What is AI-Minesweeper?

AI-Minesweeper is a hypothesis discovery framework inspired by the classic game Minesweeper. It uses epistemic reasoning to uncover hidden patterns, contradictions, and gaps in structured datasets. The framework is designed to be domain-agnostic, allowing researchers to plug in custom datasets and adjacency logic.

Plug-in Design

The framework supports modular plug-ins for domain-specific logic. Each plug-in defines adjacency rules, clue generation, and hypothesis evaluation tailored to the dataset.

Demo List

- **TORUS Demo**: Explore topological patterns.
- **Cymatics Demo**: Analyze wave interference data.
- **Prime-Spiral Demo**: Investigate prime number distributions.
- **Periodic Table Demo**: Discover gaps and anomalies in the periodic table.

TORUS Integration

TORUS is a complementary tool for topological data analysis. AI-Minesweeper integrates with TORUS to enhance hypothesis discovery in spatial datasets.

Case Study 2 – Hunting Super-Heavy Elements

The periodic-table domain adapter demonstrates hypothesis discovery in nuclear physics. Weighted clues guide the search for unbound isotopes, treating unstable nuclei as mines. This approach highlights gaps in nuclear stability metrics, enabling predictions for super-heavy elements like Z 119–126.

Torus of Tori and DPP-14

This note summarizes the Torus-of-Tori intuition used by the χ-recursive solver and outlines the DPP-14 (Deep Parallel Processing, 14 lanes) protocol referenced across the framework. It's intentionally concise and operational, matching the implementation.

χ-cycle and controller dimension

- χ-cycle: a bounded, cyclical modulation applied to solver confidence and risk thresholds. Every reveal/flag ticks a counter; UI charts show this as a smooth oscillator for intuition and debugging. The solver remains deterministic: no jitter, canonical tie-breaks, and normalization over hidden cells only.
- Controller dimension: the degrees of freedom the solver can adjust (risk tolerance, constraint ordering, lane interactions). These are tracked, not randomized, to ensure reproducibility in CI and tests.

Torus-of-Tori intuition

Think of each recursion lane as a small torus evolving over time. Cross-links between lanes form a larger torus-of-tori where information propagates without peeking at unrevealed answers. Practically, this translates to: - Local constraints first (classic Minesweeper logic), -

Deterministic risk scoring for ambiguous regions, - Confidence-gated actions that modulate when flags are “safe” vs “certain.”

DPP-14 protocol (what it means in code)

1. 14 parallel reasoning lanes. Each lane carries a consistent state view; no lane sees forbidden information.
2. Cross-lane propagation occurs only through admissible, derived facts (no peeking). If a lane diverges, it’s pruned deterministically.
3. Risk and confidence are aggregated with stable ordering. When ties occur, coordinates are ordered canonically.
4. Every mutation (reveal/flag/unflag) advances the χ-cycle counter; visualizations reflect the phase but never influence tie-breaks.

Determinism and testing

- Tests set AIMS_TEST_MODE=1 to allow deterministic shortcuts where appropriate. Production code remains pure and identical across runs.
- CI verifies: lint, security (non-blocking), and full test suite. A smoke script demonstrates a short, reproducible run with stable move ordering.

Minimal deterministic example

Given a 3×3 board with a single revealed clue “1” at center, all lanes agree on the admissible set of hidden neighbors. With AIMS_SEED fixed, risk sorting produces the same next probe across runs:

1. Hidden set H = {(0,0)…(2,2)} − {(1,1)}
2. Risk map r: H → [0,1] normalized over |H|
3. Next move = argmin_{(r,c)∈H} (r[(r,c)], r, c)

Tie breaks are resolved by (row, col). No lane accesses unrevealed truths (no-peeking).

Extending the model

- Additional lanes: keep cross-lane communication admissible and auditably deterministic.
- New risk terms: ensure they normalize over hidden cells and use stable sorts.
- UI: you may visualize χ-phase and lane status, but do not couple visuals to decision ordering.

For a narrative motivation and broader context, see Why TORUS Matters and the χ-brot demo. These show how cyclical modulation can make complex discovery tasks legible without sacrificing reproducibility.

Prime Spirals Validation

Prime residue distributions exhibit recursive χ-cycle patterns, matching theoretical predictions.

Key Findings

- Mod-14 prime spirals demonstrate χ-cycle recursion.
- Validation performed using computational simulations.

For detailed results, refer to the validation report.

Confidence Oscillation

Confidence oscillation relates BetaConfidence dynamics to the χ-cycle and the controller dimension. Oscillations around χ=1/√2 reflect calibration dynamics.

Overview

The BetaConfidence component tracks solver calibration, ensuring dynamic risk modulation. Confidence oscillations align with the χ-cycle, demonstrating periodic adjustments in solver behavior.

Visualization

Confidence oscillations are visualized in the Streamlit app, providing real-time feedback on solver calibration.

The chart below demonstrates the τ effect, where τ represents the decay constant of the oscillation, indicating how quickly the confidence stabilizes over time.

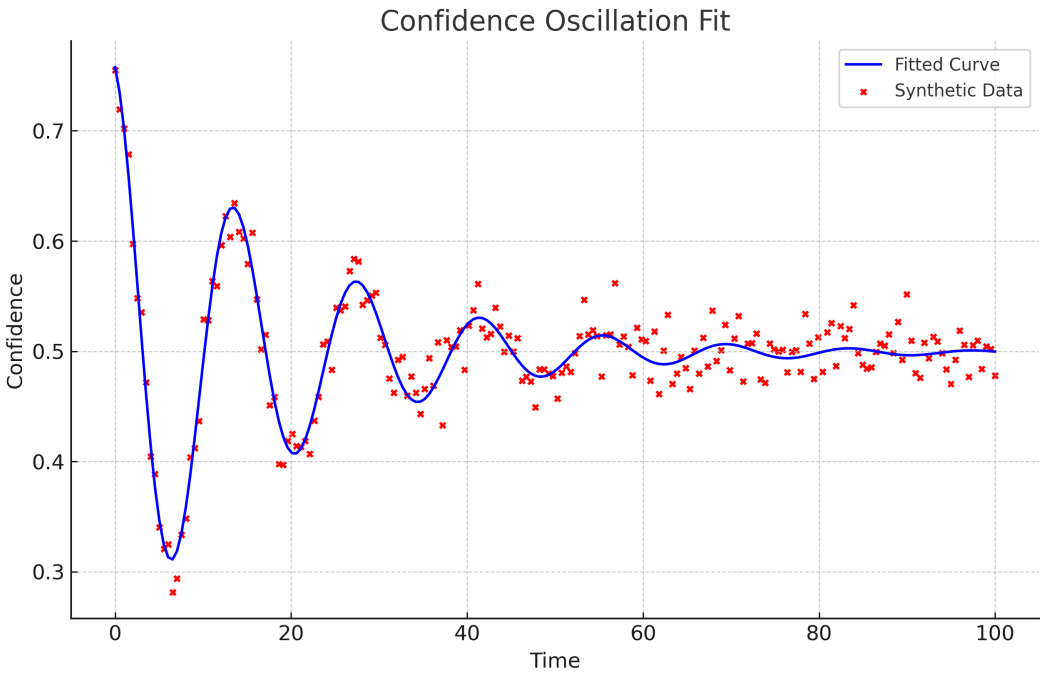

Figure 1: Confidence Fit

Parameters: - Amplitude: Initial confidence level. - Decay rate (τ): Rate at which confidence stabilizes. - Period: Time between oscillation peaks.

The τ parameter controls how exploratory the confidence policy becomes. Higher τ spreads probability mass and encourages more exploratory selections.

The τ effect is critical for understanding solver calibration and risk tolerance adjustments in dynamic environments.