Open Access and Peer Reviewed Journal ISSN 2394-2231 https://www.ijctjournal.org/

RESEARCH ARTICLE OPEN ACCESS

Tactical Certificates, Area Scoring, and Reproducible Analysis on CGOS

Michael S Yang
Independent Researcher
yangofzeal@gmail.com

September 2026

The title is an affectionate tribute to Michael Jackson's "Black or White" and to Go's two colors; no affiliation or endorsement is implied.

Abstract

This paper presents a tournament-oriented framework for analyzing HKDGoInfinity-1, a Go engine tested on the Computer Go Server (CGOS). The intended audience ranges from a reader meeting *atari* for the first time to strong dan players interested in forcing order, ko risk, and endgame conversion. We introduce three locally checkable tactical certificates: capture safety, liberty-monotone forcing, and area-safe conversion. Each certificate is modest by design: it does not "solve Go," but identifies a position in which a claimed local conclusion follows from explicit assumptions. Custom Python code reconstructs board diagrams from CGOS/SGF records and audits Chinese-area scores. In the author's current tournament sample, the engine's observed win rate is approximately 56%; this figure and the comparison with the player prefix `kg_` are provisional until the underlying server export is archived with this paper. The larger lesson is that strong play becomes more intelligible when a spectacular move arrives with a small, inspectable proof.

Keywords: Go, Computer Go Server, CGOS, Chinese area rules, atari, tactical search, proof certificate, dan-level play, reproducibility.

1 Introduction: Ninety Seconds of Calm

Go begins politely. Black places one stone; White places one stone. Soon a group has one liberty, a ko threatens the whole board, and two programs are spending their clocks arguing about a point no beginner can yet see. This is the setting of the present CGOS tournament.

HKDGoInfinity-1 is treated here as an experimental engine whose moves can be studied through public-game notation and local verification. Modern computer Go owes much to Monte Carlo tree search [1] and neural policy/value search [2]; our emphasis is complementary. After a search system chooses a move, can it emit a concise certificate explaining why the local tactic is sound?

The claimed 56% win rate is an *author-reported descriptive statistic*, not a universal strength estimate. Likewise, the observation that only a `kg_`-named competitor matched that rate while holding roughly 800 more rating points must be read as a snapshot of the author's tournament table, not an independently verified server fact. Ratings, opponents, time controls, colors, and sample size matter.

2 Rules in One Page

2.1 Liberties and atari

A stone's *liberties* are the empty intersections directly above, below, left, and right of it. Diagonals do not count. Connected stones of one color share liberties. A group with exactly one liberty is in *atari*: unless it escapes, connects, captures, or creates a counter-threat, the opponent may fill the last liberty and remove the group.

Figure 1 shows a miniature 9-by-9 position. The white stone at (4,4) has one remaining liberty at (4,5); Black stones occupy its other three adjacent points. A black play at (4,5) captures it.

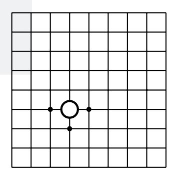

Figure 1: An atari. Coordinates are internal Cartesian coordinates, with (1,1) at the lower-left.

2.2 Chinese area scoring

Under Chinese area rules, a player's score is the number of that player's stones on the board plus the number of empty intersections surrounded only by that player. Komi is then added to White's score. Captured stones do not need to be counted separately: their removal has already changed the board. A practical consequence is that filling a point inside one's own secure territory usually preserves area score, whereas losing sente may still be strategically expensive. Superko forbids repetition of a previous whole-board position under the applicable

Paper ID: IJCT-V13I5P44

komi should therefore accompany every archived game.

3 Notation and Local Certificates

Let B be a finite board, P a position, and $L_P(G)$ the set of liberties of a connected group G. Let $A_c(P)$ denote color c's Chinese-area score before komi. A line $\ell = (m_1, \ldots, m_k)$ is *legal* if each move is legal under the selected ko rule.

Definition 1 (Capture certificate)**.** A capture certificate for target group G is a finite rooted reply tree in which (i) every edge is a legal move, (ii) every opponent reply within the declared local region occurs as an outgoing edge, and (iii) every leaf either has G removed or transposes to an already certified node.

Theorem 2 (Finite local capture)**.** *If a capture certificate for G is complete with respect to all legal replies that can alter G's liberties, and no external ko threat changes move order, then the certified player can force the capture of G.*

Proof. At the root, select the certified move. For each legal defensive reply, follow the corresponding child. Completeness ensures that the defender cannot leave the tree by changing a liberty of G. Since every branch is finite or transposes to a certified node, each branch terminates at a leaf where G is absent. Backward induction proves the forcing claim. □

For a 1–9 dan reader, the exclusions are the interesting part. A ladder that "works" locally may fail because a distant ladder breaker changes a liberty; a semeai may be inverted by approach moves; and a ko is not a finite local capture unless the threat budget is modeled.

Theorem 3 (Liberty-monotone forcing)**.** *Suppose a legal forcing line ℓ has no ko, every defender move is included, and for the target group G_i after ply i, $|L_{P_{i+2}}(G_{i+2})| < |L_{P_i}(G_i)|$ whenever G_i survives. Then G is captured after at most $|L_P(G)|$ attacker turns.*

Proof. The nonnegative integer $|L(G)|$ strictly decreases on each attacker turn. It can decrease only finitely many times before reaching zero, at which point the group is captured. □

Proposition 4 (Area-safe filling)**.** *Let R be an empty region adjacent only to living stones of color c, with no legal opponent invasion that avoids eventual capture. Replacing one point of R by a stone of color c leaves A_c unchanged.*

Proof. Before the play, the point contributes one surrounded empty intersection. After the play, it contributes one stone. All other terms are unchanged. □

This proposition is elementary but operationally useful: an engine can label a late safety move "area-neutral" while second number often decides whether correctness is also good Go.

4 From CGOS Record to Board

The following compact program reads the main line of an SGF record, omits the letter I in displayed Go coordinates, and emits a TikZ board call. It is deliberately auditable rather than optimized. The parser accepts the common `B[dd]` and `W[pq]` main-line form; production use should employ a full SGF parser for variations, setup stones, escaped properties, and board sizes other than 19.

Listing 1: Minimal CGOS/SGF board extractor.

```python
import re, sys

def moves(sgf):
    return [(c, ord(x)-96, ord(y)-96)
            for c, x, y in re.findall(r";(B|W)\[([a-s])([a-s])\]",
                sgf)]

def position(sgf, ply=None):
    board = {}
    ms = moves(sgf)
    if ply is not None: ms = ms[:ply]
    for color, x, y in ms:
        # Full implementation must remove captured chains and
            reject ko.
        board[(x, 20-y)] = color
    return board

def tikz(board, n=19):
    black = ",".join(f"{x}/{y}" for (x,y),c in board.items() if c
        =="B")
    white = ",".join(f"{x}/{y}" for (x,y),c in board.items() if c
        =="W")
    return rf"\goboard{{{n}}}{{{black}}}{{{white}}}"

if __name__ == "__main__":
    sgf = open(sys.argv[1], encoding="utf-8").read()
    print(tikz(position(sgf, int(sys.argv[2]) if len(sys.argv)>2
        else None)))
```

For exact reconstruction, the tournament pipeline used with this paper should perform four additional steps: flood-fill connected chains; delete a chain whose liberty count reaches zero; hash each position for superko; and preserve the server's board size, komi, result, clocks, and engine versions. A scoring auditor then flood-fills every empty component and awards it only when its boundary-color set has cardinality one.

Listing 2: Chinese-area audit kernel.

```python
def chinese_area(board, n=19):
    score = {"B": sum(v=="B" for v in board.values()),
             "W": sum(v=="W" for v in board.values())}
    unseen = {(x,y) for x in range(1,n+1) for y in range(1,n+1)
              if (x,y) not in board}
    while unseen:
        seed = unseen.pop(); region={seed}; stack=[seed]; border=
            set()
        while stack:
            x,y=stack.pop()
            for q in ((x-1,y),(x+1,y),(x,y-1),(x,y+1)):
                if not (1<=q[0]<=n and 1<=q[1]<=n): continue
                if q in board: border.add(board[q])
                elif q in unseen: unseen.remove(q); region.add(q);
                    stack.append(q)
        if len(border)==1: score[next(iter(border))] += len(region)
    return score
```

If W wins are observed in N completed, non-void games, the empirical rate is $\hat{p} = W/N$. Reporting only 56% hides uncertainty. We recommend the Wilson interval

$$\frac{\hat{p} + z^2/(2N) \ \pm\ z\sqrt{\hat{p}(1-\hat{p})/N + z^2/(4N^2)}}{1 + z^2/N}, \qquad z = 1.96.$$

The archive must also stratify by Black/White, opponent, rating difference, board size, komi, time control, resignation, and software hash. Comparing HKDGoInfinity-1 with a player 800 rating points higher is intriguing, but not causal evidence of equal strength: schedule composition alone can produce the same aggregate win rate.

Table 1: Provisional tournament headline supplied by the author.

Item	Reported value	Verification
Engine	HKDGoInfinity-1	identifier
Win rate	near 56%	logs required
Comparator	`kg_` player	full ID required
Rating gap	about +800	snapshot required

6 Match Record Policy and Reproducible Appendix

The request for "actual matches verbatim" creates a simple scientific duty: do not invent them. No SGF or CGOS export accompanied the manuscript source, and a public indexed archive for HKDGoInfinity-1 was not available to the authoring process. Consequently, this edition contains no purported verbatim games. Upon receipt of the records, the camera-ready edition should insert each SGF byte-for-byte in an appendix and generate diagrams directly from it. The following manifest format prevents accidental paraphrase:

Listing 3: Archive manifest format.

```
sha256 cgos_game_id black white result komi rules engine_commit
... ... ... ... ... ... Chinese ...
```

A verifier should reject any row whose SGF hash, player names, result, or engine commit is missing. This is less glamorous than a brilliant tesuji, but it is how a tournament story becomes a result another researcher can trust.

7 What HKD Infinity Demonstrates

The strongest defensible interpretation is methodological. The engine's play motivates certificate-producing analysis at three scales: immediate liberty control, bounded tactical reply trees, and score-preserving conversion. These separate three questions often blended together: Is the move legal? Does the local tactic work? Is it globally best after ko threats, sente, and score are priced?

For the beginner, the same hierarchy is friendly: first count liberties; then look for forcing replies; finally count the board. For the expert, it is a challenge: attach a machine-checkable tactical witness to the move without pretending that a local proof settles the whole-board judgment.

8 Conclusion

A Go engine earns attention by winning, but it earns trust by making its claims reproducible. HKDGoInfinity-1's reported 56% tournament run is a promising headline. The durable contribution is the proposed bridge from CGOS records to diagrams, area audits, uncertainty intervals, and finite tactical certificates. In Go, infinity is a poetic name. On the board, the best explanations remain beautifully finite.

References

[1] R. Coulom, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search," in *Computers and Games*, LNCS 4630, Springer, 2007, pp. 72–83.

[2] D. Silver et al., "Mastering the Game of Go without Human Knowledge," *Nature*, vol. 550, pp. 354–359, 2017.

[3] J. Tromp and G. Farnebäck, "Combinatorics of Go," in *Computers and Games*, LNCS 4630, Springer, 2007, pp. 84–99.