# The 4/δ Bound

Designing Predictable LLM-Verifier Systems for Formal Method Guarantee
https://arxiv.org/abs/2512.02080

# Probabilistic AI & Formal Verification: Bridging the Gap

We’ve released a technical report introducing a mathematical foundation with provable guarantees for LLM-based formal verification workflows.

**The Problem:** LLMs show great promise for automating tasks like:

- invariant generation
- proof synthesis
- counterexample-guided repair

# Probabilistic AI & Formal Verification: Bridging the Gap

But today’s approaches lack theoretical guarantees. Without convergence bounds, LLM-driven loops may:

- run indefinitely
- waste compute
- require ad-hoc timeouts
- behave unpredictably in CI/CD pipelines

For safety-critical verification, this is unacceptable.

# Probabilistic AI & Formal Verification: Bridging the Gap

**Our Contribution**

We present the **LLM-Verifier Convergence Theorem**. To our knowledge, the first formal convergence guarantee for iterative LLM-verification pipelines.

For sequential loops of the form: **propose → verify → repair → retry**

We prove that the system:

- terminates with high probability for any $\delta > 0$
- converges in expected time $E[n] \leq 4/\delta$
- satisfies exponential tail bounds

# Probabilistic AI & Formal Verification: Bridging the Gap

Here, δ represents the probability that each LLM step makes useful progress, directly linking model quality to runtime guarantees.

**Empirical Validation**

- Across 90,000+ trials, observed behavior closely matches theory, showing the bounds are not just asymptotic, they’re practical.

# Probabilistic AI & Formal Verification: Bridging the Gap

**Why This Matters for Industry**

This turns LLM-assisted verification from guesswork into engineering:

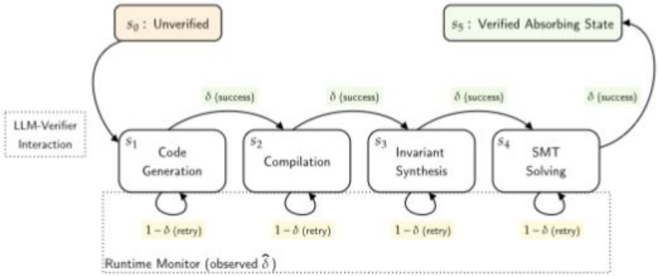
- predictable resource planning for CI/CD
- principled timeout configuration
- performance budgeting based on model capability ($\delta$)
- clear deployment regimes: marginal, practical, and high-performance

# Theoretical Background (Absorbing Markov Chain)

The **Theorem** is based on the mathematics of **Absorbing Markov Chains** and the **Geometric Distribution**.

The LLM-Verifier system is a five-state absorbing Markov Chain:

- Transient States (T): Four verification pipeline stages: s1 (CodeGen), s2 (Compilation), s3 (InvariantSynth), and s4 (SMTSolving).
- Absorbing State (A): One final, irreversible state, s5 (Verified).



# Theoretical Background (Absorbing Markov Chain)

The Markovian Property holds because the probability of generating a correct verification artifact (the transition probability) depends only on the current state and is independent of the history of previous attempts (memoryless).

# Theoretical Background (Absorbing Markov Chain)

The Markovian Property holds because the probability of generating a correct verification artifact (the transition probability) depends only on the current state and is independent of the history of previous attempts (memoryless).

### Requirements for a Geometric Distribution

1. **Binary Outcomes:** Each trial has only two outcomes: success or failure.
2. **Independence:** Each trial is independent of the others.
3. **Constant Probability:** The probability of success ($p$) remains the same for every trial.
4. **Goal:** The experiment stops after the *first* success is achieved.

# Theoretical Background (Absorbing Markov Chain)

**Convergence Time Derivation (The 4/δ Bound)**

The time spent in each transient state, $M_j$ (the residence time), is modeled by a Geometric random variable.

The **Expected Single-State Time** (time to exit state j) is the expected value of the Geometric distribution: $E[M_j]=1/\delta$.

The Total **Expected Convergence Time** is the sum of the independent residence times spent in the four sequential transient states:

E[n]=∑j(1^4)E[Mj]=1/δ+1/δ+1/δ+1/δ = 4/δ

# Theoretical Background (Absorbing Markov Chain)

This theoretical foundation leads to the main result, providing formal guarantees for the system :

**Almost Sure Convergence**: The system reaches the Verified state ($\tau<\infty$) with probability 1, provided $\delta>0$.

**Expected Iteration Bound**: The mean time to convergence is $E[\tau]=\delta/4$.

**Tail Bound**: The probability of excessively long verification runs decays exponentially: $P(\tau>k)\leq\alpha(1-\delta)$.

# Threats to Validity

1. **Assumption of Parameter Stationarity (internal)**

The most critical limitation is the assumption that the LLM's capability (δ) remains constant throughout the entire refinement process.

2. **State Space Limitation (internal)**

The model uses a simplified 5-state structure to capture the engineering pipeline.

# Threats to Validity

**3. Lack of Correlation Between Stages (Advanced Modeling Gap)**

The model assumes independence between the transient states, allowing the total time T to be a simple sum of the times spent in each state ($\tau = \sum Mj$).

**4. Generalizability to Real-World Systems (External)**

The core validation relies on a simulation framework rather than real-world LLM deployments.

# Threats to Validity

**5. Focus on Iteration Count Over Time (Construct)**

The primary metric of convergence is the iteration count (E[n]), not the actual time spent.

**6. Binary Success Definition (Construct Validity Threat)**

The model uses a simple binary definition for progress.

**Theorem 1** (LLM-Verifier Convergence Theorem) *We model the LLM-verifier process using a discrete-time Markov Chain, denoted as $X = \{X_n\}_{n\geq 0}$. The state space $S$ consists of a sequential engineering pipeline ($S = T \cup A$). First, $T = \{s_1, s_2, s_3, s_4\}$ represents the set of transient pipeline stages (* `CodeGen` *,* `Compilation` *,* `InvariantSynth` *,* `SMTSolving` *). Second, $A = \{s_5\}$ acts as the single absorbing state (* `Verified` *). Assuming a fixed success probability $\delta \in (0, 1]$ for passing any single stage, we construct the transition matrix $P = (P_{i,j})$. The individual entries $P_{i,j} = \mathbb{P}(X_{n+1} = s_j \mid X_n = s_i)$ are arranged as follows:*

*(i)* ***Transient Pipeline States****: For the transient states $i \in \{1, 2, 3, 4\}$, the transition probabilities are:*

$$P_{i,j} = \begin{cases} \delta & \text{if } j = i + 1 \quad \text{(success: advance to next stage)}, \\ 1 - \delta & \text{if } j = i \quad \text{(failure: retry current stage)}, \\ 0 & \text{otherwise.} \end{cases}$$

*(ii)* ***Absorbing State****: State $s_5$ is absorbing, meaning the verification is complete. Hence, $P_{5,5} = 1$.*

*Let $\tau = \inf\{n \geq 0 : X_n \in A\}$ be the iteration count until verification. The following bounds hold:*

1. ***Almost Sure Convergence****:*

$$\mathbb{P}(\tau < \infty \mid X_0 \in T) = 1$$

2. ***Expected Iteration Bound****: The mean time to convergence is given by:*

$$\mathbb{E}[\tau \mid X_0 = s_1] = \frac{4}{\delta}$$

3. ***Tail Bound****: Consider constants $\alpha > 0$ and $\lambda_Q \in (0, 1)$, where $\lambda_Q$ is the spectral radius of the transient submatrix $Q$. Then, for all $k \geq 0$:*

$$\mathbb{P}(\tau > k \mid X_0 = s_1) \leq \alpha \lambda_Q^k.$$

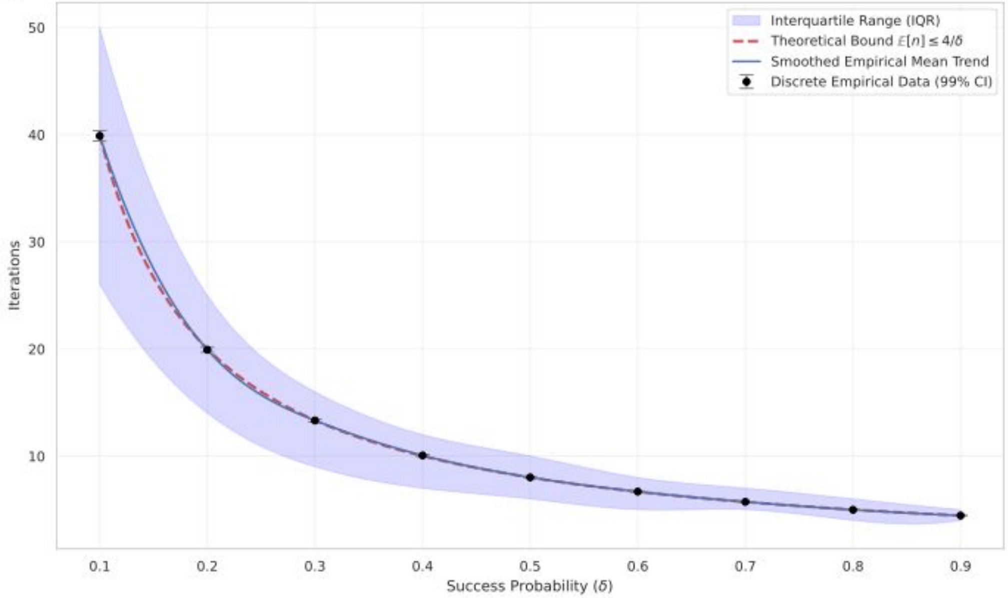
*Substituting $\lambda_Q = 1 - \delta$ results in the following exponential bound:*

$$\mathbb{P}(\tau > k \mid X_0 = s_1) \leq \alpha (1 - \delta)^k.$$

*Proof* We analyze the absorbing Markov Chain structure to establish the three guarantees.

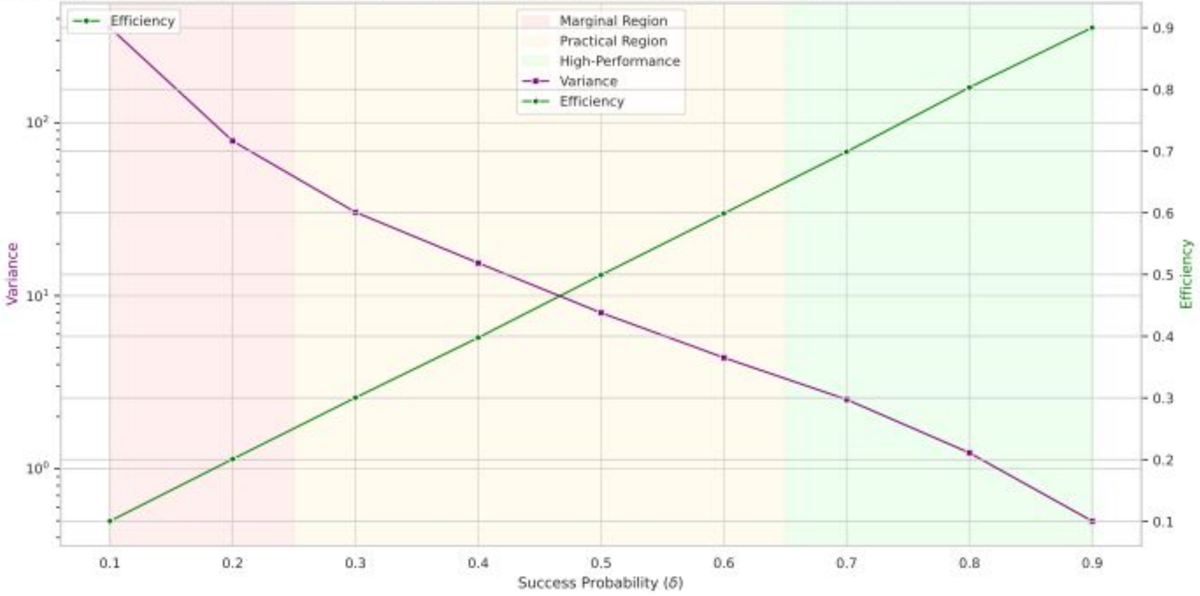
# Empirical Validation

**Fig. 6 Theoretical Bound Alignment and Empirical Convergence Rate**. The figure compares the theoretical expected bound ($\mathbb{E}[n] \leq 4/\delta$) against the empirical mean ($\mu$) across the $\delta$ spectrum. The extremely close tracking between the empirical curve (blue) and the theoretical curve (red dashed) demonstrates the tight alignment ($C_f \approx 1.0$) of the model prediction. The blue shaded area represents the IQR ($P_{25} - P_{75}$), which visually captures the dramatic decrease in system variance ($\sigma^2$) when transitioning from the marginal region ($\delta < 0.3$) to the high-performance region ($\delta > 0.6$)



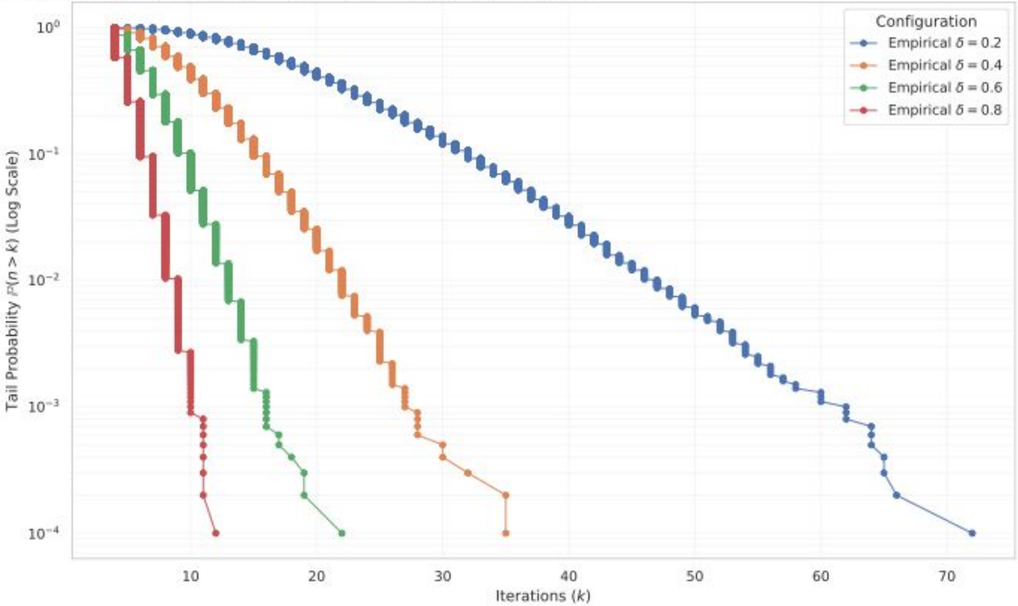
# Empirical Validation

**Fig. 7 Operational Regions Map: Performance and Stability Analysis**. This dual-axis visualization maps the framework's behavior across the $\delta$ spectrum. The left axis (purple, log scale) tracks the empirical variance ($\sigma^2$), demonstrating the system's predictability. The right axis (green, linear scale) tracks iteration efficiency ($\eta$). The plot clearly illustrates the sharp phase transition in stability: variance collapses rapidly upon entering the practical region ($\delta \geq 0.3$), confirming the empirical boundaries for safe and efficient deployment



# Empirical Validation

**Fig. 8 Tail Probability Analysis: Empirical Validation of Exponential Decay**. The figure plots the Complementary Cumulative Distribution Function (CCDF, $P(n > k)$) on a log-linear scale. The approximately linear decay observed across all tested $\delta$ values confirms the fundamental assumption that the convergence time follows an exponential tail behavior ($\mathbb{P}(n > k) \propto e^{-c\delta k}$). This validation ensures that the system's worst-case convergence time is reliably bounded and predictable, even in the marginal region



# Adaveri

# Adaveri Architecture

**AdaVeri** is an adaptive, AI-driven framework and real-time control system that enhances the speed, reliability, and safety of formal software verification, especially in safety-critical applications.

It monitors the verification process, adjusting strategies based on success rates, with a mathematical guarantee of convergence for efficient and reliable verification.

# Adaveri Architecture