A \to B \to C$: the theorem `chainBN_plnDeductionStrength_exact` shows the PLN deduction strength formula computes the exact $P(C \mid A)$ in the chain BN under explicit positivity side conditions.

Admissible link-rule schema (strength view). Let $\mathsf{strength}(e)$ denote the posterior-mean view extracted from evidence e. Then, for any world-model class where Σ entails the screening-off conditions, the following rule is admissible:

$$\frac{\Sigma \vdash \mathsf{ScreenOff}(A, B, C) \quad \Sigma; \Gamma \vdash \mathtt{link}(A, B) \Downarrow e_{AB} \quad \Sigma; \Gamma \vdash \mathtt{link}(B, C) \Downarrow e_{BC} \quad \Sigma; \Gamma \vdash \mathtt{prop}(B) \Downarrow e_B \quad \Sigma; \Gamma \vdash \mathtt{prop}(}{\Sigma; \Gamma \vdash \mathtt{link}(A, C) \Downarrow e_{AC}}$$

with the side-condition that

$$\mathsf{strength}(e_{AC}) = \mathrm{plnDeductionStrength}(\mathsf{strength}(e_{AB}), \mathsf{strength}(e_{BC}), \mathsf{strength}(e_B), \mathsf{strength}(e_C)).$$

This is a *strength-level* rewrite: the evidence e_{AC} itself is still obtained by querying the world model, unless an explicit evidence-flow law is available for the given model class.

5 A No-Go Theorem: "Complete" Link Inference Cannot Be Local

One might hope for a sequent-calculus-like system that takes local per-link evidence (`Evidence` for A, B, C, $A \Rightarrow B$, $B \Rightarrow C$) and computes complete evidence for $A \Rightarrow C$. ξPLN asserts that this is impossible in general without extra assumptions: correlations live in the joint state.

Theorem 1 (No local complete deduction rule)**.** *There is no function*

$$f : \mathtt{Evidence}^5 \to \mathtt{Evidence}$$

that, for all joint evidence states E, computes the exact link evidence for $A \Rightarrow C$ from only the local premises $\mathtt{Evidence}(A), \mathtt{Evidence}(B), \mathtt{Evidence}(C), \mathtt{Evidence}(A \Rightarrow B), \mathtt{Evidence}(B \Rightarrow C)$.

The Lean proof constructs two different joint evidence states on three atoms that agree on all these premises but disagree on the conclusion `linkEvidence(A,C)`. See `Mettapedia/Logic/PLNJointEvidenceNoGo.lean`.

Interpretation. This theorem is the formal core of "you must carry correlations". It does not make fast PLN useless; it tells us exactly what fast PLN *cannot* claim: completeness for arbitrary world models without structural assumptions.

6 Tractable Sublayers: Bayesian Networks as Restricted World Models

The complete world-table `JointEvidence` is exponential in n. To scale, we restrict the world-model class while preserving correctness *relative to that class.*

The next natural step is a Bayesian-network-style world model:

- a DAG structure specifying factorization;
- local conditional tables with Dirichlet evidence (counts) per CPT row;
- revision = add local Dirichlet counts (conjugate update);
- query = exact inference by variable elimination / junction tree, with complexity controlled by treewidth.

Lean status. We have implemented the *evidence plumbing* part of this sublayer:

- a Boolean BN CPT query type `CPTQuery` (node + parent configuration),
- a CPT posterior state `CPTState` storing `Evidence` per CPT entry, and
- an additive projection from `JointEvidence` to CPT evidence by marginalization (summing compatible worlds), which commutes with revision.

See `Mettapedia/Logic/PLNBayesNetWorldModel.lean`.

We have also begun the "fast rule exactness" bridge in the simplest nontrivial case: `Mettapedia/Logic/PLNBayesNetFastRules.lean` sets up the chain BN $A \to B \to C$, proves the required sink-factorization and normalization lemmas, derives the screening-off hypotheses, and applies `PLNDerivation.pln_deduction_from_total_probability` to obtain an *exactness theorem*: in the chain BN, the standard PLN deduction strength formula computes the correct conditional probability $P(C \mid A)$ (under the explicit positivity side-conditions).

Exact BN query answering (variable elimination / junction tree) is the next step: it will provide tractable evaluation of probabilities for larger query languages, while preserving correctness relative to the declared BN model class.

PLN spirit preserved. Classic PLN link rules (deduction/abduction/induction) become:

- compiled tactics that propose BN queries, or
- conditionally-sound lemmas under explicit assumptions (e.g. conditional independence),

rather than pretending to be globally complete link-level operators.

7 Factor Graphs: Semantic Factorization vs Operational Control

PLN has long used factor-graph and message-passing intuition for efficient inference. The recent MORK/MM2+ notes make this explicit by proposing a *Quantale-Annotated PLN Factor Graph* encoding in Atomspace (variables = formulas with truth values; factors = rule instances/potentials; messages combined with $\oplus$ and $\otimes$) [5, 4]. This is the right operational substrate, but ξPLN separates two distinct roles:

7.1 (A) Semantic factor graphs (world-model factorization)

Here factors *are* the model: the world-model posterior state is represented by a factorization of a joint distribution (BN or MRF). In this case:

- factors are genuine conditional / clique potentials;
- variable elimination / junction tree gives exact query answers (treewidth controls complexity);
- belief propagation is exact on trees and approximate on loopy graphs.

Thus a factor graph is a *world-model representation*, and correctness is relative to the declared model class.

Positive example (semantic). A BN with edges $A \to B \to C$ yields factors $P(A)$, $P(B \mid A)$, $P(C \mid B)$. If the model class is fixed to this DAG, variable elimination returns the exact $P(C \mid A)$ and the PLN deduction strength formula is admissible as a query rewrite.

7.2 (B) Operational factor graphs (inference control)

Here factors are *rule schemas* (deduction, induction, revision, etc.) whose local "potentials" are truth-value update formulas. This aligns with the PLN chainer and MORK PathMap indexing [5, 3]. However, unless each factor can be justified as a true conditional or likelihood factor in a world model, this structure should be viewed as *inference control*: it proposes which queries to ask and which rewrites to attempt, but it does not itself constitute the semantics.

Negative example (operational). A graph whose factors are "deduction" or "abduction" update formulas on STVs is useful for search, but it is not automatically a probabilistic model: unless each factor corresponds to a conditional/likelihood in some WM class, message passing is a heuristic scheduler rather than an exact semantics.

7.3 Alignment: when the two coincide

The two notions align when rule factors are *compiled* from a world-model class: each rule corresponds to a conditional or likelihood factor, and message passing implements the same queries that the world-model semantics would answer. This is exactly the intended "fast PLN rules as compiled tactics" story: the factor graph becomes a *query plan* whose correctness is discharged by explicit side conditions (Σ) such as d-separation or Markov properties.

Evidence accounting and control. Geodesic inference control suggests a decentralized evidence ledger to prevent double counting during message passing and chaining [2]. This is naturally compatible with the `WorldModel` interface: evidence is revised at the world-model layer, while factor-graph message passing is used to schedule or approximate queries, with provenance tracking guarding independence assumptions.

8 Knuth–Skilling and Heyting Foundations

Knuth–Skilling foundations of inference [7] provide axioms for valuation schemes on logical lattices, extending Cox-style plausibility calculi. In ξPLN, this plays two roles:

1. *Probability as a valuation/view.* At the world-model layer, probabilities arise as valuations on the Boolean algebra of subsets of worlds. This is the classical Kolmogorov story [8] instantiated for finite spaces, and it underwrites the ratio theorems proved in `PLNJointEvidenceProbability`.

2. *Evidence as Heyting-valued semantics.* At the evidence/algebraic layer, `Evidence` forms a rich (non-Boolean) lattice. K&S-style valuation rules persist but Boolean equalities can weaken to inequalities; see `Mettapedia/Logic/EvidenceIntuitionisticProbability.lean`.

An important cautionary fact—the "totality gate"—is already formalized: because `Evidence` has incomparable elements, it admits no faithful order-embedding into the reals. Thus one should not expect a single real-valued map to capture all evidence information. See `Mettapedia/Logic/PLN_KS_Bridge.lean`.

9 Roadmap and Deliverables

The core Lean artifacts supporting ξPLN are:

- `Mettapedia/Logic/PLNWorldModel.lean`: the world-model interface and derived views;

- `Mettapedia/Logic/PLNJointEvidence.lean`: the reference complete instance and revision-commutes-with-extraction;

- `Mettapedia/Logic/PLNBayesNetWorldModel.lean`: BN-style CPT evidence states and the additive projection from `JointEvidence`;

- `Mettapedia/Logic/PLNBayesNetFastRules.lean`: chain BN proof that the PLN deduction strength formula is exact in $A \to B \to C$;

- `Mettapedia/Logic/PLNJointEvidenceProbability.lean`: posterior-mean probability views (ratio theorems);

- `Mettapedia/Logic/PLNJointEvidenceNoGo.lean`: the local-completeness no-go theorem.

Next implementation milestones:

1. implement exact BN query answering (variable elimination first) for prop/link queries over the BN/factor-graph world-model layer;
2. generalize chain exactness to arbitrary BNs via d-separation/Markov properties so screening-off obligations are discharged structurally;
3. add a compilation layer: PLN rule patterns $\Rightarrow$ BN query plans with explicit Σ side-conditions.
4. extend query language beyond atoms to formulas/events, enabling FO/HO semantics via SatisfyingSets while keeping probability views as world-model queries.

10 Conclusion

ξPLN formalizes the slogan "pass distributions, not just truth values" in a way that is precise enough to support soundness/completeness claims and flexible enough to support tractable sublayers. The complete layer is not a competing "second PLN"; it is the reference semantics that operational PLN rules approximate, under explicit assumptions, within declared model classes.

References

[1] Nil Geisweiller and Claude. Towards a complete formalization of pln. Internal draft (nuPLN v1), 2026. Local file: `/home/zar/claude/lean-projects/mettapedia/papers/nuPLN_v1.pdf`.

[2] Ben Goertzel. Geodesic inference control. Internal draft, 2025. Local file: `/home/zar/claude/literature/Geodesic-inference-control.pdf`.

[3] Ben Goertzel. Mork-miner. Internal technical note, 2025. Local file: `/home/zar/claude/literature/MORK-Miner.pdf`.

[4] Ben Goertzel. Mork-moses implementation ideas. Internal technical note, 2025. Local file: `/home/zar/claude/literature/MOSES-MORK.pdf`.

[5] Ben Goertzel. Notes on possible implementation of pln factor graphs in mm2+/mork. Internal technical note, 2025. Local file: `/home/zar/claude/literature/PLN_FactorGraph_MORK_v2.pdf`.

[6] Ben Goertzel, Matthew Iklé, Izabela Freire Goertzel, and Ari Heljakka. *Probabilistic Logic Networks: A Comprehensive Framework for Uncertain Inference*. Springer, New York, 2009.

[7] Kevin H. Knuth and John Skilling. Foundations of inference. *Axioms*, 1(1):38–73, 2012. arXiv:1008.4831.

[8] Andrey Nikolaevich Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, Berlin, 1933. English translation: Foundations of the Theory of Probability, Chelsea, 1956.