The Faithfulness Gap

Certifying Semantic Equivalence Between Natural-Language and Formal Mathematical Statements

Noor Islam S. Mohammad[1] Tamim Sheikh[2]

[1]Istanbul Technical University [2]Jashore University of Science and Technology

The 6th Muslims in ML (MusIML) Workshop · ICML 2026

Seoul, South Korea

islam23@itu.edu.tr · pmlrbd.github.io/BPF

Outline

Three plausible Lean 4 outputs for the same sentence:

```
-- (A) Faithful
theorem evt_A {K : Set R}
    (hK : IsCompact K) (hne: K.Nonempty)
    {f : R -> R} (hf: ContinuousOn f K) :
    exists x in K, forall y in K, f y <= f x

-- (B) Hypothesis omission
theorem evt_B {K : Set R}
    (hK : IsCompact K)    -- missing: Nonempty
    {f : R -> R} (hf: ContinuousOn f K) :
    exists x in K, forall y in K, f y <= f x

-- (C) Quantifier inversion (trivially true)
theorem evt_C {K : Set R}
    (hK : IsCompact K) (hne: K.Nonempty)
    {f : R -> R} (hf: ContinuousOn f K) :
    forall y in K, exists x in K, f y <= f x
```

All three typecheck ✓

Only **(A)** is faithful.

- **(B)** — false on empty set
- **(C)** — trivially true ($x = y$), *not* the Extreme Value Theorem

The Faithfulness Gap

A formalization can be **well-typed, provable, and wrong**. Current pipelines accept (A), (B), and (C) equally.

Scale of the problem

19% of outputs from a SOTA autoformalizer on graduate analysis exhibit semantic drift.

$\Delta_\forall$ — Quantifier Inversion

Swaps $\forall x\, \exists y$ with $\exists y\, \forall x$.
Drifted F often provable yet *disjoint* from N.

Δ_C — Conclusion Generalisation

$F \Rightarrow N$ but $N \not\Rightarrow F$.
Locally plausible; globally unsound.

Δ_H — Hypothesis Omission

A premise of N is absent from F.
Formalization is *strictly stronger* than intended.

Δ_T — Type Coercion

Silent retagging, e.g. $\mathbb{R} \to \mathbb{Q}$ or total $\to$ partial.
Hardest to detect syntactically.

These four classes account for **94%** of human-labelled drift instances in a pilot study of 400 autoformalizer outputs.

Provability Fingerprint (Def. 4.1)

Let $F \in \mathcal{F}$, probe set $\mathcal{P} \subseteq \mathcal{F}$.

$$\Phi_F^+(\mathcal{P}) = \{\, P \in \mathcal{P} : \mathcal{T} \vdash F \to P \,\}$$

$$\Phi_F^-(\mathcal{P}) = \{\, P \in \mathcal{P} : \mathcal{T} \vdash P \to F \,\}$$

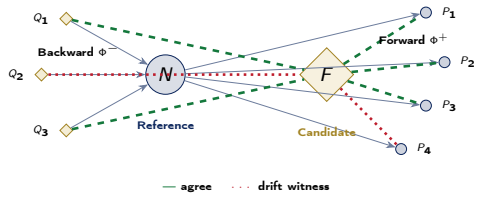
Full fingerprint: $\Phi_F = (\Phi_F^+, \Phi_F^-)$

Why Bidirectional?

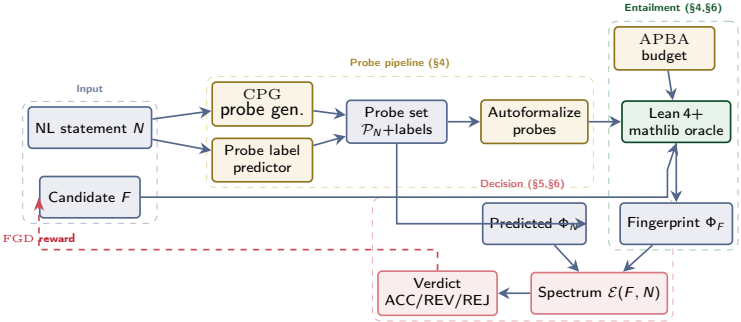
- **Forward-only**: misses strict *strengthenings*
- **Backward-only**: misses strict *weakenings*
- Bidirectional = *minimum sufficient signal*

Proposition 4.2

$\mathcal{T} \vdash F \leftrightarrow N \iff \Phi_F(\mathcal{P}) = \Phi_N(\mathcal{P})$
(for $\mathcal{P}$ closed under $\mathcal{T}$-equivalence).



Φ^+: forward probes P_i (what F implies); Φ^-: backward probes Q_j (what implies F). Disagreements are drift witnesses.

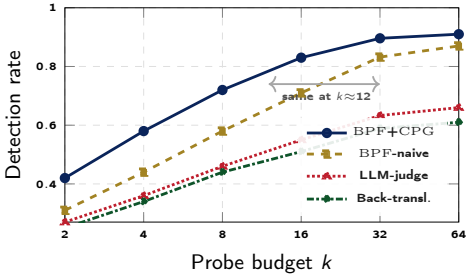

Entailment (§4,§6)
APBA budget
Probe pipeline (§4)
Input
CPG probe gen.
Probe set $\mathcal{P}_N$+labels
Autoformalize probes
Lean 4+ mathlib oracle
NL statement N
Probe label predictor
Decision (§5,§6)
Candidate F
Predicted Φ_N
Fingerprint Φ_F
FGD reward
Verdict ACC/REV/REJ
Spectrum $\mathcal{E}(F, N)$

Key insight: generic probes waste budget. Target *specific* drift classes.

For each drift class $D \in \{\Delta_\forall, \Delta_H, \Delta_C, \Delta_T\}$**:**

1. Mechanically perturb N to obtain drifted twin N^D
2. Ask an LLM to synthesise probes that distinguish N from N^D:

$$\mathcal{Q}_D(P) \propto \Pr[P \text{ from } N] \cdot \Pr[P \text{ not from } N^D]$$



Critical design choice

N^D is generated **mechanically**, not by an LLM — guarantees unambiguous drift-class labels.

CPG reaches BPF-naive's $k = 32$ rate at $k \approx 12$.

Definition 6.1

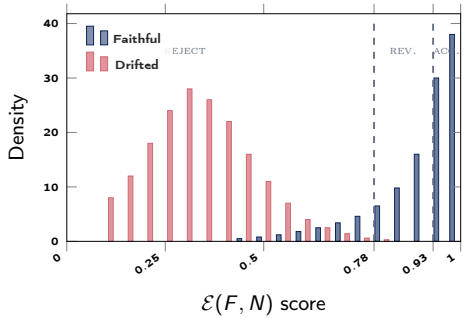
$$\mathcal{E}(F, N) = 1 - \frac{\sum_{P \in \mathcal{P}} w(D_P) \cdot 1[\text{cell}_P(F) \neq \ell(P)]}{\sum_{P \in \mathcal{P}} w(D_P)} \in [0, 1]$$

Three calibrated decision regions:

- **ACCEPT**: $\mathcal{E} \geq 0.93$
- **REVIEW**: $0.78 \leq \mathcal{E} < 0.93$ ($\rightarrow$ human triage)
- **REJECT**: $\mathcal{E} < 0.78$

Downstream uses beyond binary verdict:

- Ranking multiple autoformalizer candidates
- Triage of REVIEW region to expert
- Continuous reward for GRPO training



Theorem 8.1—Drift Detection

Let $\alpha(D)$ = witnessability rate for drift class D. With budget

$$k \geq \frac{1}{\alpha(D)} \log \frac{1}{\delta},$$

BPF flags, drift with probability $\geq 1 - \delta$.

Empirical witnessability rates:

Class	$\alpha(D)$	k for $\delta = 0.01$
$\Delta_\forall$	0.62	7
Δ_H	0.55	9
Δ_C	0.48	10
Δ_T	0.31	15

The budget $k = 32$ gives $\delta \leq 0.01$ for the first three classes.

Theorem 8.2—PAC-Faithfulness

With $k \geq \frac{1}{\alpha_0} \log \frac{1}{\delta\varepsilon}$ probes, BPF produces $\hat{H} \subseteq \mathcal{F}$ such that

$$\Pr_{F \sim \mathcal{D}_\mu} \left[F \in \hat{H} \iff F \in [N]_\mathcal{T} \right] \geq 1 - \varepsilon$$

with probability $\geq 1 - \delta$.

Theorem 7.1—APBA Budget Reduction

APBA reaches detection rate ρ with budget

$$k_{\text{APBA}} \leq k^\star / \gamma$$

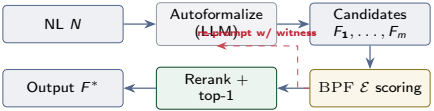
where it is $\gamma \approx 3.2$ on DriftBench.

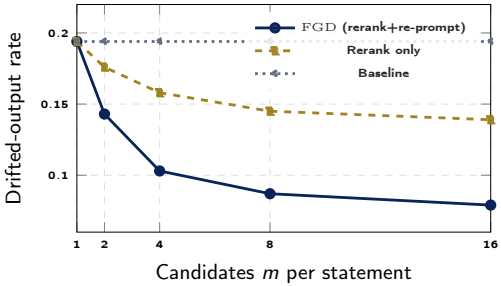
Limitation: undetectable drift

Convention drift (e.g., `Group` vs `Monoid`) is invisible to provability probes. Handled by a complementary structural test.

Algorithm:

1. Autoformalizer samples m candidates $F_1, \ldots, F_m$
2. BPF scores each ($k = 8$ probes per class)
3. Return top-ranked F^* by $\mathcal{E}$
4. **Re-prompt**: if all $\mathcal{E} < 0.78$, feed the highest-scoring witness probe as an in-context repair hint





FGD reduces drift rate **19.4% → 10.3%** at $m = 4$ (**47% relative reduction**).

Re-prompting dominates at small m; pure reranking saturates at ∼14%.

Dataset composition:

Subfield	Faith.	$\Delta_\forall$	Δ_H	Δ_C	Δ_T	Comb.
Analysis	198	64	71	58	33	31
Algebra	173	51	68	49	28	27
Topology	162	49	59	47	30	25
Number Th.	144	42	55	41	22	22
Combinator.	138	43	51	39	24	21
Category	117	35	44	33	19	19
Total	932	284	348	267	156	145

Construction:

- **Deterministic** perturbation rules → unambiguous labels
- *Wild split*: 384 pairs from real LLM outputs, annotated by two experts ($\kappa = 0.81$)
- Each entry: NL statement, candidate F, drift label, gold formalization

Why existing benchmarks miss this:

- MiniF2F, ProofNet, LeanDojo — measure *proving* capability **conditional** on the statement being correct
- The upstream faithfulness error is *invisible* to them
- DriftBench is the first to evaluate the autoformalization-faithfulness loop end-to-end

Adapts to cross-system use

BPF is system-agnostic (needs only an entailment oracle). Preliminary Isabelle/HOL results: F1 = 0.88 vs. 0.91 on Lean 4.

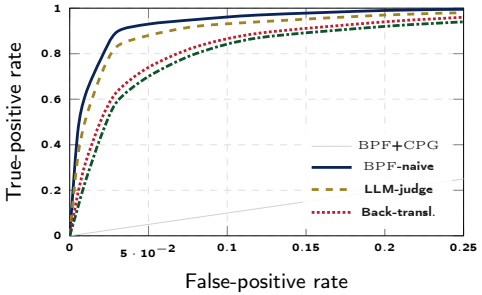
Release

`pmlrbd.github.io/BPF`
Code + benchmark + prompts

Main Results

Drift detection on the controlled DRIFTBENCH split

Method	F1	Det@3%	FPR	Cost
Typecheck only	0.27	0.114	0.020	1×
Provability	0.42	0.412	0.030	18×
BLEU-ref	0.55	0.301	0.030	1×
Back-translation	0.66	0.589	0.030	4×
LLM-judge	0.71	0.633	0.030	2×
BPF-naive	0.85	0.832	0.031	22×
BPF+CPG	**0.91**	**0.896**	**0.030**	26×
+APBA	0.91	0.894	0.031	8×

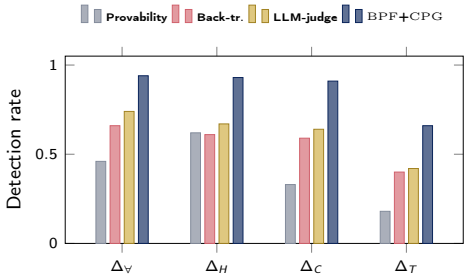


Key takeaways:

- BPF+CPG reduces LLM-judge residual error by **72%**
- APBA preserves F1 at 3.2× **lower cost**
- AUC = 0.962 vs. 0.938 (BPF-naive), 0.879 (LLM-judge)

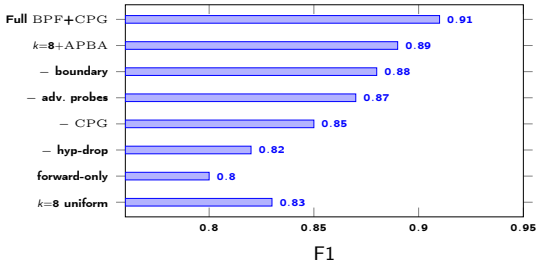
Per-Class Detection & Ablations

Detection rate by drift class (3% FPR):



Δ_T hardest ($\alpha = 0.31$); CPG closes much of the gap.

Ablation (F1 on controlled split):



CPG (+6 pts) and bidirectional (+11 pts) are the two largest contributors.

Summary & Contributions

The faithfulness gap is the bottleneck for trustworthy autoformalization.

Our six contributions:

1. **Formalise** faithfulness as a relation between interpretation distributions
2. **BPF** — probe-based, reference-free certifier with bidirectional consequence-neighbourhood matching
3. **CPG** — contrastive probe synthesis targeting each of the four canonical drift classes
4. **Equivalence Spectrum** — continuous score + **APBA** — 3.2× budget reduction
5. **Theoretical bounds** — Drift Detection Theorem + PAC-Faithfulness Theorem
6. **FGD** — 47% drift reduction at generation time

Key numbers

- BPF+CPG: F1 = 0.91, 89.6% detection @ 3% FPR
- APBA: 3.2× fewer oracle calls
- FGD: drift rate 19.4% → 10.3%
- DriftBench: 2,183 pairs, $\kappa = 0.81$

Limitations

- **Convention drift** undetectable by BPF (structural test is complementary)
- Entailment oracle compute-intensive (APBA mitigates)
- Probe-label noise bounded by PAC theorem

Code & Benchmark:

`pmlrbd.github.io/BPF`

Thank you!

Questions?

Paper

pmlrbd.github.io/BPF
The 6th Muslims in ML (MusIML) Workshop @ ICML 2026

Contact

Noor Islam S. Mohammad
islam23@itu.edu.tr

Istanbul Technical University | Jashore University of Science and Technology