LittlePly as a chess agent

CM 14 and the harness that turns predictions into play

Early draft v0.2 | 10 November 2026

Abstract. LittlePly is a chess agent made from a learned model and a designed harness. CM-14 supplies a move policy and fixed-White win/draw/loss estimates. The harness maintains legal state, chooses where to search, reuses computation, coordinates bounded Stockfish assistance, manages the clock and executes moves. The measured player is their combination. We describe this system, separate several dated evaluation configurations, and review CM-15 and CM-16 as lessons from unsuccessful replacement attempts. The available development results justify retaining CM-14 as the reference, but do not establish a general Elo rating, isolate the contribution of each component, or prove that auxiliary training objectives reduce strength.

1 The unit of research is the agent

A plausible next move is only one ingredient of reliable play. A strong policy can miss a tactical reply, fail to convert a win, or lose while thinking. Conversely, legal filtering and external search can make a weak raw prediction look competent. We therefore identify the complete agent by its checkpoint, harness release, configuration and execution environment. Changing any of these can change the result.

This framing gives LittlePly two connected research questions: how to train a more useful policy and outcome estimator, and how to turn those predictions into stronger decisions within a fixed compute and clock budget. Model diagnostics and game outcomes answer different parts of those questions. Browser play, which uses the model and legal filtering without online search or Stockfish assistance, is a different agent from the online system. [1,2]

2 CM 14 is the learned component

CM-14 is an 86,966,016-parameter decoder-only Transformer with 12 layers, width 768, 12 heads, 512 context positions and 1,975 vocabulary entries. A complete UCI move is one token, including promotions. It reads game history from the standard starting position. Move logits and fixed-White outcome logits are normalized separately; outcome labels are not input context. The WDL values are learned game-result estimates, not established calibrated probabilities or engine action values. [1]

The checkpoint was trained from random initialization with hybrid Muon/AdamW optimization, not initialized from CM-13 or the later CM-16 experiment. An eight-hour first stage used 23,691,842 selected human games and 172,656 engine games, targeting 20% of eligible supervised moves from the engine source. An eight-hour continuation used a frozen 8,137,577-game human view balanced by supervised moves across 53 months. The retained checkpoint is update 210,395. Repeated exposures are not unique games or positions. [1]

All eligible moves of both players receive next-move supervision; engine book moves remain context only. A deterministic 15% sample, stratified over game thirds, receives WDL supervision. Two-GPU training uses BF16 autocast with FP32 master state, packed game-isolated attention, Muon for transformer matrices and AdamW for the remaining parameters. The dated reproduction recipe records the exact settings and time-based schedule. [1]

3 The harness is part of the player

The following describes the serving architecture documented on 8 November. Individual evaluations freeze their own versions and settings. In particular, the older depth-3 decoder and later depth-6 assisted configuration are separate experimental systems. This paper does not silently apply one configuration's score to another. [2-4]

State and legal actions

The controller owns the board, full history, repetition state, terminal rules and clocks. The policy is normalized over legal moves, excluding outcome and special tokens. Context is not silently cropped. Legal output therefore reflects both the model and the rules implementation; it is not proof that the unfiltered model has learned every rule.

Selective neural search

The harness starts with the legal policy and expands alternatives when that policy is insufficiently concentrated. In the documented later runtime, uncertainty is recomputed at expanded nodes for either player: confident positions can follow one continuation, while ambiguous positions open more branches. The neural opening gate keeps absolute plies 0-5 policy-only; separate helpers remain available. WDL estimates supply backed-up values, with White maximizing and Black minimizing White's expected score, W + 0.5D. Probability concentration is a heuristic trigger, not calibrated uncertainty.

This is selective search over model-proposed candidates. A defense outside the explored branch set can be missed. A depth, node or time limit describes the computation allowed, not a guarantee of full-width search. The earlier 10-second experiment used a top-three, depth-3 decoder and different opening eligibility, as recorded in its own protocol. [3]

Memory and reuse

Persistent search work, history-keyed policy/WDL caches and Transformer key/value caches reduce recomputation in the later runtime. Reuse must remain compatible with model identity and game history. Identical boards need not have identical repetition histories. These caches reuse inference; they do not update weights or train on the live game. [2]

Bounded external assistance

Stockfish 19 supplies bounded mate and defensive evidence alongside the neural process. Freshness, exact-history and legality checks constrain its use. A mate report is engine evidence, not an independently verified proof certificate. A separate sparse-endgame conversion rule may select a Stockfish move after qualifying estimates on consecutive bot turns. These interventions are explicit parts of the agent. Opening books and Syzygy tablebases were not enabled in the documented live configuration. [2]

Clocks and execution

The harness allocates time from the remaining clock and increment, reserves time for execution, and respects deadlines and node limits. It retains a completed choice or falls back to a legal move rather than waiting indefinitely. The service also handles concurrent games, events, stale results, submissions and acknowledgements. Correct predictions alone do not guarantee correct execution. The fixed-cap decoder's recorded clock loss makes this distinction measurable. [2,3]

Decision loop. Observe state and clock; obtain legal policy and WDL; search and consult helpers within budget; validate and submit; record the decision and resulting state. Each intervention should retain its source so that a system win is not attributed automatically to the learned policy.

4 Evidence belongs to a frozen configuration

Chess score is (wins + half the draws) / games. The rows below are separate cohorts, not a single time series of controlled improvements. All use Sunfish as the listed opponent, with 1,000 requested nodes per move in the documented protocols. [1,3-5]

System and cohort	W / D / L	Score	Scope
CM-14 historical mate-only	68 / 8 / 24	72.0%	100 games; no conversion
CM-14 matched CM-15 reference	Reported as score	72.5%	100 games; mate + conversion
CM-15 initial completed run	64 / 9 / 27	68.5%	100 games; matched helpers
CM-14 10-second neural search	73 / 12 / 15	79.0%	100 games; no Stockfish helpers
CM-14 later assisted system	49 / 1 / 0	99.0%	50 games; 25 opening pairs

The 100 game search experiment

The unassisted depth-3 search configuration scored 79.0% over 50 color-swapped opening pairs, including one genuine clock loss. Its opening-pair bootstrap 95% interval was 72.5%-85.5%. Relative to the historical 72.0% mate-helper and 72.5% mate-plus-conversion references, estimated differences were +7.0 points (interval -3.5 to +18.0) and +6.5 points (-4.0 to +17.0). Every improvement interval crossed zero. These are exploratory comparisons with different helper policies, compute and host conditions, not isolated causal gains. [3]

On a preserved, nonrandom 48-game subset, the 10-second and 3-second settings scored 79.17% and 68.75%. The paired difference interval was -5.21 to +25.0 points. Recorded model decision time increased about 1.94 times. The fixed 10-second ceiling did not adapt to the remaining clock; the loss on time belongs in the headline score. The forfeiting attempt lacked saved telemetry, so its duration cannot be reconstructed as if observed. [3]

The later 50 game assisted result

CM-14 update 210,395 with release 4cee1bf258f6-675c36654c3b scored 49 wins and one draw: 98% wins and 99% chess score. The protocol used 25 paired openings with 16 forced plies, 300+2 clocks reset after the opening, CPU neural search capped at 10 seconds, depth 6 and 4,096 nodes, plus Stockfish mate, conversion and draw-save assistance. There were no recorded forfeits or illegal moves, and cleanup was complete for all 50 games. Four cgroup diagnostic copies were missing. [4]

Recorded candidate decisions came from the model policy (492), neural WDL search (596), Stockfish conversion (92) and Stockfish mate assistance (223). These counts document use, not causal credit: a policy-selected move can still depend on the surrounding harness, and helper counts are not independent experiments. The small selected suite was repeatedly inspected under shared CPU load. This is a 50-game development result, not completed 100-game validation, a general Elo estimate, or proof that the harness alone caused the higher score. [4]

5 Lessons from CM 15 and CM 16

CM 15 did not demonstrate a strength improvement

CM-15 added a sampled future-move task alongside next-move and outcome prediction. Its completed initial Sunfish run scored 68.5% against the matched CM-14 reference's 72.5%, a difference of -4 percentage points with a paired interval of -15 to +7 points. Both used the frozen mate checks and 97% endgame-conversion protocol. This supports no demonstrated improvement; the interval does not prove harm or equivalence. These results describe the initial completed run, not a later stopped continuation. [5]

Against CM-14 directly, CM-15 recorded 7 wins, 7 draws and 11 losses in 25 games. The 24 paired games scored 41.7%, with interval 29.2%-54.2%; the extra drawn Black game was excluded from that paired interval. Outcome forecasts were more informative later in games, but the overall Brier improvement over a cross-fitted baseline also had an interval crossing zero. Nine draws limited calibration evidence. Better-looking forecasts or auxiliary-task performance cannot substitute for a playing-strength test. [5]

CM 16 was a changing multiobjective experiment

CM-16 combined next-move prediction, WDL, a future-move target and board-occupancy reconstruction. It also changed initialization, data and supervision details relative to CM-14. The selected continuation ran from 5 November 09:42 UTC to 9 November 17:06 UTC: 103.4 elapsed wall hours, about 4.3 days, reaching update 665,007 and 5,480,928,080 recorded ordinary target visits. Visits include repeated exposure; wall time includes interruptions and is not GPU-hours. [6]

The retained trajectory changed during that interval: equal ordinary-loss weighting began at update 247,803; shared-history policy/WDL at 307,472; and next-move/WDL-only training at 418,201. A result from the whole lineage cannot isolate the original four-task objective or any one repair. CM-16 remained unpromoted after weaker historical evaluations against the CM-14 reference. [6]

The recorded intermediate update-369,403 comparison scored 33 wins, 2 draws and 15 losses against Sunfish, and 8 wins, 0 draws and 42 losses against CM-14, each over 50 games. This is historical intermediate-checkpoint evidence, not a result for final update 665,007. The latest-checkpoint GPU cohorts failed their audit because required GPU telemetry was missing, and the final assessment stopped before a clean conclusion. Their scores are excluded here. [4,6]

The compute tradeoff is a hypothesis to test

Auxiliary tasks can consume sequence space, labels and compute that could otherwise train the policy. Lower useful next-move throughput and weak playing results motivated concern about this tradeoff. The records do not supply a matched estimate of how much strength was lost to that mechanism: data, initialization, weighting, WDL semantics, training duration and evaluation harness also changed. Short throughput tests and interrupted wall-clock averages are not interchangeable measures. [6,7]

The practical lesson is to keep a strong simple baseline and require each added task to earn its compute budget. Measure ordinary targets per second and playing strength at matched budgets, then isolate objectives one at a time. CM-15 and CM-16 provide reasons for those controls, not evidence that future-move prediction or board supervision is inherently ineffective.

6 A testable model and harness research program

Freeze identity before comparison. Record checkpoint and export hashes, harness commit/configuration, helper versions and budgets, device and host load, openings, colors, clocks, adjudication and failure handling. The same weights with different assistance or time control are different agents. Public source snapshots should retain their dates; the November 3 reproduction bundle is not an exact reproduction of the later runtime. [1,2]

Separate the two experimental axes. For harness work, hold CM-14 fixed and compare policy-only, neural-search, helper and complete configurations under matched conditions. For training work, hold the harness fixed and vary one objective or training choice. Compare both equal compute and equal ordinary-target exposure, with multiple seeds where feasible. Caching experiments should report latency and effective search effort as well as outcomes.

Use fresh evaluation evidence. Keep the repeatedly inspected development suite for diagnosis and reserve a fresh holdout for promotion decisions. Color-swap openings, retain the pairing in uncertainty estimates, and specify the analysis before looking at results. Include clock losses and other failures in the primary outcome. Unfinished or audit-invalid cohorts must remain visibly incomplete rather than silently counted as clean wins or losses.

Measure the mechanism as well as the score. Report move source, latency, search depth and node count, cache use, stop reason, helper intervention, clock remaining and fallback. For training, distinguish unique data, repeated visits, task examples, active labels and GPU time. Check WDL semantics and calibration separately from move prediction. Missing historical telemetry cannot be repaired by guessing.

Conclusion. CM-14 is the current reference because it remains the retained model within a working, explicitly assisted agent. The strongest reported development result belongs to a particular model-plus-harness configuration. The next claim to earn is a reproducible improvement under controlled budgets and fresh evaluation, with enough attribution to say whether it came from learning, search, assistance or execution.

Sources and reproducibility

[1] LittlePly, CM-14 release record and November 3 reproduction guide. Source records: content/cm14/release.json and content/reproduction/guide.html. Model, training and historical mate-only cohort.

[2] LittlePly, agent architecture verified 8 November 2026. Source snapshot: lib/agent-positioning.mjs. Later serving changes require a new configuration identity.

[3] LittlePly, CM-14 10-second uncertainty-search evaluation, 6 November 2026. cm14-search10-report.md records paired bootstrap methods, timing and limitations.

[4] LittlePly, CM-14 assisted 50-game summary and companion PGN, completed 8 November 2026. The summary also preserves the intermediate CM-16 comparison.

[5] LittlePly, CM-15 completed initial experiment, study 19. Source record: content/studies/studies.json. Retrospective game thirds and later continuations remain separate.

[6] LittlePly, CM-16 outcome record, 9 November 2026. cm16-outcome.json preserves trajectory changes, periodic results, audit exclusions and source hashes.

[7] LittlePly, CM-16 dated adaptation recipe and short throughput experiments. Source: content/studies/cm16/recipe.html. Historical launch statements do not describe the later stopped state.

Records [3], [4] and [6] are stored under content/studies/agent-20261109. The companion v0.2 source manifest records full SHA-256 input identities. Retained CM-14 inference export: 4cee1bf258f6061843f6073cd21d65f8981153ae47108551df617cb3053eea72. This draft supersedes v0.1's emphasis on the CM-16 recipe; the original PDF remains unchanged. Links lead to project evidence, not independent peer review.