AIPV 2026 — Workshop Brainstorming Summary

AI, Proof and Verification @ FM 2026

A synthesis of workshop discussions and team presentations

Why Formal Methods × AI?

Three motivations emerged consistently across the workshop:

- **Safety-critical systems** demand correctness guarantees that informal testing alone cannot provide
- **Scientific and mathematical domains** face ambiguous or underspecified requirements — e.g. formalised mathematics in Lean
- **AI systems that auto-repair programs** risk propagating misunderstandings when no formal specification exists to check against

The Gap Between ML and Formal Methods

A central tension the workshop kept returning to:

- Formal properties must be **relevant**, **correct**, and **distinguishing** — vacuous or tautological specs are not useful
- A single program can admit **multiple valid interpretations**; formalisation forces a choice among them
- The gap between **machine learning** (learning from data) and **formal logic** (reasoning from axioms) remains wide
- Different proving communities — Mathlib, Mizar, Rocq — disagree on what counts as an **obvious step**

Theme 1 — AI for Formal Methods

Using AI to reduce the burden of constructing and checking formal arguments

- Use AI to **remove redundant constraints** from verification conditions automatically
- Use a reinforcement-learning agent to identify the **root cause of a classifier's decision** and target it
- **Semantic perturbation**: apply a high perturbation budget to unimportant input features and a low budget to important ones

Theme 2 — FM for AI Systems

Using formal methods to give guarantees about AI and LLM behaviour

- **Generating formal specs from natural language**: translate informal property descriptions into machine-checkable specifications
- **Refinement proofs assisted by LLMs**: use LLMs to construct or repair program refinements, then verify them formally
- **Validating that a spec captures intent**: a human-in-the-loop methodology to confirm the formal spec matches what was meant
- **LLMs for program repair**: raise a quality gap — generated repairs currently lack *understandability*

Theme 3 — Making Formal Methods Accessible

Formal guarantees should be a domain conversation, not a programming task

- An LLM can mediate between an **informal domain document** and a hidden theorem prover
- The domain expert sees only: **clarifying questions**, a **summary of assumptions**, and a **plain-language explanation** of the result
- The expert never needs to write proof code or interact with the prover directly
- Formalization, proof search, and traceability remain behind the curtain

Open Research Questions

Problems the community identified as unsolved:

1. **Compressing AI-generated proofs** — current proofs are large and hard to inspect or trust
2. **Semantic equivalence** — when do two proofs mean the same thing?
3. **Human vs. LLM proof quality** — how does LLM output on Mathlib pull requests compare to human contributions?
4. **Notions of obviousness** — what counts as a trivial step differs across Mathlib, Mizar, Rocq, and other communities
5. **Synthetic training data** for theorem-proving AI, grounded in real Lean proofs

Access and Equity

Who currently benefits from LLM-assisted formal methods?

- State-of-the-art LLM-assisted proving relies heavily on **closed-source, proprietary models**
- Access is concentrated among a small group of well-resourced institutions
- Formal verification tools that require deep expertise to operate **exclude the very domain experts who most need verified answers**
- Any credible solution must prioritise **understandability**, not just formal correctness

Directions for Action

Three concrete proposals from the workshop:

1. **Unified curriculum for Formal Methods** — expand and modernise university FM courses to reflect current AI integration
2. **Standard Protocol Language** — a shared specification language for the formal verification of LLMs and neural networks
3. **AI-integrated verification tools** — tools that produce **human-understandable annotations and explanations**, not just proof certificates

Closing

The recurring insight across all discussions:

Formal methods and AI are not in competition — but bridging them requires solving hard problems on both sides: specification, access, understandability, and trust.

AIPV 2026 — AI, Proof and Verification Workshop @ FM 2026