

Reading Depth Does Not Transfer: The Distillation Ceiling in Computer Go

Taiki Kojima

koji3744@hotmail.com

Kojima Family Clinic <https://orcid.org/0000-0003-2069-6079>

Research Article

Keywords: knowledge distillation, computer Go, MCTS, evaluation methodology, negative results, search depth, LLM vs game AI

Posted Date: June 9th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9966155/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Abstract

Knowledge distillation has been spectacularly successful in large language models, where small students efficiently approximate large teachers because both operate at the same task abstraction: next-token prediction. We report a structurally different outcome when distillation is applied to game AI for the game of Go. Using the pure-distillation pipeline introduced in Kojima (2026, in preparation; hereafter Paper I), we trained a 7M-parameter student on three successive rounds of KataGo b18c192 labels at 4000 visits per position; the training corpus grew from 45 thousand positions (the Paper I endpoint) to 120 thousand and then to 200 thousand. Across this 4.4× data scaling, the student’s top-1 oracle agreement on the fixed Move Quality evaluation set *monotonically degraded*: 0.170 → 0.120 → 0.040 (top-3: 0.270 → 0.160 → 0.080). Value MAE was non-monotonic (0.303 → 0.193 → 0.290) and showed no net improvement across the 8× increase from generation 2 to generation 3.

We interpret this as evidence for a structural difference between LLM and game-AI distillation. The teacher’s strength in Go is roughly half neural-network evaluation and half MCTS search depth; **the search-depth component does not transfer through single-position labels**, regardless of how many such labels are provided. A complementary five-intervention study (Section 5) tested value-head capacity, β_z target blending, score-head re-activation, and ownership-weight tuning; four interventions failed and one auxiliary intervention (ownership-weight tuning) produced measurable improvement, consistent with the same ceiling.

1. Introduction

Distillation in large language models has become a default scale-efficiency technique: small open-source students (Alpaca, Vicuna, WizardLM, DeepSeek’s family) reliably recover a large fraction of a much larger teacher’s quality by training on the teacher’s outputs. The implicit reason is that teacher and student do the same thing — next-token prediction — so the teacher’s output distribution is exactly the supervision target the student needs.

In MCTS-based game AI the situation is different. Strong Go engines derive their playing strength from the *interaction* of two components: a neural network that outputs per-position priors, and a Monte Carlo Tree Search procedure that refines those priors through deep tactical reading. A static per-position label — KataGo’s policy distribution after 4000 visits, its value, its expected ownership map — represents the teacher’s *output*, not the teacher’s *process*. The student, at runtime, runs its own MCTS with its own (much smaller) budget. This raises a structural question: **what survives distillation when the teacher’s strength has a search-depth component that the student’s runtime cannot reproduce?**

We report a three-generation distillation experiment that bears on this question. The student, pipeline, evaluation methodology, and teacher are those of Paper I (Kojima, 2026, in preparation); we add two further distillation generations beyond the Paper I endpoint, each adding ~ 80,000 newly-labeled KataGo self-play positions at the same 4000-visit budget. On the fixed 100-position Move Quality evaluation set, top-1 oracle agreement declined monotonically from 0.170 to 0.120 to 0.040 — a 4.25× decrease over a

4.4× data increase — with similar monotone decline in top-3 and no net improvement in value MAE. KL divergence between student and oracle policy *did* improve monotonically (3.498 → 3.242), but the student’s argmax shifted away from the oracle’s top move. We interpret this as evidence that the distillation pipeline has hit a ceiling that is not relieved by additional data of the same kind.

1.1 Contributions

1. **A three-generation monotone-degradation experiment.** Holding architecture, optimizer, augmentation, hyperparameters, and labeling protocol fixed, scaling the distillation corpus from 45k to 200k positions produces *monotone decline* in top-k oracle agreement. We are not aware of prior direct experimental evidence in computer Go that distillation has a quantitative ceiling that more of the same data cannot overcome.
2. **A structural-asymmetry argument** between LLM distillation and MCTS-based game-AI distillation, grounded in the observation that single-position labels capture only the teacher’s static evaluation and not its search depth (Section 6.1).
3. **A five-intervention compatibility study** (Section 5) confirming that the same ceiling resists architectural deepening of the value head, β_z target-blending changes, score-head re-activation, and direct loss-weight scaling. Only auxiliary ownership-weight tuning produced measurable improvement.

2. Related Work

2.1 Distillation in LLMs vs game AI

Knowledge distillation (Hinton et al., 2015) trains a small student to match a larger teacher’s soft output distribution; soft targets encode the teacher’s relative preferences over outputs more richly than one-hot labels. NLP has scaled the paradigm dramatically (Alpaca — Taori et al., 2023; Vicuna — Chiang et al., 2023; WizardLM — Xu et al., 2023): small students routinely recover 80–90% of the teacher’s benchmark quality. In game AI, distillation has typically appeared as a *post hoc* compression step — large self-play network first, smaller deployment student second (Wu, 2019). Paper I (Kojima, 2026, in preparation) introduced distillation as the *primary* training signal for a small Go network at moderate scale (50k positions); the present work extends that scale fourfold and asks what happens at the upper end of the data-scaling curve.

2.2 Search-vs-evaluation decomposition in MCTS agents

The MCTS-based game-agent literature (Silver et al., 2016, 2017, 2018; Wu, 2019) consistently observes that strong play emerges from the *combination* of network evaluation and tree search, and that at small network scales (≤ 10 M parameters) the search component carries a disproportionate share of the total strength — on the order of several stones in 19×19 Go. We use this background in Section 6 to interpret the negative result reported here.

2.3 Negative results and LLM-vs-game-AI transfer analogies

Workshops at ICLR (“I Can’t Believe It’s Not Better”), NeurIPS, and EMNLP have established venues for mechanistically-motivated negative results. A few recent informal essays have argued that LLM-style distillation should generalise to other imitation-shaped tasks; the present paper offers contrary evidence for search-based agents, with quantitative measurement of the gap. We are not aware of a prior paper making this argument with direct experimental evidence in computer Go.

3. Experimental Setup

Architecture, optimizer, augmentation, Move Quality evaluation protocol, hardware, and teacher engine are unchanged from Paper I and are summarised here only briefly. Full details are in *Paper I*, Sections 3 and 4 and *Appendix C*.

3.1 Student, teacher, evaluation (— see Paper I)

The student is PVSNetVBO, a 7M-parameter residual ConvNet (128 channels $\times$ 6 residual blocks, four output heads: policy, value, score, ownership). The teacher is KataGo v1.14.0 b18c192 in analysis mode at maxVisits = 4000. Hardware is a single NVIDIA RTX 4000 SFF Ada. Move Quality is evaluated on the fixed 100-position evaluation set from Paper I (50 KGS + 50 KataGo self-play), labeled once by the teacher at 4000 visits and reused unchanged; metrics are top-1 and top-3 oracle agreement, KL divergence, cross-entropy at the oracle top-1, and value MAE, all computed on the student’s *raw forward pass* (no MCTS). We additionally use the 50-position tactical-emphasis subset described in Paper I Section 5.5.

3.2 Three-generation distillation protocol

All three generations use identical hyperparameters: Adam with cosine LR decay from 3×10^{-4} to 0, batch size 128, 10 epochs, 8-fold dihedral + colour-swap augmentation, loss weights $w_{policy} = 1.0$, $w_{value} = 3.0$, $w_{ownership} = 0.7$, $w_{score} = 0$, and $\beta_z = 0.7$. Each generation initialises from the previous-generation checkpoint via `--init_any`. The only variable across the three generations is the corpus (Table 1): generation 1 is the Paper I endpoint at $\sim 45k$ positions; generation 2 adds 80k freshly-labeled positions from a 1,000-game KataGo self-play run (seed 42); generation 3 adds another 80k from a disjoint 1,000-game run (seed 4242). Replay extras are identical across generations.

Table 1

Three-generation training-corpus composition. The two self-play pools (v1, v2) are disjoint by construction (separate seeds, separate SGF directories, independently labeled). Replay extras are identical across generations.

Generation	Init from	Primary new corpus	Replay extras	Total
d14w_own07	d14p (Paper I)	(Paper I endpoint)	KGS 10k, kgsp30k	≈ 45,000
d15_kgsp	d14w_own07	sp1000_v1 (80k, seed 42)	kgsp30k, KGS 10k	120,000
d16_kgsp	d15_kgsp	sp1000_v2 (80k, seed 4242)	sp1000_v1, kgsp30k, KGS 10k	200,000

A note on the value MAE column reported below: with $\beta_z = 0.7$ and non-zero score targets, the value head is supervised toward $0.7 \cdot z + 0.3 \cdot s$. For d14w_own07 the s targets are zero throughout; for d15_kgsp and d16_kgsp they are non-zero for the freshly-labeled portion. Cross-generation value-MAE comparison is therefore *not* a clean measurement of value calibration alone; see Section 4.2 and Paper I (Section 5.2 footnote).

4. The Three-Generation Result

4.1 Move Quality across generations

Table 2

reports the Move Quality of d14w_own07, d15_kgsp, and d16_kgsp on the 100-position general evaluation set. Table 3 reports the same on the 50-position tactical set. Figure 1 plots the data.

Generation	Corpus	top-1	top-3	KL	CE	value MAE
d14w_own07 (Paper I endpoint)	45k	0.170	0.270	3.498	4.895	0.303
d15_kgsp (+ 80k SP)	120k	0.120	0.160	3.360	5.080	0.193
d16_kgsp (+ 80k more SP)	200k	0.040	0.080	3.242	5.018	0.290

Table 2

Move Quality on the 100-position general evaluation set. top-k oracle agreement degrades monotonically across the three generations even as the training corpus grows 4.4x; KL improves while value MAE is non-monotonic with no net gain.

Generation	top-1	top-3	KL	CE	value MAE
d14w_own07	0.220	0.320	4.141	4.845	0.400
d15_kgsp	0.160	0.300	3.656	4.690	0.441
d16_kgsp	0.100	0.200	3.515	4.465	0.398

Table 3. Move Quality on the 50-position tactical evaluation subset. The same monotone-decline pattern in top-k is reproduced, with the magnitude smaller (top-1: 0.220 $\rightarrow$ 0.100).

Three observations stand out.

1. **(a) Monotone degradation in top-k.** top-1 oracle agreement on the general set drops 0.170 $\rightarrow$ 0.120 $\rightarrow$ 0.040 – a factor of 4.25 in the metric Paper I identified as the most diagnostic. Top-3 follows the same monotone pattern (0.270 $\rightarrow$ 0.160 $\rightarrow$ 0.080). The tactical-set numbers show the same monotone decline, with a smaller magnitude (0.220 $\rightarrow$ 0.100).
2. **(b) Non-monotonic value MAE, no net improvement.** Value MAE goes 0.303 $\rightarrow$ 0.193 $\rightarrow$ 0.290 on the general set. The intermediate generation has a substantially lower value MAE than either neighbour, but the third generation has *not* improved on the first despite an 8 $\times$ increase in training-corpus size. We discuss the mechanism in Section 4.2.
3. **(c) KL improves, top-k worsens.** KL divergence monotonically decreases (3.498 $\rightarrow$ 3.360 $\rightarrow$ 3.242), but top-k argmax agreement gets worse. The student’s distribution-shape is becoming closer to the oracle’s in the cross-entropy sense, while its *argmax* shifts away from the oracle’s top move – the opposite of what move-by-move play requires.

4.2 The value-MAE non-monotonicity

Two mechanisms contribute. First, with $\beta_z = 0.7$ the value head is supervised toward $0.7 \cdot z + 0.3 \cdot s$, but s was zero in d14w_own07 (the head learned $0.7 \cdot z$) and non-zero in d15_kgsp and d16_kgsp (the head learned a blend of game-outcome and score-margin). The MAE against z_{oracle} is therefore not a pure measurement of value calibration. Second, the fraction of training where the head sees real s targets grew from 67% in d15 to 80% in d16, so the head fits the blend more cleanly. We cannot cleanly separate the two mechanisms in the present experiments. What we can say: the *direction* of change from d15 to d16 – worse value MAE despite more data of the same kind – provides no evidence that more data is helping value calibration. The 8 $\times$ increase produced no net improvement.

4.3 Qualitative observations

Sabaki-played qualitative experience (same strong-amateur evaluator, sims = 4000, $c_{\text{puct}} = 1.6$, no Dirichlet noise): d14w_own07 plays a recognisably distilled-from-KataGo style with the limitations documented in Paper I; d15_kgsp plays qualitatively similarly in head-to-head against d14w_own07 (the evaluator reported outcomes as ambiguous), but against an unrelated reference engine (LeelaZero at comparable nominal strength) d15 lost most games. d16_kgsp was not Sabaki-tested at length given the strong negative Move Quality signal; the limited play we inspected was consistent with the metric – moves the evaluator characterised as "less coherent" than either predecessor.

A further qualitative observation, **AI-versus-AI style clash**: when our distilled networks play one another, the same evaluator reports the moves as visibly worse than when either network plays the unrelated LeelaZero reference. We read this as policy fragility induced by the distillation procedure – networks

distilled from the same teacher share a narrow neighbourhood of the teacher’s output distribution and step quickly out of that neighbourhood when they play each other. This is a methodological warning to readers planning their own distillation experiments: *do not evaluate distilled networks only against each other*.

5. The Five-Intervention Compatibility Study

Prior to the d16 result, we ran a five-intervention study on the d14p / d14w_own07 baseline, each intervention modifying one variable at a time. Table 4 summarises the outcomes; the full per-intervention narrative appears in an earlier internal draft (“Five Interventions, One Success”) and is available on request.

Table 4

Five-intervention compatibility study on the d14p / d14w_own07 baseline. Four of five interventions failed to improve live play; only auxiliary ownership-weight tuning succeeded.

#	Intervention	Variable	Outcome	Notes
1	Direct w_value scaling	$w_value \in \{1, 2, 3\}$	<i>mixed</i>	Inverted-U; optimum at $w_value = 2$ (d14q). Value MAE improves modestly but policy regresses at $w_value = 3$.
2	β_z target-blend sweep	$\beta_z \in \{0.7, 0.85, 1.0\}$	<i>failed</i>	$\beta_z = 1.0$ produced the highest top-1 but visibly weaker Sabaki play (MQ–Sabaki divergence).
3	Deeper value head	add FC layer ($4 \rightarrow 256 \rightarrow 128 \rightarrow 1$)	<i>failed</i>	All metrics degrade; tactical top-1 – 36% vs d14w_own07. Capacity is not the operative constraint.
4	Score-head re-activation	$w_score = 0.1$; relabel 5k positions with non-zero s	<i>failed</i>	Top-1 + 5.9%, value MAE – 14%, but live play degrades. Trunk biases toward territorial reading at the cost of tactical precision.
5	Ownership-weight increase	$w_ownership \in \{0.3, 0.7, 1.0\}$	succeeded	Only intervention with measurable positive effect. $0.3 \rightarrow 0.7$ produces d14w_own07, the first model the evaluator prefers over d14p baseline. $0.7 \rightarrow 1.0$ degrades.

The pattern is consistent with the three-generation result of Section 4: the network can absorb auxiliary supervision (ownership) but resists direct interventions on the value head and on loss-weight balance. Multi-task *balance* is the operative variable; direct capacity or direct supervision is not.

6. Discussion: Why Distillation Hit a Ceiling

6.1 The asymmetry between LLM and Go distillation

We argue that distillation is differentially effective in LLMs versus MCTS-based game AI because of a structural difference in what “the teacher’s strength” is made of. In an LLM, the teacher’s strength reduces, at inference, to a *single forward pass per output token*. A large GPT-4 makes one such pass; a small Vicuna also makes one. The difference between them is captured at this single-step level, and distillation transfers exactly that distribution. There is no procedural component *between* forward passes.

In Go, the teacher’s strength is the *interaction* of two components. The network produces per-position priors; MCTS performs deep tactical search using those priors. Published results suggest the two combine multiplicatively, with the search component carrying roughly half the practical strength at small network scales. When we distill, we transfer (a copy of) the network. We do not, and cannot via single-position labels, transfer the search component, which exists only as a runtime procedure operating on the network the student is given. The student inherits the network but runs its own MCTS under its own (much smaller) budget. The student’s effective playing strength is therefore *bounded above* by (student’s network at student’s search budget), not by (teacher’s network at teacher’s search budget). The fraction of the teacher’s playing strength that distillation can transfer is the fraction that lives at the per-position-evaluation level — substantially less than 1 for any MCTS-based agent.

This is the asymmetry. In an LLM, *the teacher’s strength lives at the level distillation transfers*. In Go, *roughly half of the teacher’s strength lives at a level distillation does not transfer*. The 4.4× data-scaling experiment of Section 4 tests this prediction quantitatively. If the LLM analogy were operative, monotone scaling improvement would be expected. What we observe is monotone *degradation* in the metric most diagnostic of policy quality, with no net improvement in value calibration — consistent with the student having *already* absorbed the per-position component of the teacher’s strength at the smaller-corpus scale.

6.2 What the result does not show, and practical implications

Three corollaries forestall misreadings. **First**, the result does not say *more compute* is useless: a larger student (32M–64M parameters) has a higher per-position ceiling and may absorb data the 7M student cannot; our limited 160-channel ablation was negative at smaller corpora and is worth repeating at 200k. **Second**, the result does not say distillation in Go is *uniformly* useless: Paper I demonstrated that distillation works in the early-stage regime where the student is far from its evaluation-capacity ceiling (top-1 went from 0 to 0.16 across four generations). The ceiling identified here is a *small-student* ceiling. **Third**, the result does not say search-depth is *un-transferable in principle*: techniques exist that target the teacher’s MCTS visit distribution over a window of plies rather than only the root distribution (“process distillation”). We have not implemented them.

For compact-network practitioners: (i) anticipate a data-scaling ceiling in the 100k–150k-position range for 7M architectures, and front-load Move Quality evaluation to detect it early; (ii) once the ceiling is reached, prefer architecture scale-up (channel-width doubling, or process distillation) over corpus scale-

up; and (iii) evaluate against unrelated reference engines, not only against sister distilled networks, since the AI-vs-AI style clash phenomenon makes sister-network arenas misleading.

7. Conclusion

We presented a three-generation distillation experiment showing that a 7M-parameter Go network’s top-k oracle agreement *monotonically degrades* across a 4.4× scaling of the KataGo-labeled training corpus, with no net improvement in value calibration. We argued that the result is structural: the teacher’s strength has a search-depth component that single-position labels do not transfer, in contrast to LLM distillation where teacher and student operate at the same single-step level. A complementary five-intervention study confirmed that direct value-head capacity, β_z target-blending, score-head re-activation, and direct loss-weight scaling each fail to break the ceiling; only auxiliary multi-task balance (ownership-weight tuning) produced measurable improvement.

Distillation is not a uniform technique. What it transfers, and what it leaves on the table, depends on what the teacher’s strength is made of. In Go, half of that strength lives where the distillation pipeline cannot reach. For practitioners, the path forward past the ceiling is architecture scale, MCTS-aware process distillation, or hybrid distillation–RL stages — not more data of the same kind.

Declarations

Acknowledgements

The 7M-parameter student network was iteratively improved over the course of this work; the distillation training, evaluation infrastructure, and paper drafts were prepared with assistant drafting support from Claude (Anthropic).

References

1. Chiang W-L et al (2023) Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. lmsys.org blog post
2. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. NeurIPS Deep Learning Workshop
3. Kojima T (2026), in preparation Move Quality Beyond Arena: Pure Distillation for Small Go Networks
4. Preprint <https://www.researchsquare.com/article/rs-9749016/v1>
5. Silver D et al (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529:484–489
6. Silver D et al (2017) Mastering the game of Go without human knowledge. Nature 550:354–359
7. Silver D et al (2018) A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362(6419):1140–1144

8. Taori R et al (2023) Stanford Alpaca: An instruction-following LLaMA model. GitHub repository
9. Wu DJ (2019) Accelerating self-play learning in Go. arXiv:1902.10565
10. Xu C et al (2023) WizardLM: Empowering large language models to follow complex instructions. arXiv:2304.12244

Figures

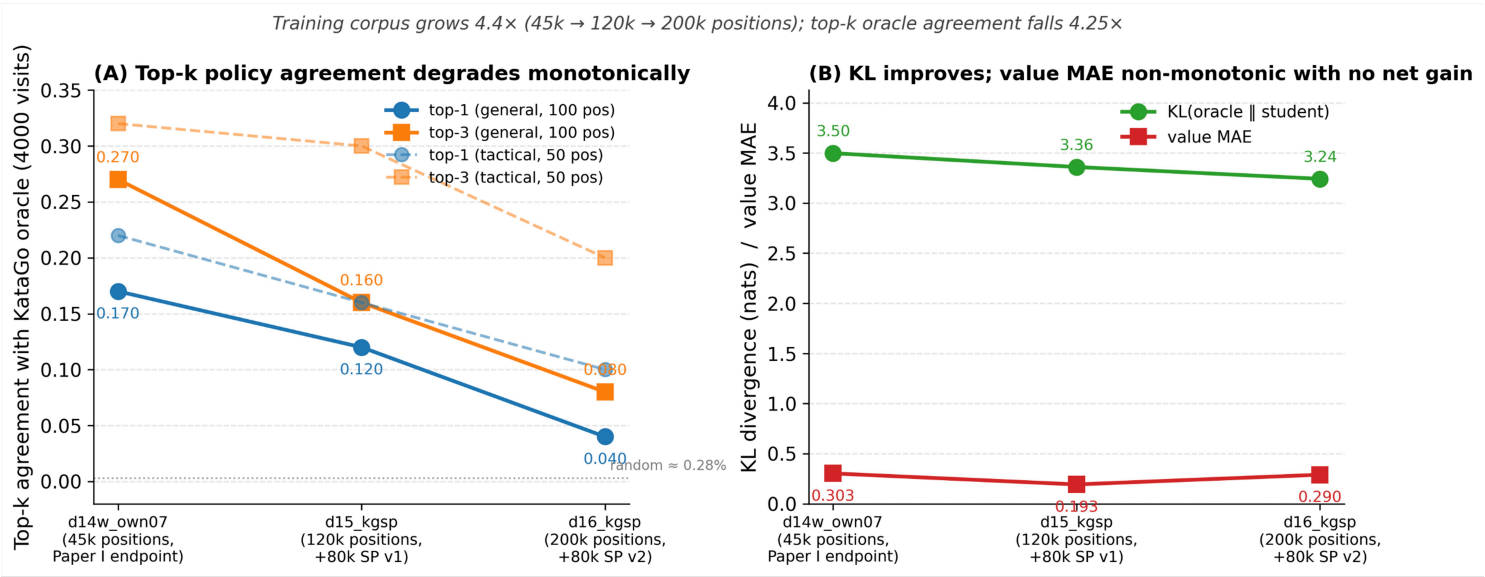


Figure 1

Three-generation Move Quality results. Panel A: top-k oracle agreement on both the general (solid, n = 100) and tactical (dashed, n = 50) evaluation sets, plotted against the three distillation generations; all four curves decline monotonically. Panel B: KL divergence between student and oracle policy (green) improves monotonically across the three generations, while value MAE (red) is non-monotonic and shows no net gain from the 8× corpus increase between d14w_own07 and d16_kgsp.