# ShashChess 39: The Evolution of Resilience



A quantum leap in chess theory, inspired by physics

**ShashChess 39 marks a milestone in the evolution of chess engines, cementing its identity not as a simple derivative of Stockfish, but as an engine with a unique and profoundly innovative strategic core. This version pushes the boundaries of positional understanding, fusing chess theory with physics principles to create unprecedented game intelligence.**

## What is ShashChess? A Summary

ShashChess is a cutting-edge chess engine built on Stockfish that revolutionizes the chess approach. Instead of relying solely on numerical evaluation, it integrates:

- **The Shashin Zone Theory:** Classify each position according to a strategic style (e.g., Tal, Capablanca, Petrosian) to dynamically adapt search behavior.

- **WDL Probabilistic Model:** It goes beyond simple evaluation, estimating the probabilities of Win, Draw, and Loss for a deeper understanding of positional nuances.

- **MCTS Hybrid and Reinforced Learning:** Combine alpha-beta search with Monte Carlo Tree Search (MCTS) and search learning for dynamic exploration and real-time strategic adaptation.

## The Big News of ShashChess 39

ShashChess 39 builds on the solid foundation of version 38, introducing crucial improvements that elevate its performance and strategic intelligence.

# 1. The Principle of Minimum Resilience: Thermodynamics on the Chessboard

The most profound innovation of ShashChess 39 is the introduction of the concept of **minimum resilience**, a principle borrowed from thermodynamics, as suggested by Alexander Shashin in his book, for the management of chaotic positions, implemented in the new algorithm `gandtResilientShashinRange()`.

A chess position is treated as a physical system where the engine, faced with a chaotic (high entropy) position, does not rely on a single evaluation, but simulates a "cooling" process to find the most efficient path to stability and advantage.

- **Phase 1: High Energy Approach (Tal Style):** The system explores the most aggressive and tactical path first to see if a quick resolution can be forced.

- **Phase 2: Decay to Stability (Capablanca Style):** If the attack is inconclusive, the engine reduces entropy seeking positional balance.

- **Phase 3: Final Consolidation (Petrosian Style):** If necessary, the system settles into the most robust, lowest-energy state to consolidate its advantage.

This scientific approach allows ShashChess to handle chaos with superior resilience, finding the optimal strategy that fits the intrinsic nature of the position.

## Technical Deep Dive: The Physical Model

- **Entropy (S):** It measures disorder and uncertainty. It is modeled as $S \sim = \sum p(x) \log p(x)$, where $p(x)$ is the probability distribution of the pieces over legal moves.

- **Resilience (R):** It is the ability to resist disturbances (threats). Conceptually defined as $R = E - T \cdot S$, where $E$ is the strategic potential energy and $T$ is the "tactical temperature".

- **Practical Example:** In one position `Tal-Petrosian` (tactical chaos + solid defense), ShashChess 39 evaluates whether a sacrifice is *thermodynamically sustainable* before run it.

# 2. Recognition of Fortresses and Preservation Logic

ShashChess 39 introduce un **sophisticated algorithm for fortress recognition**, positions in which one side, even at a material disadvantage, can force a draw. Thanks to the new flag `theFortress`, the engine is now able to:

- **Avoid unnecessary efforts** in an attempt to win theoretically drawn positions.

- **Playing for the fortress** when he is at a disadvantage.

- **Preserve the fortress** during the ordering of moves, giving priority to moves that do not weaken the defensive structure.

**3. Shashin Framework Expansion: Adaptive Calibration**

The Shashin positional model has been enhanced with finer calibration of the search extensions depending on the area:

- **Hybrid Zones:** More precise management of border areas such as `Capablanca-Tal` and `Capablanca-Petrosian`.

- **Adaptive Margin Adjustment:** The extension margins dynamically adapt to the position style to optimize calculation time:

    - **High Tall:** More aggressive margins (`-55/-75`) to facilitate tactical research.

    - **Capablanca:** Strategic moderation (`+45/+65`) for positional search.

    - **Petrosian:** Defensive prudence (`-25/-35`) to consolidate the position.

### 4. Hybrid MCTS and Reinforcement Learning Enhanced

- **MCTS Improved:** The MCTS engine now includes a node limit (`MCTSMaxNodes`) to prevent excessive memory use and better awareness of the danger to the king.

- **Optimized Learning:** The reward function has been improved to reward not only victory, but also **stability resilient**, reducing the overestimation of speculative sacrifices.

## Scientific Rigor: Testing Strategy and Results

To validate its progress, ShashChess 39 was rigorously tested against the Stockfish benchmark. The number of positions was specifically chosen to minimize statistical fluctuations.

- **Test 1: Suite of 256 Complex Positions**

    - **Configuration:** 1 minute/position, single thread (to further reduce statistical noise).

    - **Result: ShashChess 39: 138 solved positions** vs. Stockfish: 134. Greater accuracy in high entropy locations.

- **Test 2: Match Series on 50 "Nuggets" Positions**

    - **Configuration:** 300 total games (3 reps), 4 threads per motor, on sharp and unbalanced positions.

    - **Result:** Final score of **+64 / =176 / -60** in favor of ShashChess 39.

## Conclusion: Beyond Brute Force

With ShashChess 39, the engine evolves from a tactical powerhouse to a deeply strategic and adaptive system. Thanks to physics-inspired resilience modeling, fortress detection, and refined Shashin logic, it confirms itself as a serious and original contribution to the landscape of modern chess engines.

# Thanks

The success of this project would not have been possible without the contributions of many individuals and institutions. Special thanks to:

- **Afro Ambanelli** for his meticulous and precious collection of "nugget" positions and more.

- **Peter Martan** for his excellent collection of "hard" positions.

- **Massimo Venuto** for tireless software and hardware support.

- **Alessandro Morales** for their fundamental support with Gemini AI, which has opened new frontiers of development.

- All **institutions university** and to an **important financier** for providing the hardware needed for testing and development.

- To all the **chess programmers community** for the criticisms, even if not always constructive, which have pushed this project to constantly improve.