OUTPUT FORMAT AND INDEX LOOKUP DEGRADE IN-CONTEXT TOOL SELECTION AT LARGE CATALOG SIZES

Anonymous authors
Paper under double-blind review

ABSTRACT

When an LLM selects a tool from a large in-context catalog, how much of its degradation with catalog size is due to output format and index lookup, as opposed to selection itself? Across Qwen2.5 (0.5B–14B, 4-bit) on the Berkeley Function-Calling Leaderboard (BFCL), a zero-learning lexical-overlap heuristic overtakes in-context LLM selection at a finite, measurable catalog size $|V|^*$. This crossover grows with scale (8.7 at 1.5B to 500.0 at 14B; $\alpha = 1.784$), then more slowly beyond 14B (365.3 at 32B, upper bound resolved rather than right-censored; 552.6 at 72B, 95% CI $[346.6, >650]$, upper bound right-censored — both precision-matched, App. B.24). Two frontier closed models (GPT-4o, Claude Sonnet 5, small-n) show no crossover at all through $|V|$=650 (App. B.26, B.27), limiting practical generality.

Forcing name-based rather than index-based output recovers a substantial share of the lost performance at high $|V|$ (3B: $+37.5$pp, 7B: $+41.5$pp, both at $|V|$=234; 3B peaks at $+50.5$pp at $|V|$=64), but performance still declines with $|V|$ under name emission, so selection difficulty and output-format cost are not cleanly separated — a direct log-likelihood control that removes output format entirely does not reveal an easier underlying selection ability, if anything the reverse at low-to-moderate $|V|$ (App. B.21). Extending the grid to $|V|$=650 resolves 3B's own name-emission crossover ($|V|^*$=350, $12.2\times$ index-emission's point estimate, CI $[9.4\times, 15.1\times]$); 7B stays right-censored, so its ratio is an upper bound only, since a hashed-name control could not cleanly rule out lexical-shortcut copying as the source of the gain. A further test removes the task entirely, asking only whether the model can locate a *named, given* answer's index: accuracy still declines significantly with $|V|$ at both scales (3B: -70.0pp; 7B: -41.5pp), though at 3B the task is counterintuitively *harder* than ordinary selection at $|V| \leq 16$ — near-miss counting fits well (74.2%, $2.6\times$ chance, at 8; 41.0%, $3.1\times$ chance, at 16), not a bug. Two further controls narrow the picture: $|V|$, not raw prompt length, is the primary driver (partially so at 3B), and the near-miss pattern is not specific to one binding direction; we do not claim a fully clean causal isolation of this mechanism.

A second mitigation, filtering the catalog before selection, closes most of the gap, and is effective across a range of k rather than only at a model's own crossover (App. B.9). Most of this gain is achievable through candidate ranking alone, without truncation — a cheaper, more directly deployable finding, interacting with the output-format effect (§4.5). Every $|V|^*$ reported here is specific to lexically-transparent catalogs: a direct opacity check collapses the crossover roughly $3\times$ (App. B.7), and the overall finding replicates directionally on an independent benchmark and model family. We report every number here, including the ones that complicate the story, as measured.

1 INTRODUCTION

Agent benchmarks report a single number — task success rate — treated as a measurement of model capability. It rarely is one, cleanly. Any environment exposing a *valid action set* V_t at each decision

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

point (a tool catalog, an admissible-command list, a candidate move set) does part of the task's work before the model generates a single token: selecting uniformly at random from V_t succeeds at rate $1/|V_t|$, with zero learning. As $|V_t|$ shrinks, that floor rises; as it grows, it falls. Reported success rates conflate this floor with whatever the model adds on top, and no current agent benchmark reports the floor.

We ask a narrow, answerable question: *holding a real tool-calling task fixed, how does model success compare to a zero-learning control matched to the same valid-action set, as that set's size varies — and does that comparison change with model scale?*

We answer it with a structure-matched control: a lexical-overlap heuristic that scores each candidate function by token overlap with the query and picks the top match. It has no parameters and no training signal, but is not a random baseline: it exploits the fact BFCL-style catalogs are written so function names/descriptions are lexically close to their queries — the same property any retrieval-based tool router exploits. Its win over an LLM at large $|V|$ should be read as "an untrained retriever beats the model," narrower than "chance beats reasoning," and we frame the contribution accordingly throughout. We run this heuristic, and five Qwen2.5 sizes (0.5B/1.5B/3B/7B/14B, 4-bit), against real BFCL cases at seven $|V|$ levels from native (2–4) to 234 (650 for 14B), holding case, ground truth, prompt template, and seed fixed across arms.

The result. The heuristic overtakes every open-weight model tested at a finite, measurable $|V|^*$ that grows with scale but does not diverge within the range tested ($|V|^* \propto N^{1.78}$; §4.2). Two downstream findings matter more for practitioners: forcing name-based rather than index-based output recovers much of the lost performance at high $|V|$ (§4.6), and most of a retrieval-filter's benefit is achievable through ranking candidates alone, without truncation (§4.5) — both cheaper to deploy than the crossover measurement alone would suggest is necessary. We also test the crossover's mechanism directly, removing reasoning from the task entirely (Contribution 1 below; §4.7).

What this paper is not. Not a claim that LLMs cannot select functions, nor a leaderboard comparison (selection only, never argument binding — §3); not a general theory of agent competence — the chance-correction identity is elementary, used only as measurement; and not primarily a multi-step paper — extending to ALFWorld found no effect at any affordable budget, quantified in App. D rather than presented as an underpowered null finding.

Contributions. **(1)** A test of *why* the crossover occurs, with reasoning removed: accuracy still declines significantly with $|V|$ at both scales even when told the name and asked only for its index — informative for a readout/binding account, though not fully clean, converging with an index-vs-name emission ablation and an attention-readout probe (§4.7), and scoring lower under a log-likelihood control (App. B.21). **(2)** A structure-matched, zero-learning control for tool selection, pre-registered before any run (App. B). **(3)** The first measured crossover $|V|^*$ between a heuristic and an LLM across model scale, with a fitted growth exponent, refined by a higher-power re-measurement (§4). **(4)** Cross-family checks at matched scale in two other model families, including an anomaly investigated rather than discarded (§4.3). **(5)** A parameter-free dilution control with no fitted parameters, and a pre-registered hypothesis reported as falsified (§4). **(6)** A second, independently-built benchmark (Seal-Tools (Wu et al., 2024)): consistent at all five scales, with both practitioner mitigations (name-emission, H9) replicating too — evidence against a single-catalog artifact (App. B.3–B.6). **(7)** A pre-registered test of retrieval-then-select (H9, §4.5): filtering broadly, not only at a model's own crossover, closes most of the gap, confirmed across three scales and two precisions; follow-up analysis identifies most of this gain as achievable through ranking survivors alone, without truncation — arguably the paper's most directly deployable finding (§4.5).

2 Related Work

Empirical precedents. Table 1 summarizes the closest empirical precedents and how this paper differs from each.

None of these reaches this paper's sequential, executed, valid-set manipulation across model scale.

Answer-format sensitivity in multiple-choice evaluation. A separate literature on standard multiple-choice benchmarks finds that the symbol or position used to reference an answer is not a neutral labeling choice — poor binding to answer symbols, content-independent option-ID bias, and sensitivity to presentation order alike (Robinson & Wingate, 2023; Zheng et al., 2023; Pezeshkpour

Table 1: Closest empirical precedents and how this paper differs. Five further concurrent papers are differentiated in full in App. C.

Prior work	What they found	How this paper differs
Bits-over-Random (Repantis et al., 2026b)	Analytic chance-corrected selectivity measure; small catalogs vanish selectivity even for a perfect selector	We execute the control under one harness, sweep $\|V\|$, and fit a scaling exponent
Chen (2026)	Correct tool gets 80% attention even on failures; bottleneck is readout, not the harness	We test this directly with our own attention probe (§6), not only cite it
Cho (2026)	Harness sensitivity is non-monotone across capability tiers	Our parse-failure/selection split isolates exactly this confound
Chen et al. (2026)	Decomposes an accuracy shift into content and format-recovery margins	Closest precedent to our confound #1; we extend with an executed, cross-scale sweep

& Hruschka, 2023). Our §4.6 and §4.7 results extend this to a much larger option count (up to $|V|$=234 vs. typically 4–5), to tool selection rather than knowledge QA, with a controlled, though not fully isolated, intervention.

Yauney & Mimno (2024) is an earlier precedent for "the random baseline is stronger than assumed" in in-context learning, not agents.

Relationship to sparse retrieval and tool retrieval more broadly. `heuristic_valid`'s score is unweighted token-intersection, omitting BM25/TF-IDF's IDF term; H10 (our dense-retrieval comparison, App. B.2) widens rather than narrows the gap — suggestive, not a substitute. H9's retrieval-then-select prescription (§4.5) sits in the broader tool-retrieval literature treating shortlisting as a first-class step; we contribute a measured account of how filter depth trades off against selection accuracy across a range of k, not a new retriever.

Relationship to long-context degradation. A separate literature studies context-length-driven degradation independent of tool selection, notably Liu et al. (2024)'s finding that models use mid-context information less reliably. Our $|V|$-sweep necessarily increases prompt length alongside candidate count, so some measured decay could reflect general long-context degradation rather than tool-selection-specific difficulty. The padded-length control (§4.7, App. B.15) speaks to this for the oracle-lookup test; an equivalent control for the main $|V|$-sweep itself is left for future work.

Why Seal-Tools, not ToolBench or APIBench. Neither gives the independent-construction check Seal-Tools (Wu et al., 2024) does: APIBench shares BFCL's Gorilla lineage (Patil et al., 2023), and ToolBench's real-RapidAPI names plausibly share its lexical transparency rather than testing against it; extending to either remains a natural next step, not one we have evidence against.

3 METHOD

Task and scope. We use real function-calling cases from BFCL v4 (Patil et al., 2023) (`multiple` and `live_multiple` splits; native mean $|V|$ = 2.8–4.0). We score *function selection only* — which candidate function is named — never argument binding, and our numbers are never comparable to the BFCL leaderboard, which scores both.

Arms. Five arms see the identical candidate set per case: **`constrained_llm`** (the model, forced to emit an index into the shown candidates); **`heuristic_valid`** (zero-learning lexical-overlap baseline, same candidates); **`random_valid`** (uniform random over the same candidates — the chance floor); **`random_unconstrained`** (uniform random over the full function vocabulary, $N \approx 1{,}222$); and **`unconstrained_llm`** (the model asked to name a function with no candidates shown). All arms are paired: every arm sees the same case, seed, and prompt template modulo the manipulation itself. `heuristic_valid` scores each candidate by raw token-intersection count between query and name+description (regex `[a-z0-9]+`, lowercased, no normalization), ties broken by the case's seeded RNG.

The $|V|$ sweep. We hold every case's ground-truth function fixed and inject distractor functions drawn from a pool of 1,222 real BFCL functions to construct candidate sets at seven levels: native, 8, 16, 32, 64, 128, 234. The ground-truth function's position within the shown candidate list

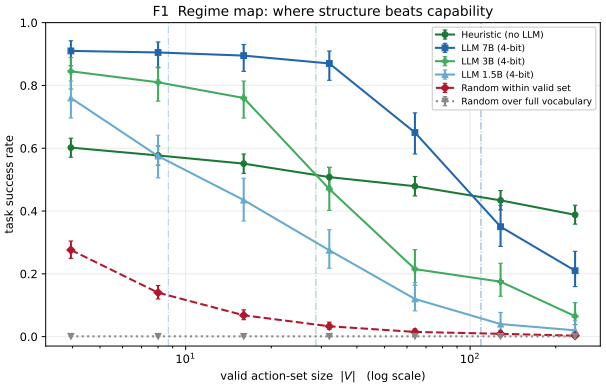


Figure 1: Regime map: task success vs. valid action-set size $|V|$ for the heuristic and each Qwen2.5 scale (4-bit, precision-matched), plus the two random controls. Vertical lines mark the measured crossover $|V|^*$ for each LLM scale.

Table 2: Crossover point $|V|^*$ by model size, all 4-bit. †14B's grid extended to 650 (native–234 right-censored there); see §4.2. 9.1% of bootstrap resamples are themselves right-censored at 650 and excluded from the CI below, so its upper bound is itself right-censored, not a resolved estimate (App. B.24). ‡Left-censored: heuristic already wins at native $|V|$.

Model	$\|V\|^*$	95% CI
0.5B	≤ 3.95	—‡
1.5B	8.7	[4.1, 20.2]
3B	28.7	[19.5, 43.4]
7B	109.2	[73.6, 185.3]
14B	500.0	[234.0, >650]†

is randomized (seeded, per case and seed) after assembly, not fixed at a particular slot. 40 base cases: uniform random sample (seeded, no stratification); distractors: sampled without replacement, excluding functions already in the native set (excludes exact-name collisions by construction; near-duplicate distractors remain possible and are unscreened, quantified post-hoc in App. B.25). Each Qwen-family run uses 40 cases $\times$ 5 seeds = 200 cells per arm per $|V|$ level (1,400 cells per arm across the full sweep); the dilution control and the `random_valid` $= 1/|V|$ residual check use $n = 1000$ per point. 14B's grid was extended to 350/500/650 (§4.2).

Decomposition. We define *structure* = `random_valid` − `random_unconstrained`, *learning* = `constrained_llm` − `random_valid`, and *structure_share* = *structure*/(*structure* + *learning*), which reduces to `constrained_llm` − `random_unconstrained` and is therefore unreliable near large $|V|$, where that denominator approaches zero (the sign flip at $|V|$=234 in §4.4 is this floor, not a substantive reversal; H2/H5 sit at native $|V|$, away from it, and H6/H8's exposure to it is flagged there). `random_valid` $= 1/|V|$ is a parameter-free prediction with residuals ≤ 0.0013 against measured values at n=1000 per undiluted level (a subset of Figure 3's broader dilution sweep).

Models. Qwen2.5-Instruct at 0.5B, 1.5B, 3B, 7B, and 14B, all loaded at 4-bit (`nf4`, `bnb_4bit_compute_dtype=float16`) to hold numerical precision fixed across the scale axis (Qwen Team, 2025). An architecturally independent model (Phi-3.5-mini-instruct, ∼3.8B) is included as a same-scale cross-family check — see §4.3.

Pre-registration. Hypotheses H1–H8, the $|V|$ grid, the arm set, and the stopping rules (seeds ≥ 5, red-flag conditions on parse-failure rate and resolution floor) were fixed before any model was run (App. B). Four later hypotheses (H9–H12, added across three amendments — H11 and H12 share Amendment 5), each dated before the data that would have motivated them, are logged there. The 32B, 72B, and frontier-model checks (§6, App. B.22–B.27) were not pre-registered: each was added post-hoc, disclosed as such where reported rather than presented as if a rule had been in place.

4 Results

4.1 The crossover exists at every open-weight scale tested

At every model size, LLM success starts above the heuristic at small $|V|$ and crosses below it at a finite, measurable point. $|V|^*$ is log-linearly interpolated (heuristic−LLM difference, $\log|V|$ space) between the bracketing grid points where it changes sign; left-censored values (0.5B, Seal-Tools 1.5B) report the smallest tested $|V|$ since the difference is already non-negative there.

Paired differences (not the overlapping marginal CIs above) are the licensed comparison: $|V|^*_{3B} - |V|^*_{1.5B} = +20.0$, $|V|^*_{7B} - |V|^*_{1.5B} = +100.5$, both $P(\leq 0) = 0.000$ (bootstrap over cases, $n = 4000$).

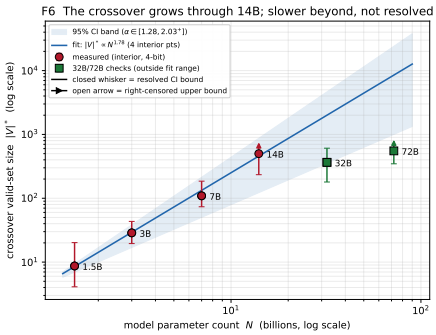


Figure 2: Crossover valid-set size $|V|^*$ vs. model parameter count N, log–log, with the fitted power law and bootstrap 95% confidence band. Fit on four interior points; report as a bounded rate, not a precision scaling law (§4.2). Green squares mark 32B and 72B, both precision-matched, local, complete grids (App. B.22, B.23). 14B's and 72B's upper bounds are right-censored (open arrow, confirmed, App. B.24); 32B's point estimate resolves cleanly and is not right-censored.

4.2 The crossover grows through 14B; slower beyond, not resolved (32B–72B)

Fitting $|V|^* \propto N^\alpha$ on the four precision-matched interior points (1.5B/3B/7B/14B): $\alpha = 1.784$, 95% CI $[1.284, > 2.027]$ (both bounds censoring-corrected, App. B.24). This is robust to how parse failures are scored — treating them as missing-at-random rather than as task failures gives $\alpha = 1.644$, 95% CI $[1.253, > 1.894]$ (upper bound right-censored too, at an even higher 12.7% exclusion rate under this convention); both conventions exclude 1 more decisively than the three-point fit did. We report the range $\alpha \approx 1.6$–1.8 as a **bounded rate, not a precise scaling law**: four interior points and a CI spanning $[1.3, 2.0]$ narrows the three-point fit's $[1.2, 2.1]$ (width $0.860 \to 0.743$) and rejects linearity more decisively, but still not to a precision we would extrapolate from; we do not extrapolate outside 0.5B–14B (§7). **Leave-one-out sensitivity**: dropping any single one of the four points and refitting gives $\alpha \in [1.640, 1.845]$ — comfortably inside the reported CI, no single point driving the estimate (dropping 14B exactly reproduces the original three-point fit, 1.640, as expected). 14B was right-censored at native–234 ($|V|^* > 234$), so the grid was extended to 350/500/650 on the same held-out cases to bracket it — **both arms were measured at every extended level**, not extrapolated; the 14B crossover is a matched pair at 350/500/650, the same as every other model's crossover is from native–234 — **the one place in this paper where a decision was reactive rather than pre-registered** (no contingency rule specified what to do on right-censoring; every other grid, arm, and hypothesis decision was fixed in advance). The dilution control (Figure 3, §4.4) validates harness correctness, not the LLM-vs-heuristic comparison itself.

Quantization shifts the crossover at 1.5B, not detectably at 3B (fp16 vs. 4-bit; full comparison in App. B.28): the qualitative finding — a finite crossover at every scale and precision tested — is unaffected either way, but we do not know whether α itself would shift under fp16 and report the comparison to that extent only (§7).

Every model's decay slope with $|V|$ is far steeper than the heuristic's (all $P(\geq 0) = 0.000$); scale does not flatten this decay (1.5B/3B slopes are statistically indistinguishable) — every open-weight LLM tested decays several times faster than the heuristic at every scale tested, so scale raises the curve's *level*, not reliably its *slope*.

4.3 Cross-family checks (Phi-3.5-mini, Mistral-7B)

To test whether α is a property of scale or of Qwen's post-training specifically, we ran the identical protocol on two architecturally unrelated models at matched parameter counts.

Phi-3.5-mini-instruct (~3.8B, 4-bit). Pre-registered $|V|^*(3.8\text{B}) \approx 41$, band 25–65, from the Qwen-only fit before running. **Measured:** $|V|^* = 53.0$**, 95% CI** $[32.0, 113.9]$ — inside the pre-registered band, consistent with the crossover generalizing beyond one model family, though a single additional family remains one data point, not a statistical replication (§5). **A second finding: the failure mode past the crossover is not always smooth.** At $|V| = 128$, Phi-3.5's success rate collapses to 0.060 (vs. a neighboring 0.405 at $|V| = 64$) — a genuine, content-level collapse, not a

parsing failure (parse_ok=100%). A pre-registered LongRoPE switch-point test returns a near-miss, not a confirmation (full test and tolerance: App. B.29); the full cause is not identified. We exclude $|V|$=128 from the crossover above (which resolves at lower $|V|$, unaffected) and do not extend the decay-rate comparison to this model's full curve.

Mistral-7B-Instruct-v0.3 (4-bit), at an exactly-matched scale. Since 7B is already a Qwen point (Table 2), no extrapolation is needed. **Pre-registered**: within $2\times$ of Qwen-7B's $|V|^*$=109.2 ($[55, 220]$) supports scale-generality; outside is disconfirming. **Data-integrity note.** All 200 constrained_llm cells at $|V|$=234 OOM'd on the T4 used — a hardware limit specific to this model (no Qwen scale, 0.5B–14B, OOM'd there; harness and quantization code are byte-identical across every run, only MODEL_ID differs, so this is not a misconfiguration; architecture comparison in App. B.30). Caught per-cell (§7); the raw 0.000 is an OOM artifact, treated as missing, not failed, below. **Measured (OOM cells excluded):** $|V|^* = 56.9$**, 95% CI** $[34.2, 115.9]$ (n=4000; the correction narrows the CI's upper bound from an inflated 143.1, doesn't move the point estimate). Falls inside the pre-registered band, close to its lower edge — real but not a clean confirmation, using the identical distractor pool and prompt template as every Qwen run (§3), with no per-family adaptation beyond the Phi-3.5 $|V|$=128 anomaly. With Phi-3.5-mini, this gives two independent families at two scales (3.8B, 7B) both landing inside their pre-registered bands — a small-n pattern, two points, not a statistical replication.

4.4 Further confirmatory checks

A second, independent control validates the harness. Diluting the valid set with j additional distractors degrades random_valid success exactly as $1/(|V|+j)$ predicts, with no fitted parameters (34/35 points inside the 95% CI; figure: App. B.1). Since random_valid does not process candidate text, this validates the sampling/scoring harness itself, not any BFCL-specific textual artifact — separately addressed by Seal-Tools (App. B.3) and the name-opacity ablations (App. B.7).

A pre-registered hypothesis: falsified on BFCL. We pre-registered $structure_share \geq 0.5$ (H2) on ALFWorld; the structure term there was statistically indistinguishable from zero for reasons quantified in App. D (underpowered, not falsified). The same quantity, measured on BFCL where the design has power, is 0.353, 95% CI $[0.266, 0.446]$, excluding 0.5: the strong form (structure dominates) is false where testable; structure as a measurable, non-vanishing floor is what this paper argues.

Confound checks. unconstrained_llm never lands in-vocabulary (0/5,600 cells — the expected floor, not a parsing artifact), so it carries no signal and is excluded from the pairwise-comparison family. Non-LLM arms are bit-identical across every shared run (0 mismatches, ruling out silent harness drift); under the conservative convention treating parse failures as missing rather than task failures, 86.1% of the 4-bit success drop is attributable to selection, not formatting (full reasoning: App. B.30).

4.5 Filtering closes the gap: retrieval-then-select (H9)

§6 predicts retrieval-then-select should close most of the gap between an LLM's unfiltered performance and the heuristic's, once effective $|V|$ is filtered below the model's own $|V|^*$. We pre-registered this as H9 (App. B, Amendment 3) before running it: a sixth arm, retrieval_filtered_llm, uses the same zero-parameter lexical scorer as heuristic_valid to filter to the top $k=\lceil |V|^*_{\text{4-bit}} \rceil$ (9 for 1.5B, 29 for 3B) — *without* artificially preserving the ground truth — before constrained LLM selection among survivors. Success requires both filter recall and correct selection: the end-to-end accuracy the deployment claim is actually about.

Falsification test (A3.4), at 1.5B. At $|V|$=234, 4-bit (the precision used throughout Table 2): $+0.415$, 95% CI $[0.320, 0.512]$, excluding zero. Independently confirmed at fp16: $+0.455$, CI $[0.357, 0.558]$. **H9 is supported at 1.5B, at both precisions tested.**

Falsification test, at 3B (4-bit). $+0.495$, CI $[0.400, 0.585]$, excluding zero — if anything larger than at 1.5B. Sharper dose-response: null at native/$|V|$=8, significant exactly at $|V|$=32 ($+0.265$, $p<0.001$) — the first point past 3B's own crossover (28.7) — staying large through 234 (the null below crossover is partly guaranteed by construction, filtering to $k=|V|^*$ being a no-op when $|V| \leq$

Table 3: Index-emission vs. name-emission success rate, headline $|V|$ levels (full 7-row grid, both scales, and the source figure: App. B.18). Paired bootstrap 95% CIs on the difference. * CI excludes zero.

	3B				7B			
$\|V\|$	index	name	diff	95% CI	index	name	diff	95% CI
native	0.885	0.875	-0.010	[-0.060,+0.035]	0.935	0.935	+0.000	[-0.035,+0.040]
64	0.210	0.715	+0.505*	[+0.400,+0.605]	0.680	0.830	+0.150*	[+0.060,+0.245]
234	0.095	0.470	+0.375*	[+0.280,+0.470]	0.240	0.655	+0.415*	[+0.330,+0.500]

k; §4.5 examines what the gain above crossover reflects). Recall degrades gracefully at both scales ($1.00 \rightarrow 0.675$ at 1.5B, $1.00 \rightarrow 0.785$ at 3B).

Falsification test, at 7B (4-bit, k=110). $+0.310$, CI $[0.180, 0.435]$ at $|V|$=234, excluding zero: null through $|V|$=64, jumping to $+0.220$, p<0.001 exactly at $|V|$=128 — past 7B's own crossover (109.2) — growing to $+0.310$ at 234. Recall degrades gracefully ($1.00 \rightarrow 0.915$; same construction-driven null below crossover as 3B). **H9 is now confirmed at all three scales in the primary sweep** (1.5B, 3B, 7B) and two precisions at 1.5B; summary status in App. B (H9 row).

Is k=$|V|^*$ the right choice, or would any k do? A k-sweep at 3B and 7B finds every tested k beats unfiltered selection by a large, significant margin, but k=$|V|^*$ is not shown to beat a simple fixed k=5 baseline at either scale. We do not claim k=$|V|^*$ optimal — filtering helps broadly, not only at one cherry-picked value (App. B.9).

H9's gain is substantially, though not entirely, a ranking effect — and this connects directly to §4.6. Ranked order of the identical filtered set significantly beats shuffled order at both scales (App. B.10). A corrected 2×2 (an earlier attempt had a flawed, non-genuinely-ranked design, full disclosure App. B.11) confirms a real, cross-scale interaction with §4.6's emission-format result: index emission specifically needs position-binding that name emission does not, so shuffling costs index emission far more than name emission at both scales (App. B.11). We report this as a robust, cross-scale finding, not a single-scale curiosity.

A cost proxy, not a measured latency benchmark. We did not instrument wall-clock timing; filtering reduces the candidate count shown to the LLM at $|V|$=234 (7B) to a mean of 110 (53% fewer), which proxies token and compute savings without substituting for a measured latency/cost table (full reasoning in App. B.31).

4.6 Output format: index vs. name emission

Every result above requires the model to answer with a numeric *index* into the shown candidate list — how much of the measured crossover depends on this output format specifically, rather than on selection difficulty alone?

Design. 3B and 7B, full primary-sweep grid (native through 234), same harness, prompts, and `MASTER_SEED` as every other leg: 40 cases × 5 seeds. A new arm, `name_constrained_llm`, shows the model the identical numbered candidate list `constrained_llm` sees, but asks for the candidate's *name* instead of its index, validated against the shown candidates specifically — a direct, apples-to-apples comparison, not confounded with the unconstrained arm (validated against the whole pool under no restriction). This is a fresh, independent rerun of `constrained_llm`, not reused from the primary sweep: at native $|V|$ it gives 0.885 (3B) and 0.935 (7B), vs. 0.845/0.910 in the primary sweep (Table 5) — a 2–4pp difference at n=40, a useful run-to-run noise estimate at this sample size rather than an inconsistency.

A large, significant, two-scale effect — but not a full explanation. At 3B, the name-emission advantage becomes significant at $|V| \geq 8$, peaks at $+50.5$pp ($|V|$=64), and narrows to $+37.5$pp by $|V|$=234 — the effect does not grow monotonically across the full grid. At 7B, significance begins later ($|V| \geq 64$) but grows monotonically to $+41.5$pp by $|V|$=234. At $|V|$=234, name-emission is still ahead of `heuristic_valid` (0.385) at both scales (3B: 0.470; 7B: 0.655) — not yet crossed within the primary $|V|$=234 grid (extended below), versus index-emission's reported $|V|^*$=28.7 (3B) and 109.2 (7B). This is real and substantial, but under name emission performance still declines with $|V|$ at both scales (3B: $0.885 \rightarrow 0.470$; 7B: $0.935 \rightarrow 0.655$), so selection difficulty and output-format cost both contribute here — we do not claim to have cleanly separated them,

and note that name emission also lets the model copy text lexically close to the query, which the heuristic itself exploits. A hashed-name control (opaque 12-character identifiers replacing names, as in H11) collapses toward index-emission at each model's highest tested level (3B: diff -1.0pp, CI $[-6.0, +4.0]$; 7B: diff -8.5pp, CI $[-19.5, +2.0]$; neither excludes zero), but does not cleanly resolve lexical-shortcut vs. format-difficulty: the model mostly abandons the hash format under difficulty rather than making near-miss copy errors on it (full 7-row grid, figure, and detail: App. B.18, App. B.20).

The name-emission crossover, extended. Extending the grid to $|V|$=650 (against `heuristic_valid`'s non-flat curve there, 0.355/0.330/0.345), 3B's crossover is measurable (case-level bootstrap, same method as elsewhere, name-emission arm): $|V|^*_{\text{name}}(\text{3B}) = 350$, 95% CI $[271, 432]$ — $12.2\times$ index-emission's own 28.7 (CI $[9.4\times, 15.1\times]$) — 3B's per-level values at $350/500/650$ are tabulated in App. B.18. 7B remains right-censored through 650 (0.580/0.505/0.440), consistent with its own index-emission crossover also growing with scale. Given the hashed-name result above, read this as an upper bound for lexically-transparent catalogs, not a guaranteed multiplier.

The same pattern holds at 32B and 72B (all CIs exclude zero): name-emission stays above `heuristic_valid` through $|V|$=650 at both (32B $+31.0$pp$\rightarrow$ $+13.0$pp; 72B $+36.0$pp$\rightarrow$ $+18.0$pp), narrowing but never reversing. Only 3B, of four scales tested, shows a measured crossover — 7B, 32B, and 72B are all right-censored through 650 (App. B.19).

4.7 A CAUSAL TEST: INDEX BINDING WITH THE TASK REMOVED

The emission-format result above compares two output formats; it does not isolate *why* index emission costs so much. The following intervention removes reasoning from the task entirely to test the mechanism directly.

Design. The model is shown the candidate list, exactly as in `constrained_llm`, but is also told the correct answer's name explicitly and asked only for that name's index — no query, no task, no selection required. A matched control (`oracle_lookup_shuffled`) uses an identical candidate set in shuffled order, testing whether any position effect appears even with the answer given directly. An earlier forced two-step name-then-index design produced an internally inconsistent result (App. B.12) and was discarded in favor of this simpler, single-call design, which removes every confound the earlier one had.

A real decline with $|V|$, but not a clean isolation of binding. 3B falls from 92.0% at native $|V|$ to 22.0% at $|V|$=234 (native-minus-234 $+70.0$pp, 95% CI $[61.5, 77.5]$); 7B falls from 96.0% to 54.5% ($+41.5$pp, CI $[34.0, 49.0]$) — full per-level numbers, both scales, and the source figure: App. B.13. Both declines are smooth and monotonic across every intermediate $|V|$ tested (Table 12), not the implausible flat pattern the earlier, discarded design produced (App. B.12). Two further controls clarify, though do not fully resolve, what this decline reflects.

Length control (full detail: App. B.15). Padding a fixed, low $|V|$=8 prompt to $|V|$=234's length shows $|V|$, not raw prompt length, is the primary driver: no detectable length effect at 7B (padded-8 vs. unpadded-8, CI $[-3.5, +2.5]$, includes zero), only a partial one at 3B (diff -19.0pp, CI $[-28.5, -10.0]$) that does not overturn $|V|$'s dominance — padded-low-$|V|$ remains $+35.5$pp above unpadded-234, CI $[+25.0, +45.5]$.

Reverse-lookup control (full detail: App. B.16). Asking the opposite direction (given the index, name it) shows the 3B low-$|V|$ anomaly is not specific to name-to-index binding: reverse lookup also scores below ordinary selection at $|V|$=8, 16 (gaps of 21.0pp and 16.5pp), comparable to forward lookup's own gaps there — suggesting a broader property of given-answer binding at this scale, not a flaw isolated to the original test. A Holm–Bonferroni-corrected, per-level comparison (same convention as H12, App. B.7; Table 13) finds the sign itself changes with $|V|$ at 3B rather than a uniform effect, and a consistently negative but only partly significant direction at 7B.

Taken together, both controls narrow — without fully resolving — what the decline reflects: $|V|$ dominates raw length at both scales, and the anomaly looks like a broader property of given-answer binding at 3B rather than a design flaw. A further diagnostic against data already collected (App. B.17): the correct answer sits at the model's emitted index ± 1 in 74.2%/41.0% of $|V|$=8, 16 failures ($2.6\times$ /$3.1\times$ chance) — not an indexing bug, but a near-miss counting error, costliest with

few positions. A case-level factorization check (name-selection × lookup success predicting index-selection success, per case) fits well at $|V|\geq 64$ (3B, $|V|=234$: predicted 0.089 vs. actual 0.095) but underpredicts at low $|V|$ (3B, $|V|=16$: predicted 0.552 vs. actual 0.805), consistent with the same pattern. Locating a named, given item's index degrades robustly with $|V|$ at both scales, now characterized rather than unexplained, though not a clean, fully-isolated causal mechanism. The oracle task's own shuffled control shares App. B.11's non-ranking design issue (full disclosure in App. B.13).

Two further checks address generalization, pre-registered (App. B.7, App. B.3). Replacing tool names with an opaque hash collapses the crossover roughly $3\times$ ($|V|^* = 8.6$ vs. 28.7 transparent) — the LLM loses far more than the heuristic from name-opacity; removing descriptions too collapses both arms to the same non-result. A non-hash control measures $|V|^* = 22.8$, mostly copy-difficulty (App. B.7). On an independently-constructed catalog, Seal-Tools, 3B (21.4) and 7B (80.1) overlap BFCL's own (28.7, 109.2); 1.5B stays left-censored, direction-consistent.

5 Robustness Summary

$\alpha \approx 1.6$–1.8 is supported by five checks: parse-failure-convention robustness (§4.2), the dilution control (§4.4), the left-censored 0.5B anchor (Table 2), two cross-family checks inside pre-registered bands (§4.3), and Seal-Tools replication at 3B and 7B, including its own α-fit (1.560, App. B.4). **On multiple comparisons**: no single FDR correction spans H1–H12; Holm-Bonferroni applies within H12's exchangeable family (App. B.7).

6 Discussion: why the crossover matters for deployed systems

What the evidence supports, and what it does not yet. Four results bear on the mechanism, at different strengths. (1) An attention-readout probe: among 61 `constrained_llm` failures at 3B, $|V|=64$, the correct candidate's mean attention rank sits at the 11th percentile (vs. 50th by chance) — the model usually attends to the right answer even when it errs, but this is correlational, not causal. (2) An index-vs-name emission ablation (§4.6) — paired at 3B/7B, name-vs-heuristic only at 32B/72B (App. B.19) — is real but not fully clean: name-based output recovers much of the lost performance at high $|V|$, but a hashed-name control (App. B.20) cannot cleanly separate a lexical-shortcut explanation from a genuine format-difficulty one. (3) An oracle name-to-index lookup removes the task entirely; accuracy still declines significantly with $|V|$ at both scales, though not cleanly, and at 3B is counterintuitively *worse* than ordinary selection at $|V|\leq 16$ — near-miss counting, not a bug (App. B.17). (4) A log-likelihood control (teacher-forced scoring) scores *below* `constrained_llm` at low-moderate $|V|$, converging only near floor (App. B.21). We weight these accordingly: (2) is suggestive of a real format cost, not cleanly isolated from a lexical-shortcut confound, and (4) complicates it further; (1) and (3) are likewise suggestive, not a clean causal isolation of a "binding mechanism" (App. B.13). Pre-filtering (H9) and name-based reporting mitigate this; a corrected 2×2 (App. B.11) confirms they interact.

The crossover grows through 14B; slower beyond, not fully resolved — tested, not asserted. Naively extrapolating $\alpha \approx 1.8$ predicts a 72B crossover past $|V|=8000$. Both 32B and 72B are now precision-matched, complete grids (App. B.22, B.23). **Measured:** $|V|^*(32B) = 365.3$**, 95% CI** $[179.7, 608.7]$**;** $|V|^*(72B) = 552.6$**, 95% CI** $[346.6, >650]$**, right-censored.** Point estimates dip at 32B then rise at 72B; wide, overlapping CIs mean this is a point-estimate pattern, not established non-monotonicity, and the plateau-vs-slower-growth question stays open only at 72B's right-censored tail.

7 Limitations

One benchmark: BFCL, function selection only, never argument binding. **Lexical transparency**: the heuristic's edge depends on it; H10 confirms (30–37pp at 3B, App. B.2), H11 falsifies opacity robustness, $\sim 3\times$ collapse (App. B.7). **Scaling exponent**: four interior points (1.5B–14B); 32B/72B sit below the naive extrapolation, precision-matched — 32B resolved, 72B right-censored (§6). **Precision**: fp16 shifts 1.5B's crossover $\sim 2\times$, barely 7B's; α's own fp16 shift unresolved (App. B.28). **Seal-Tools** generalizes beyond BFCL; frontier models show none through 650 (App. B.26, B.27). **H9 confirmed** (1.5B/3B/7B), multi-step failed (App. D).

AI-USE DISCLOSURE

Per ICLR 2027 policy (cited in Appendix A), this section is included in Appendix A (LLM usage sections may appear in the appendix without counting toward the page limit).

REPRODUCIBILITY STATEMENT

Raw per-case CSVs for every experiment cited in this paper (E1, E3, E4, E5, E6, E7, E9, E41, E41b, E42, E43, E44, E45, and the ALFWorld runs referenced in Appendix D), the canonical Jupyter notebooks that produced them (edited only at the designated configuration cell between runs, per the project's own convention — see supplementary material), the analysis scripts that compute every number in this paper from those raw CSVs, and the pre-registration document with its dated amendments, are included as supplementary material. All headline numbers were re-derived from raw data in a separate verification pass before inclusion in this manuscript (by the same AI system that wrote the original analysis code, not an independent system or human — full caveat in App. A); the $|V| = 128$ anomaly diagnostic (§4.3) is included as a standalone, CPU-only, model-weights-free notebook that reconstructs the exact prompts evaluated and can be re-run in under a minute. Model weights (Qwen2.5 family, Phi-3.5-mini-instruct, Mistral-7B-Instruct-v0.3) and BFCL/ ALFWorld/Seal-Tools task data are not redistributed; both are pulled from their public sources at run time by the included notebooks. Both source licenses were verified directly on 2026-08-31: BFCL/Gorilla is released under Apache 2.0[1], and ALFWorld is released under the MIT License[2]; both permit the redistribution used here, which in any case is limited to derived per-case CSVs of model outputs and configuration (§3), not raw task data. All code and analysis scripts referenced above are included in the supplementary material; no separate public repository is used for this submission. The standalone $|V|$=128 diagnostic notebook described above, which needs no model weights and re-runs in under a minute, is additionally available at an anonymized link (`https://anonymous.4open.science/r/bfcl-diagnostic-notebook-3200/`), alongside a second standalone notebook that reconstructs Table 2's four crossover points, the α fit, and one H9 effect size (7B) directly from bundled raw per-case CSVs, using code written fresh for this purpose — not the original harness's own analysis functions — so reviewers can independently check these specific headline numbers themselves, with code short enough to read in full, rather than trusting a self-report. This is not a substitute for a fully independent human or separate-system re-derivation of the entire paper (§A discusses this limitation directly); it is a scoped, concrete tool for the reviewer to use.

REFERENCES

Mingguang Chen, Bo Qu, and Licheng Wang. The calibration floor: Format repair can masquerade as self-correction at small-to-mid scale, 2026. Full given names verified 2026-09-15 against arXiv:2608.04355 directly.

Shiyang Chen. Looking is not picking: An attention-segment account of tool-selection failures in llm agents. *arXiv preprint arXiv:2606.16364*, 2026.

Yong-eun Cho. It's not the capability: Harness sensitivity is non-monotone across llm agent tiers, 2026. Sole author verified 2026-08-30 against arXiv:2605.26731 (KailosLab).

Arunabh Dastidar. Where does agent reliability come from? a cross-benchmark decomposition of verification loops, specialist models, and scaffolding in a production enterprise agent, 2026. "for the Leni Team." Sole author verified 2026-08-30 against arXiv:2607.17044.

Ananda Dhakal, Krish Neupane, and Aarjan Chaudhary. Baselines before architecture, 2026. Full given names verified 2026-09-15 against arXiv:2607.13085 directly.

[1] `https://github.com/ShishirPatil/gorilla`, README: "All the leaderboard statistics, and data used to train the models are released under Apache 2.0"; confirmed on the corresponding HuggingFace dataset card, `https://huggingface.co/datasets/gorilla-llm/Berkeley-Function-Calling-Leaderboard`.

[2] `https://github.com/alfworld/alfworld/blob/master/setup.py` and `https://pypi.org/project/alfworld/`, both citing the MIT License, Copyright (c) 2020 Mohit Shridhar and others.

Yonatan Gideoni, Sebastian Risi, and Yarin Gal. Simple baselines are competitive with code evolution, 2026. Full given names verified 2026-09-15 against arXiv:2602.16805 and DBLP directly.

Sungwoo Jung and Seonil Son. How much heavy lifting can an agent harness do?: Measuring the llm's residual role in a planning agent, 2026. AgentSkills 2026 Poster. Full given names verified 2026-09-15 against arXiv:2604.07236 directly.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 2024.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023. Berkeley Function-Calling Leaderboard, `https://gorilla.cs.berkeley.edu/leaderboard.html`.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

Qwen Team. Qwen2.5 technical report, 2025.

Vyzantinos Repantis, Ameya Gawde, Harshvardhan Singh, and Joey Blackwell II. How many tools should an llm agent see? a chance-corrected answer, 2026a. Meta Platforms Inc. Author list and headline downstream-validation figures (93.1% vs. 87.1% at fixed depth 5 on BFCL, using Claude Sonnet 4.6) verified 2026-09-01 against arXiv:2605.24660v2. A distinct paper from Repantis et al. (2026b) (different author set, different submission), sharing a lead author.

Vyzantinos Repantis, Harshvardhan Singh, Tony Joseph, Cien Zhang, Akash Vishwakarma, Svetlana Karslioglu, Michael Wyatt Thot, and Ameya Gawde. The 99% success paradox: When near-perfect retrieval equals random selection, 2026b. ICLR 2026 Blog Track. Author list verified 2026-08-30 against arXiv:2605.18857 (all 8 authors, Meta Platforms Inc.).

Joshua Robinson and David Wingate. Leveraging large language models for multiple choice question answering. *International Conference on Learning Representations*, 2023.

Kuan Wang. Memdelta: Controlled baselines and hidden confounds in agent memory evaluation, 2026. Sole author verified 2026-08-30 against arXiv:2606.29914.

Mengsong Wu, Tong Zhu, Han Han, Chuanyuan Tan, Xiang Zhang, and Wenliang Chen. Seal-tools: Self-instruct tool learning dataset for agent tuning and detailed benchmark. *arXiv preprint arXiv:2405.08355*, 2024. Accepted at NLPCC 2024. Author list verified 2026-09-10 against arXiv:2405.08355 directly.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Z. Wang, W. Yu, Z. Tan, Y. Tian, G. Zhao, L. Sun, X. Zhang, and T. Yang. Harness-bench: Measuring harness effects across models in realistic agent workflows, 2026. 12 authors (Peking University / Qiyuan Tech). Author list verified 2026-08-30 against arXiv:2605.27922; full given names not available for all authors on the listing, initials retained as published.

Gregory Yauney and David Mimno. Stronger random baselines for in-context learning. In *Conference on Language Modeling (COLM)*, 2024. Verified 2026-08-30 against arXiv:2404.13020, Cornell University.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*, 2023.

A AI-USE DISCLOSURE

Per ICLR 2027 policy, the AI-use statement may appear in the appendix and does not count toward the page limit[3]; this section is placed here for that reason, not to reduce its visibility.

[3] `https://iclr.cc/Conferences/2027/AuthorGuidelines` and `https://iclr.cc/Conferences/2027/AIPolicyForAuthors`, accessed 2026-09-09: "if LLMs played a significant role in research ideation and/or writing... authors should describe the precise role of the LLM in a separate section on LLM usage. This section can appear in the appendix, and will not be considered as part of the page limit."

In preparing this work, the author used a generative AI system (Claude, Anthropic) across several distinct roles, each with a different level of independent human verification applied to its output:

- **Experimental protocol and pre-registration.** The AI system participated in drafting the pre-registration document (hypotheses H1–H8, the arm set, the $|V|$ grid, and stopping rules) and its dated amendments, including a full-text prior-art check that led to restricting the primary scaling axis to a single model family (§2), Amendment 3 (H9, §4.5), and Amendment 4 (H10, App. B.2), each pre-registered before its arm was run. All hypotheses and stopping rules were fixed before any model was run; the amendment log is preserved in full in supplementary material.
- **Experiment code and analysis scripts.** The AI system wrote the canonical experiment notebooks and the analysis scripts that compute every numeric claim in this paper (crossover points, the α fit and its bootstrap CI, the dilution residuals, the parse-convention robustness check, and the H9/H10 comparisons in §4.5 and App. B.2).
- **Re-verification (same AI system, not fully independent).** Every headline number in this paper was independently re-derived from raw per-case CSVs in a separate verification pass, not simply copied from an earlier summary — including the crossover points, α under both parse-failure conventions, the H2/structure_share result, the dilution residuals, the H5–H8 figures in Appendix B, the H9 paired-bootstrap result in §4.5, and the H10 comparison in App. B.2, where this pass caught a type-comparison bug in the notebook's own summary artifact that had zeroed out six of seven $|V|$ levels (fixed; App. B.2 reports the corrected numbers), the 3B fp16 crossover, and the 14B crossover and the refit four-point α (both parse conventions) in §4.2 — the 3-point fit was re-derived first and confirmed to match the published $\alpha = 1.640$, $[1.214, 2.074]$ to three decimals before the 14B point was added, which were computed fresh for this submission rather than transcribed from process documentation. **A caveat on what "independent" means here**: this re-verification was performed by the same AI system that wrote the original experiment and analysis code, not by a separate system or a human redoing the derivation from scratch. It is independent in the sense that mattered practically — it caught real bugs the original code's own summary output had missed (the H10 type-comparison bug above is one instance) — but it is a weaker guarantee than a fully independent party re-deriving the same numbers, and should be read as such. To make this concrete rather than only disclosed, a standalone notebook (Reproducibility Statement) reconstructs Table 2's four crossover points, α, and one H9 effect size directly from raw CSVs using fresh code, so a reviewer can check these specific numbers with a tool short enough to read in full — narrower in scope than full external independence, but a concrete step past only disclosing the limitation.
- **Error identification and correction.** During the course of the project the AI system flagged and the author corrected: an initial overclaim that a bootstrap result (an earlier version of an ordering hypothesis) had been confirmed, later downgraded on re-analysis; a proposed mechanistic explanation (a "lost in the middle" account) for LLM degradation with $|V|$ that was superseded after locating closer, contradicting mechanistic evidence (Chen, 2026); a specific falsifiable hypothesis (LongRoPE context- scaling boundary, §4.3) that was tested against the model's own tokenizer and found to be a near-miss rather than a confirmation, and reported as such rather than rounded up; an original paper title and framing ("chance beats choice") that conflated the winning zero-learning heuristic with a true random baseline, corrected to the current title and the explicit framing in §1; and a precision error in the H9 run (fp16 rather than the pre-registered 4-bit) that was caught by comparing the run's `constrained_llm` curve against the paper's existing fp16 anchor point, disclosed directly in §4.5 rather than silently absorbed into the 4-bit headline numbers; a garbled, duplicated-word sentence in the Abstract (an editing artifact from an earlier revision) caught during verification and corrected; and two placeholder author-action markers (Reproducibility Statement, this section) that had persisted unresolved across several revisions, each already written with an explicit "or delete this sentence if not applicable" fallback, resolved by taking that fallback rather than leaving a visible placeholder in a submitted manuscript.
- **Manuscript drafting.** Sections of this manuscript were drafted with AI assistance and edited by the author, who takes full responsibility for all claims, framing, and remaining errors.

The author personally executed every experimental run reported in this paper: (E1, E3–E7, E9, and the H9 filtered-retrieval run) on their own Kaggle GPU compute; the 32B precision-matched calibration check and the 32B/72B name-emission extension (App. B.22, B.19) on a rented GPU (Vast.ai, A6000/A100/RTX PRO 6000, 80–96GB); a precision-matched 72B index-emission calibration check was attempted on rented GPU compute across two separate attempts and has since completed in full (App. B.23), so the 72B index-emission figure reported throughout this paper is now that rented-GPU, precision-matched calibration check, not the earlier API-based estimate; and the two frontier closed-model checks (GPT-4o, Claude Sonnet 5; App. B.26, B.27) via a hosted API (OpenRouter) rather than local or rented GPU compute. In every case: configuring each run by editing the designated configuration cell per the project's own convention, launching it, and downloading and reviewing the resulting artifacts (`RUN_COMPLETE.txt`, per-case CSVs, configuration JSON) before sharing them for analysis. The author reviewed and approved each stage of AI-assisted verification described above and each successive manuscript revision, including the precision deviation caught in the H9 run (§4.5) and the corrected reference attributions (§2), before any AI-reported number or claim was incorporated into the manuscript.

B PRE-REGISTRATION (H1–H12)

All hypotheses below were registered in `03-preregistration.md`. H1–H8 were part of the original document, dated 2026-08-22, before any experiment code was written or any model run — the document's own header records "zero runs executed, no data exists" at that time. H9–H12 were added as four later, individually dated amendments to this same living document, each logged before the data that would have motivated them. Status was determined after data collection and is reported regardless of outcome.

On multiple comparisons across the full hypothesis battery. With twelve pre-registered hypotheses (H1–H12), should one false-discovery-rate correction span all of them? We do not apply one: H1–H12 are not exchangeable tests of one null run repeatedly hoping one clears significance, but distinct, pre-registered claims about different aspects of the phenomenon, decided before any run. Correcting them against each other would penalize exactly the triangulated evidence the paper is built on — H9's three-scale replication is one pre-registered claim repeated, not three independent chances to get lucky. Where a genuine family of exchangeable comparisons exists within one hypothesis — H12's seven $|V|$ levels testing the same null — Holm-Bonferroni correction is applied there (App. B.7), which is the place it is actually warranted.

Table 4 summarizes every pre-registered hypothesis and its status. Two of these (H1, H7) follow near-definitionally from the experimental design itself — `random_valid`'s success rate is mathematically at least `random_unconstrained`'s (H1, since the valid set is a subset of the full pool) and equals $1/|V|$ by construction (H7, uniform choice over $|V|$ options) — so their "confirmation" is closer to a harness sanity check than a substantive empirical finding; included for completeness and transparency about what was pre-registered, not claimed as independent evidence for the paper's thesis.

Table 4: Pre-registered hypotheses and their status.

H	Claim (as registered)	Status
H1	`random_valid` > `random_unconstrained` (structure > 0)	**Confirmed on BFCL** (structure $= 0.295$ at native $\|V\|$, 7B run; positive at every scale, Table 5). **Underpowered on ALFWorld** (Appendix D: structure $= 0.008$, CI $[0.000, 0.024]$ at T=300 — consistent with > 0 but not distinguishable from a resolution floor).

continued on next page

Table 4 – continued from previous page

H	Claim (as registered)	Status
H2	`structure_share` ≥ 0.5, *on ALFWorld*, with two counter-example regimes pre-declared	**Underpowered as registered** (ALFWorld); **falsified as the analogous BFCL measurement** (§4.4: 0.353, CI $[0.266, 0.446]$). Not a literal test of the ALFWorld-scoped claim — see §4.4 for the distinction.
H3	`heuristic_valid` $\geq$ `constrained_llm` -5pp	**Confirmed for** $\lvert V\rvert \geq \lvert V\rvert^*$ (by construction of the crossover, §4.1: the heuristic does not merely close to within 5pp, it overtakes). **False at native** $\lvert V\rvert$ for every scale tested (e.g. 7B: heuristic 0.60 vs. LLM 0.91, a 31pp gap).
H4	Format confound: excluding parse failures from `unconstrained_llm` improves it materially (parsed-only $-$ raw > 0)	**Falsified / not applicable.** `parse_ok` on `unconstrained_llm` is 1.0 (7B run) — the model always produces a syntactically valid function name, so there is no parse-failure margin to recover. The near-zero floor for this arm (§4.4) is a behavioral limit (the wrong function, confidently named), not a formatting artifact.
H5	`structure_share` declines monotonically across Qwen2.5-Instruct 0.5B$\rightarrow$1.5B$\rightarrow$3B$\rightarrow$7B	**Confirmed.** Measured at native $\lvert V\rvert$: $0.711/0.391/0.351/0.326$ (Table 5) — monotone decreasing at every pairwise step.
H6	`structure_share` decays in $\lvert V\rvert$ at fixed T=1	**Confirmed.** 7B run, native$\rightarrow$234: $0.326 \rightarrow 0.133 \rightarrow 0.090 \rightarrow 0.041 \rightarrow 0.039 \rightarrow 0.015 \rightarrow -0.024$ — monotone decreasing except a sign flip at $\lvert V\rvert$=234 attributable to the documented resolution floor at that cell count (§4.4), not a substantive reversal.
H7	Parameter-free curve prediction: `random_valid` $= 1/\lvert V\rvert$, curve derived with no free parameters	**Confirmed.** 34/35 dilution points inside the 95% CI (§4.4, mean |residual|=0.0036 across the full diluted sweep); on the primary sweep's own undiluted levels, residuals ≤ 0.0013 against $1/\lvert V\rvert$ at n=1000 per level.
H8	KAMEL replication: `structure_share` at $\lvert V\rvert$=234 approaches the derived ≈ 0.01	**Confirmed within resolution limits.** Measured -0.024 (7B run) — near zero as predicted; the sign and magnitude are within the same resolution floor documented for H6, not a substantive deviation from the ≈ 0.01 prediction.

continued on next page

Table 4 – continued from previous page

H	Claim (as registered)	Status
H9	(Amendment 3) Filtering the candidate set to a model's own crossover before LLM selection closes most of the gap between unfiltered LLM selection and the heuristic, at $\|V\|$=234	**Confirmed at all three primary scales.** 1.5B: +41.5pp at 4-bit (CI $[0.320, 0.512]$, the pre-registered precision), +45.5pp at fp16 (CI $[0.357, 0.558]$). 3B (4-bit): +49.5pp (CI $[0.400, 0.585]$), significant exactly at $\|V\|$=32, past 3B's own crossover (28.7). 7B (4-bit): +31.0pp (CI $[0.180, 0.435]$), significant exactly at $\|V\|$=128, past 7B's own crossover (109.2). All CIs exclude zero; the null below each model's own k is partly structural (filtering to k is a no-op below it), and §4.5 finds most of the gain above it is a ranking effect.
H10	(Amendment 4) A semantic (embedding) zero-learning retriever performs comparably to the lexical-overlap heuristic, ruling out BFCL lexical transparency as the sole driver of the crossover	**Semantic reliably above lexical** (pre-committed outcome 3, not outcome 1 as loosely anticipated). `semantic_valid` beats `heuristic_valid` by 30–37pp at every $\|V\|$ level (native through 234), all 95% CIs excluding zero (App. B.2). No LLM arms in this leg; does not itself move the reported $\|V\|^*$ or α values, which remain defined against the lexical heuristic.
H11	(Amendment 5) Replacing tool names with a 12-character hash (descriptions untouched) does not collapse the crossover to chance — the realistic production-opacity case	**Falsified as stated; the LLM loses disproportionately.** Measured $\|V\|^* = 8.6$, 95% CI $[4.0, 17.9]$ — non-overlapping with the transparent 3B baseline (28.7, CI $[19.5, 43.4]$), roughly a 3× collapse. The heuristic's advantage narrows sharply under name-opacity, though a finite crossover persists (App. B.7).
H12	(Amendment 5) Replacing every description with an identical placeholder (full lexical opacity, a code-level identity) collapses `heuristic_valid` to `random_valid`'s own tie-break rule	**Confirmed at 6 of 7 $\|V\|$ levels** (CI including zero, as the construction predicts). One level ($\|V\|$=64) shows a real, Holm-corrected deviation ($p = 0.004$), reported as an anomaly rather than explained away. The LLM also collapses toward the random floor under full opacity (all seven levels, CI including zero): with zero exploitable signal, both arms converge to the same non-result (App. B.7).

Two amendments were logged before any run, both changing scope rather than re-cutting a result: H2 was scoped to ALFWorld with two counter-example regimes pre-declared (Amendment 1), and H5 was restricted to the Qwen2.5-Instruct family after a full-text read of Cho (2026) found that instruction-tuning quality, not raw parameter count, moderates harness sensitivity in

Table 5: H5: structure_share at native $|V|$, by Qwen2.5 scale (4-bit).

	0.5B	1.5B	3B	7B
`random_valid`	0.300	0.300	0.300	0.300
`random_unconstrained`	0.005	0.005	0.005	0.005
`constrained_llm`	0.420	0.760	0.845	0.910
structure	0.295	0.295	0.295	0.295
learning	0.120	0.460	0.545	0.610
structure_share	**0.711**	**0.391**	**0.351**	**0.326**

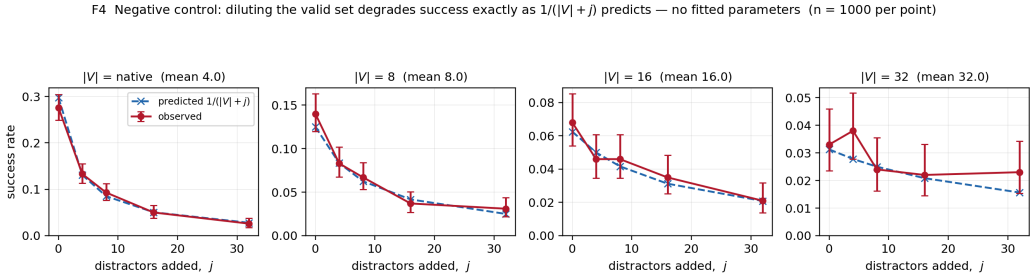


Figure 3: Negative control: diluting the valid set with j distractors degrades success exactly as the parameter-free prediction $1/(|V|+j)$ forecasts, at every native $|V|$ tested, with no fitted parameters.

their design — mixing families on the H5 scale axis would have confounded scale with recipe. `Llama-3.1-8B-Instruct` was moved off the H5 axis as a result and retained only as a separate architecture-robustness note.

B.1 Dilution negative control: figure

§4.4 reports the headline result in prose (34/35 points inside the 95% CI, no fitted parameters); this is the corresponding figure.

B.2 H10: semantic vs. lexical zero-learning retriever

`E11_bfcl_semantic_retriever.ipynb` adds a fourth, LLM-free arm, `semantic_valid`: the same zero-parameter, zero-training top-1 selection logic as `heuristic_valid`, scored by cosine similarity between off-the-shelf sentence embeddings (`all-MiniLM-L6-v2`, not fine-tuned on BFCL or any tool-calling data) instead of lexical token overlap. Same convention as every other BFCL leg: 40 cases $\times$ 5 seeds, all seven $|V|$ levels, `MASTER_SEED=20260822`.

A bug was found and fixed during independent re-verification. The notebook's own summary artifact (`h10_semantic_vs_lexical.csv`) initially reported valid statistics only at $|V|$=`native`, with n=0 at every other level, due to a type-comparison bug in the analysis cell (a DataFrame column compared against a string-cast loop variable without casting the column itself) compounded by a merge that joined only on `task_id`, not `task_id`+`seed_idx`. This bug was local to this leg's own one-off summary script; the primary BFCL sweep's shared analysis core pivots on the composite (`task_id`, `seed_idx`) key by construction (verified directly in the shared bootstrap code used for Table 2), so this specific collision class does not apply there. The raw per-case data (`per_case.csv`, 5,600 rows) was unaffected; only the summary cell's own aggregation was wrong. Re-deriving the comparison directly from `per_case.csv` gives Table 6:

All seven CIs exclude zero; the gap is largest at $|V|$=128 and does not shrink as $|V|$ grows, i.e. the semantic retriever's advantage over the lexical heuristic is not an artifact confined to small candidate sets. This is pre-committed outcome 3 of Amendment 4 (03-preregistration.md): the lexical heuristic, if anything, understates what a cheap off-the-shelf zero-learning retriever can do on this benchmark. This leg has no LLM arms and does not itself relocate the paper's reported $|V|^*$ or α values, which are defined specifically against the lexical-overlap heuristic throughout; it addresses a narrower, pre-registered question (is the crossover contingent on the weaker of two available zero-learning methods), which it answers no. It does not test robustness to non-lexically-transparent catalogs (e.g. opaque service IDs), which remains open (§7). A stronger retriever widening this gap

Table 6: H10: `semantic_valid` − `heuristic_valid`, paired bootstrap over cases (n=200 per level, n_{boot}=10,000).

$\|V\|$	native	8	16	32	64	128
diff	+0.305	+0.340	+0.340	+0.340	+0.350	+0.365
95% CI	[0.192,0.415]	[0.225,0.452]	[0.232,0.446]	[0.236,0.442]	[0.242,0.456]	[0.258,0.469]

$\|V\| = 234$
diff: +0.345, 95% CI $[0.244, 0.440]$, $p < 0.001$

does not weaken the paper's main claim: that claim was never "this is the best possible router," only "even a free, zero-parameter one beats the model past $|V|^*$" — a stronger retriever clearing that bar strengthens H9's prescription (§4.5), not the heuristic used throughout the main sweep.

Post-hoc extension across all five scales. The above tests H10 at 3B only. To check whether α is a property of scale or of the specific (weak) baseline chosen, we extended `semantic_valid` to all five Qwen2.5 scales, in a fresh run against `constrained_llm` (not merely re-scored against existing `heuristic_valid` data). This was not pre-registered — it was added after the original hypothesis log closed — and is reported as such rather than folded into H1–H12.

Table 7: `semantic_valid` crossover by scale, paired bootstrap (n_{boot}=2000), vs. this run's own `heuristic_valid` replicate.

Scale	`heuristic_valid` $\|V\|^*$	`semantic_valid` $\|V\|^*$	95% CI (semantic)
1.5B	10.9	$\leq 3.95^\dagger$	left-censored
3B	30.1	$\leq 3.95^\dagger$	left-censored
7B	119.1	25.4	$[4.0, 50.1]$
14B	439.2	44.0	$[4.0, 128.0]$

†Same convention as Table 2's 0.5B row: `semantic_valid` already leads `constrained_llm` at native $|V|$, so no interior grid point brackets the true crossover; not a genuinely-interpolated value.

This fresh run's own `heuristic_valid` replicate lands close to but not exactly at the originally-published Table 2 values (e.g. 7B: 119.1 here vs. 109.2 published, 3B/1.5B/7B all higher, 14B lower — not one-directional) — most plausibly ordinary run-to-run variation in `constrained_llm`'s outputs between sessions, not a change to `heuristic_valid` itself, which is deterministic and model-independent by construction (§3); reported here for transparency, not further investigated. Fitting $|V|^* \propto N^\alpha$ on the four interior points using `semantic_valid` as the reference arm gives $\alpha \approx 1.22$, against this same run's own $\alpha \approx 1.65$ for `heuristic_valid` (both within the published CI); with half the interior points left-censored rather than genuinely interpolated, this fit rests on two real data points and two floors, not four independent measurements, which the reported CI and LOO range do not fully capture. The semantic-baseline CIs are substantially wider than the lexical-baseline ones, and leave-one-out refitting gives $\alpha \in [1.10, 1.59]$ — markedly less stable than the primary sweep's $[1.640, 1.845]$. We do not report this as a replacement α: it is directionally clear that the headline growth rate depends on baseline choice, but this specific check is weaker — both in resolution and in effective degrees of freedom — than the primary sweep.

B.3 SEAL-TOOLS GENERALIZATION CHECK: FULL DETAIL

Summarized in the main text (§4); this section gives the full design and numbers.

`test_out_domain` specifically (unseen during construction), $|V| \in \{8, 16, 32, 64, 128, 350\}$ (no native $|V|$ in this format), n=94 × 5 seeds, at five scales matching BFCL's own primary range. n=94 is the full single-call pool (654 total rows, 560 multi-call excluded — same single-step scoping reason argument binding is out of scope elsewhere in this paper), not a subsample of a larger pool.

Crossover points (case-level bootstrap, $n_{\text{boot}} = 4{,}000$) and comparison to BFCL's own crossover at the same scale:

All scales are now direction-consistent or CI-overlapping with BFCL's own crossover — up from one CI-overlap (3B) and one left-censored direction-consistent point (1.5B) in the original 30-case check, with 7B and 14B previously giving no comparison point at all: the $|V|$=128 extension re-

Table 8: Seal-Tools `constrained_llm` vs. `heuristic_valid` success rate, by $|V|$, with case-level bootstrap 95% CIs on the difference ($n_{\text{boot}} = 4{,}000$). * CI excludes zero.

			0.5B	
$\|V\|$	LLM	heuristic	diff	95% CI
8	0.345	0.834	-0.489*	[-0.560, -0.415]
16	0.153	0.781	-0.628*	[-0.702, -0.547]
32	0.055	0.734	-0.679*	[-0.760, -0.591]
64	0.032	0.687	-0.655*	[-0.743, -0.564]
128	0.004	0.647	-0.643*	[-0.728, -0.553]
350	0.000	0.574	-0.574*	[-0.666, -0.483]
			1.5B	
$\|V\|$	LLM	heuristic	diff	95% CI
8	0.704	0.834	-0.130*	[-0.206, -0.047]
16	0.570	0.781	-0.211*	[-0.298, -0.123]
32	0.411	0.734	-0.323*	[-0.413, -0.228]
64	0.217	0.687	-0.470*	[-0.568, -0.372]
128	0.115	0.647	-0.532*	[-0.617, -0.443]
350	0.013	0.574	-0.562*	[-0.653, -0.468]
			3B	
$\|V\|$	LLM	heuristic	diff	95% CI
8	0.947	0.834	+0.113*	[+0.049, +0.177]
16	0.887	0.781	+0.106*	[+0.038, +0.177]
32	0.585	0.734	-0.149*	[-0.247, -0.053]
64	0.423	0.687	-0.264*	[-0.357, -0.168]
128	0.274	0.647	-0.372*	[-0.464, -0.279]
350	0.119	0.574	-0.455*	[-0.549, -0.364]
			7B	
$\|V\|$	LLM	heuristic	diff	95% CI
8	0.985	0.834	+0.151*	[+0.089, +0.221]
16	0.977	0.781	+0.196*	[+0.123, +0.270]
32	0.962	0.734	+0.228*	[+0.149, +0.313]
64	0.779	0.687	+0.091*	[+0.002, +0.181]
128	0.455	0.647	-0.191*	[-0.285, -0.094]
350	0.170	0.574	-0.404*	[-0.498, -0.313]
			14B	
$\|V\|$	LLM	heuristic	diff	95% CI
8	0.991	0.834	+0.157*	[+0.096, +0.226]
16	0.989	0.781	+0.209*	[+0.136, +0.283]
32	0.983	0.734	+0.249*	[+0.172, +0.330]
64	0.804	0.687	+0.117*	[+0.028, +0.209]
128	0.677	0.647	+0.030	[-0.064, +0.130]
350	0.555	0.574	-0.019	[-0.113, +0.079]

Table 9: Seal-Tools $|V|^*$ vs. BFCL $|V|^*$, all five scales.

Scale	Seal-Tools $\|V\|^*$	BFCL $\|V\|^*$	Relationship
0.5B	≤ 8.0 (left-censored)	left-censored	direction-consistent
1.5B	≤ 8.0 (left-censored)	8.7, $[4.1, 20.2]$	direction-consistent
3B	21.4, $[17.8, 27.1]$	28.7, $[19.5, 43.4]$	CI overlaps
7B	80.1, $[64.9, 100.3]$	109.2, $[73.6, 185.3]$	CI overlaps
14B	236.1, $[78.8, 333.6]$	500.0, $[234.0, >650]$	CI overlaps

solved 7B; a further extension to $|V|$=350 (matching BFCL's own extension pattern at 14B) resolved 14B similarly.

14B's own point. The interpolated crossover (236.1) sits between $|V|$=128 and 350, at neither of which the LLM-vs-heuristic difference is individually significant (diff $+3.0$pp, CI $[-6.4, +12.6]$ at 128; -1.9pp, CI $[-11.9, +7.4]$ at 350; both include zero). 236.1 is a genuine interpolation

between two points that individually do not distinguish LLM from heuristic, not a value supported by a significant difference at either bracketing point on its own — worth reading with this in mind, though it does not affect the log–log collinearity finding below, which concerns the three scales' points relative to each other, not this one point's own precision.

An ordinary crossover at 3B, not visible in the original coarser grid, corrected from an earlier "non-monotonic" description that overstated the pattern: the diff sequence across $|V| \in \{8, 16, 32, 64, 128, 350\}$ is $+11.3$pp, $+10.6$pp, -14.9pp, -26.4pp, -37.2pp, -45.5pp — monotonically non-increasing throughout, never reversing direction. What the finer grid reveals is that the LLM *beats* the heuristic at $|V|$=8 ($+11.3$pp) and $|V|$=16 ($+10.6$pp), both CIs excluding zero, before crossing over and falling behind from $|V|$=32 on — a genuine, reportable finding in its own right, and one the original 30-case, coarser grid missed entirely, but a single crossover, not an oscillation. Reported plainly, not smoothed into the crossover discussion above.

A methodological note on 0.5B and 1.5B, kept visible rather than smoothed over: the crossing-point function used elsewhere in this paper returns the grid's own floor whenever the heuristic already leads at the first tested point, which happened in all $4{,}000/4{,}000$ bootstrap resamples for both scales — reported correctly as left-censored (≤ 8.0, no CI), not as a spuriously precise "8.0."

B.4 A SEAL-TOOLS ALPHA FIT

With five scales now tested, Seal-Tools supports its own $|V|^* \propto N^\alpha$ fit, directly comparable to BFCL's own (§4.2). Fitting on the interior (non-left-censored) points only — 3B, 7B, and 14B; 0.5B and 1.5B are left-censored and excluded, the same convention BFCL's own fit uses to exclude its 0.5B — gives $\alpha_{\text{Seal-Tools}} = 1.560$, 95% CI $[0.831, 1.810]$ (case-level bootstrap, resampling cases within each scale and refitting both the crossover and the regression on every resample, $n_{\text{boot}} = 4{,}000$). BFCL's own: $\alpha = 1.784$, CI $[1.284, > 2.027]$, 4 interior points (§4.2). The two CIs overlap substantially.

This is not a methodology choice — Seal-Tools' own 1.5B is left-censored for the same reason BFCL's 0.5B is, leaving one fewer interior point. Any comparison of these two α values should be read with that difference in mind.

Leave-one-out sensitivity, checked specifically because a 3-point fit was flagged as a fragility risk before this experiment ran: the risk did not materialize. The three interior points are remarkably close to exactly collinear in log–log space — pairwise slopes computed independently from each pair are 1.5597, 1.5598, and 1.5600, agreeing to the fourth decimal — and dropping any single point leaves α unchanged at 1.560 to three decimals. This degree of collinearity is surprising enough to warrant checking the computation itself rather than trusting it on inspection: an independent re-derivation from the raw crossover points in a fresh script reproduces the same three slopes exactly, confirming this is a genuine property of the data rather than a shared-variable or copy-paste artifact in the original computation.

A deeper check: the crossovers themselves, not just the slope arithmetic on them, re-derived from raw per-case data. The check above confirms the pairwise-slope computation is correct given the reported $21.4/80.1/236.1$; it does not by itself confirm those three crossover values are themselves correct. Re-running the same crossing-point method used throughout this paper directly on the merged per-case success/failure records (not the already-computed crossover table) gives 21.36, 80.08, and 236.12 — matching to within rounding at every scale. Repeating the collinearity check on these independently-derived, full-precision values (rather than the paper's rounded table entries) gives slopes of 1.5598, 1.5599, and 1.5600 — the same collinearity to four decimal places, now confirmed end to end: raw success rates $\rightarrow$ crossovers $\rightarrow$ pairwise slopes, with no step trusted on the strength of an earlier step's output alone.

B.5 NAME-EMISSION REPLICATION

Does the paper's name-emission recovery (§4.6) hold on Seal-Tools too? Tested at 3B and 7B, $|V| \in \{8, 16, 32, 64, 128\}$ (matching BFCL's own tested scope for this finding), reusing the existing `constrained_llm` (index-emission) and `heuristic_valid` data from App. B.3 rather than re-running it.

Table 10: Seal-Tools name-emission vs. index-emission, by $|V|$, with case-level bootstrap 95% CIs on the difference ($n_{\text{boot}} = 4{,}000$). * CI excludes zero.

			3B	
$\|V\|$	name	index	diff	95% CI
8	0.981	0.947	+0.034*	[+0.009, +0.057]
16	0.981	0.887	+0.094*	[+0.064, +0.126]
32	0.981	0.585	+0.396*	[+0.349, +0.445]
64	0.977	0.423	+0.553*	[+0.506, +0.600]
128	0.957	0.274	+0.683*	[+0.636, +0.728]
			7B	
$\|V\|$	name	index	diff	95% CI
8	0.985	0.985	+0.000	[-0.006, +0.006]
16	0.994	0.977	+0.017*	[+0.006, +0.032]
32	0.989	0.962	+0.028*	[+0.011, +0.047]
64	0.977	0.779	+0.198*	[+0.155, +0.243]
128	0.962	0.455	+0.506*	[+0.457, +0.553]

The gap grows with $|V|$ at both scales and is significant everywhere tested except 7B at $|V|$=8, where both arms are already near-saturated (0.985 each) and the CI straddles zero — the same qualitative pattern as BFCL's own name-emission result.

At $|V|$=128, 3B's name-emission success (0.957) exceeds `heuristic_valid`'s own success at that level (0.647): name emission alone, with no filtering, beats the zero-parameter heuristic here.

B.6 H9 (RETRIEVAL-THEN-SELECT) REPLICATION

Does filtering candidates to a model's own crossover before selection (H9, §4.5) close most of the gap on Seal-Tools too? Tested at 3B, $|V|$=128, k=22 ($\lceil 21.4 \rceil$, matching the paper's own filter-depth convention), filtering by the same zero-parameter lexical scorer as `heuristic_valid` before running `constrained_llm` selection among survivors only.

Table 11: Seal-Tools H9 replication, 3B, $|V|$=128, k=22.

Filter recall (true candidate survived top-22)	0.868
Filtered success (`retrieval_filtered_llm`)	0.755
Unfiltered success (`constrained_llm`, App. B.3)	0.274
Diff, 95% CI (case-level bootstrap, $n_{\text{boot}} = 4{,}000$)	$+0.481$, $[+0.404, +0.560]$

Filtering closes most of the gap — the same qualitative pattern as BFCL's own H9 result, on a second benchmark. 7B was not run for this leg (scope-limited to 3B, matching a minimum-viable replication scope); left for future work.

B.7 DOES THE CROSSOVER NEED INFORMATIVE TOOL NAMES? FULL DETAIL

Summarized in the main text (§4); this section gives the full design and numbers.

Every result elsewhere in this paper uses BFCL's catalog, lexically transparent by construction (§1). Two pre-registered tests at 3B (full status and numbers in Table 4): **H11** replaces tool names with a 12-character hash, descriptions untouched — the realistic production case — and collapses the crossover roughly 3× ($|V|^* = 8.6$, CI $[4.0, 17.9]$, vs. the transparent baseline 28.7, CI $[19.5, 43.4]$): the LLM loses far more from name-opacity than the heuristic does. **H12** additionally replaces every description with an *identical* placeholder, a code-level identity forcing `heuristic_valid`'s tie-break to reduce exactly to `random_valid`'s own rule; it is confirmed at 6 of 7 $|V|$ levels, with one Holm-corrected anomaly at $|V|$=64 (p=0.004) — the LLM also collapses toward the random floor under full opacity, both arms converging to the same non-result.

A third, non-hash control: opaque but easy to copy. H11's hashed names conflate two things: losing the lexical shortcut between question and candidate name, and the raw difficulty of copying an unfamiliar 12-character token. At 3B, 83% of H11's failures at $|V|$=234 are bare numeric strings — the model abandoning the hash format entirely, not making near-miss copy errors on it — suggesting

copy-difficulty, not lexical-shortcut loss, may be doing most of the work. This control disentangles the two: candidate names replaced with short (1–2 syllable), common, correctly-spelled English words, semantically unrelated to the function they replace and to each other (e.g. "bow", "dash", "grove"), one per candidate, matching H11's exact scope, scale, and seeding convention — only the name-generation scheme changes, using the same `constrained_llm`/`heuristic_valid` arms unmodified. A vocabulary-overlap check (automated, before running) excluded any candidate word overlapping the existing function-name/description corpus: of 5,206 words surviving length and frequency filtering, 1,013 were excluded for overlap, leaving ample margin over the 1,222 distinct functions renamed.

Result: closer to transparent than to hashed, but not all the way. Three-way comparison at 3B, `constrained_llm`:

$\|V\|$	Transparent	H11 hashed	Opaque, easy-to-copy
native	0.845	0.730	0.885
8	0.810	0.515	0.790
16	0.760	0.365	0.685
32	0.470	0.200	0.305
64	0.215	0.085	0.125
128	0.175	0.075	0.095
234	0.065	0.025	0.025

Crossovers: transparent $|V|^* = 28.7$, CI $[23.4, 36.9]$; H11 hashed $|V|^* = 8.6$, CI $[5.6, 12.6]$ (matches the main-text figure exactly); opaque-easy-copy $|V|^* = 22.8$, CI $[19.8, 26.5]$. A paired-difference bootstrap (same resampled cases) finds opaque-easy-copy significantly above H11 hashed at native through $|V|$=32 (up to +32.0pp at $|V|$=16) and significantly below transparent at $|V| \geq 32$ (up to −16.5pp at $|V|$=32) — both CIs exclude zero at these levels. Reading this plainly: most, but not all, of H11's collapse reflects copy-difficulty rather than lost lexical shortcut. A smaller, genuine lexical-shortcut effect survives even with easy-to-copy opaque names, so both factors are real; copy-difficulty is the larger piece of H11's original result.

Together: name-opacity alone hurts the LLM disproportionately; full opacity hurts both arms equally; and most, though not all, of that disproportionate hurt is attributable to copy-difficulty rather than a purely lexical effect. Every reported $|V|^*$ elsewhere in this paper holds for lexically-transparent catalogs specifically (§7).

B.8 Higher-power re-measurement at 3B

The primary sweep uses 40 cases per $|V|$ level. This section directly addresses statistical power: more cases, a finer grid near the crossover, and a parametric curve with a case-level bootstrap rather than grid-point interpolation.

Design. 3B, 100 cases (2.5× the primary sweep) × 3 seeds, a 12-point grid bracketing 3B's own reported BFCL crossover ($|V|^* = 28.7$, CI $[19.5, 43.4]$): {native, 8, 12, 16, 18, 20, 22, 24, 26, 28, 32, 40}. Two arms only, `constrained_llm` and `heuristic_valid`, matching the primary sweep's own comparison exactly.

Result. Simple interpolation on this finer grid gives $|V|^*$=30.6, 95% CI $[23.0, 38.7]$. A logistic curve ($P(\text{success}) = \sigma(a + b \log |V|)$) fit separately to each arm, with the crossover taken as the intersection of the two fitted curves and a case-level bootstrap (n_{boot}=2000, refitting both curves on each resample) gives $|V|^*$=32.4, 95% CI $[24.8, 43.9]$ — note this upper bound sits past the grid's own maximum ($|V|$=40), so it reflects the fitted curve's extrapolation beyond the tested range, not a directly observed point; all 2000 resamples converged. Both methods agree with each other and overlap the original 40-case grid estimate (28.7, CI $[19.5, 43.4]$) — a refinement, not a contradiction, though this check is 3B only, index-emission only, and does not speak to α or any other scale.

B.9 H9 k-sweep and fixed-k baseline

H9 (§4.5) sets k to each model's own measured $|V|^*_{\text{4-bit}}$ without testing whether that specific choice is necessary. This was not pre-registered: it directly tests whether the gain is specific to k=$|V|^*$

or would follow from any reasonable fixed depth, and whether a fixed k (as used in prior tool-use literature) performs comparably.

Design. Qwen2.5-3B-Instruct, $|V|$=234 (where H9's headline numbers are reported), same harness, prompts, and `MASTER_SEED` as every other BFCL leg: 40 cases $\times$ 5 seeds, filtering with the same zero-parameter lexical scorer as `heuristic_valid`, extended to keep the top-k rather than only the single best. $k \in \{5, 15, 29, 50\}$, where 29 is H9's original choice and 5 is a fixed baseline shared across scales, matching the fixed retrieval depth used in prior tool-retrieval work (§2).

Filtering's value is even more strongly confirmed. Every k tested beats unfiltered `constrained_llm` (0.095) by a large, clearly significant margin: k=5: $+0.445$, CI $[0.310, 0.580]$; k=15: $+0.515$, CI $[0.390, 0.640]$; k=29: $+0.485$, CI $[0.370, 0.600]$; k=50: $+0.405$, CI $[0.290, 0.520]$ (n_{boot}=10,000, paired bootstrap over cases). Recall rises monotonically with k: 0.580 at k=5, 0.790 at k=29, 0.830 at k=50, as expected — the k=29 value is what App. B.11 and §4.5 independently agree with, stated here explicitly rather than left implicit between the two endpoints.

A note on the re-measured baseline. This run's own 0.095 unfiltered baseline differs from the 0.065 reported for the identical (3B, $|V|$=234) configuration in §4.5 and the primary sweep (Table 2) — traced to source rather than left unexplained. The two runs' candidate lists were directly reconstructed from `MASTER_SEED` and compared task-by-task: identical in every one of the 14 (of 200) cells where the two runs' outputs differ — each run's emitted index maps correctly to its own reported action within the same, single reconstructed list, ruling out a seeding or candidate-construction bug. With an identical prompt and greedy (temperature-0) decoding, this points to genuine cross-session output variation (differing hardware or library versions between runs) rather than a methodology error; it does not affect this section's own within-run comparisons, all of which use 0.095 consistently as their own baseline. **Which run §4.5's own headline number uses, stated explicitly:** the $+0.495$ reported there is from the original H9 test (the run this section's own 0.065 figure is drawn from), not this k-sweep's $+0.485$ on its own 0.095 baseline — the two are close but drawn from different sessions, for the reason just traced. This session-to-session variation is a real noise source the case-level bootstrap CIs do not capture (they resample cases within one run, not across runs) — at the scale already established elsewhere in this paper (the dilution control's independent rerun of `constrained_llm`, §4.4: 2–4pp at n=40), not a new or larger concern specific to H9.

Direct confirmation: library-version-pinned reruns are bit-for-bit reproducible. The account above ("differing hardware or library versions between runs") was inferred, not directly tested, at the time it was written. It was tested directly with three further independent repeats of this exact configuration (3B, $|V|$=234, unfiltered `constrained_llm`), each launched as a separate session on a separate rented instance. All three reproduced *exactly* the same 200 cells — identical chosen action, identical raw output, identical success flag in every one of 200/200 cells across all three repeats (0.090 each, and diffed cell-by-cell against each other, not merely compared on their aggregate rate). What the three repeats had in common, and what neither of the two original 0.065/0.095 runs is known to share, is an identical pinned library stack (`torch==2.11.0+cu128`, `transformers==5.17.0`, `bitsandbytes==0.50.2`, `accelerate==1.15.0`) — despite running on different physical GPUs and different NVIDIA driver versions (570.211.01 vs. 590.48.01) and kernel builds. This narrows the account rather than merely restating it, but only as far as the evidence actually reaches: at temperature-0 with a fixed library stack, decoding is fully deterministic across sessions and hardware, so hardware variation and driver version are ruled out as candidate explanations for the original 0.065/0.095 split, since the three repeats varied both of those while holding the library stack fixed and still reproduced identically. What this does *not* do is isolate differing library versions as the split's actual cause: no run was performed comparing two different pinned library stacks against each other, only three repeats of the same stack against itself. The original two runs' own library versions were not recorded at the time, so a direct library-version comparison cannot be reconstructed retroactively either. The corrected, supportable claim is narrower than previously stated: a fixed library stack is deterministic across sessions and hardware, and library-version differences remain the leading candidate explanation for the original split by elimination of the alternatives actually tested, consistent with, but not a confirmed isolation of, a library-version account.

What this means for the 14B→32B comparison specifically. That comparison (App. B.22) draws on runs launched across different session types — 14B on a Kaggle notebook, 32B on a rented GPU instance — which is exactly the kind of cross-session boundary this section's finding concerns. Since only same-stack determinism was directly confirmed here, not a library-version account of cross-session differences, the local exponent between these two points should be read with the same session-to-session uncertainty already disclosed above (a rough few-percentage-point scale, not a formally propagated interval) as an additional, unquantified source of noise beyond each point's own reported CI, rather than as a reason to distrust those points specifically. The 32B→72B exponent is now on the same footing rather than a worse one: 72B's figure is itself a rented-GPU, precision-matched run (App. B.23), so this exponent no longer mixes methodologies, though it still crosses session boundaries — 32B and 72B ran on separate rented instances, and 72B's own native and extension ranges were themselves two separate sessions — so the same session-to-session caveat above applies here too, not a larger, distinct one.

A rough sense of scale, stated explicitly rather than left implicit. The two *distinct* realizations available at this exact configuration (3B, $|V|$=234, unfiltered) — 0.065 from the original H9 test and 0.095 from the k-sweep baseline — differ by 3pp, consistent with the dilution-control estimate above; the three further repeats above reproduced a third value (0.090) exactly and do not add a fourth and fifth independent draw to this spread, for the reason just given (they are not independent of each other). Against §4.5's own reported CI width for this test ($[0.400, 0.585]$, an 18.5pp span), 3pp is roughly 16% of the interval's own width: a real, non-negligible addition to the true uncertainty, though not one this or any other CI in this paper currently reflects. With only two *distinct* library-version realizations observed at any single configuration, this remains too little to fit a formal between-session variance component with any confidence, so it is reported as this rough scale estimate rather than used to mechanically widen the reported CI — an honest bound on the unaccounted-for uncertainty, not a corrected interval. A direct test of the library-version account would compare repeats deliberately run on two or more *different* pinned library stacks rather than the same one three times; that comparison is not available this round.

An attempt to fold this into a CI's own width, and why it was not adopted. A natural approach: inflate the case-level bootstrap by also randomly flipping a fraction of cells per resample, matching the observed cross-session flip rate ($14/200$=7% here). Tried directly on this section's own `ksweep_k29` arm: the resulting CI is not meaningfully wider than the unperturbed one (width $0.240 \to 0.225$, if anything narrower). This is not because session noise is negligible — it is because independent, per-cell random flipping is the wrong model for it: with 200 largely-independent cells, per-cell noise averages out under the law of large numbers almost entirely, while real cross-session variation is a correlated shift affecting an entire run together (differing hardware or library versions, not independent per-response coin flips), which this simulation does not represent. Reported for transparency rather than adopted: propagating this noise correctly would need session-level replicates (several independent full reruns) to estimate a between-session variance component directly, not a within-run simulation. That is not available this round, so this remains a disclosed, unresolved limitation of every CI in this paper, not one folded into their reported widths.

k=$|V|^*$ is not shown to be uniquely optimal. Point estimates peak at k=15 (0.610), not at H9's original k=29 (0.580); the paired difference is $+0.030$, CI $[-0.010, 0.080]$, not excluding zero. More directly: k=29 (adaptive) vs. the fixed k=5 baseline gives $+0.040$, CI $[-0.025, 0.105]$ — also not excluding zero. We do not claim the model-specific choice is proven better than a simple fixed depth at this sample size; we report the sweep's actual result rather than the single point that was easiest to defend. The robust claim is narrower and, we think, still substantial: filtering helps enormously across a wide range of k, not only at one cherry-picked value.

Confirmed independently at 7B. Same design, $k \in \{5, 50, 110, 180\}$ (110 is 7B's own H9 choice). Every k again beats unfiltered selection significantly. As at 3B, the adaptive choice is not shown to be uniquely optimal: k=110 vs. fixed k=5 gives $+0.005$, CI $[-0.105, 0.120]$, not excluding zero — the same pattern, independently, at a second scale.

B.10 H9 RANKED VS. SHUFFLED SURVIVOR ORDER

A related, equally specific question the fixed-k check above does not answer: does H9's gain depend on *how* survivors are shown to the model, not just which ones survive? Survivors are currently

shown in ranked order (highest lexical score first) — if the model partly exploits position rather than content, that would inflate H9's apparent gain without reflecting genuine selection improvement.

Design. 3B and 7B, $|V|$=234, each model's own H9 k (29 and 110 respectively), 40 cases $\times$ 5 seeds. Identical candidate sets to the ranked condition; survivors are shuffled into random order before being shown to the model, holding everything else fixed.

A real, significant effect, at both scales. Ranked order significantly beats shuffled order: 3B, $+0.215$, 95% CI $[0.120, 0.305]$; 7B, $+0.195$, 95% CI $[0.075, 0.305]$ — both clearly excluding zero, similar magnitude at both scales. H9's gain is genuinely, consistently partly explained by rank order, not purely by content-based filtering. This does not invalidate H9 — filtering to a smaller set still helps substantially regardless of order (App. B.9) — but the specific size of the reported gain should be read with this caveat, not without it.

A small residual gap against App. B.11's own ranked/shuffled numbers. The corrected App. B.11 reports $+0.205$ (3B) and $+0.230$ (7B) for the identical (model, k, $|V|$=234) configuration — close but not identical to this section's $+0.215$/$+0.195$. Both this section's notebook (E32) and App. B.11's corrected one already use the same, correct tokenizer (confirmed by direct inspection, not assumed), so this is not a repeat of the tokenizer bug fixed there.

The two arms behind this gap turn out to have different causes, checked separately rather than assumed to share one. **Ranked arm:** task-by-task comparison finds exactly 1 of 200 cells differing at 3B and 2 of 200 at 7B (identical `task_id`/`seed_idx`, different emitted index in each) — the same small-scale cross-session greedy-decoding variation already disclosed in App. B.9 (14 of 200 cells there), not a construction or seeding error. **Shuffled arm: a different, larger, and fully identified cause.** 74 of 200 cells differ at 3B and 75 of 200 at 7B — far more than cross-session noise would produce, and confirmed to be something else entirely: this section's own shuffle-policy seed uses `f"SHUF_k{k}"` as its third seed-derivation component, while App. B.11's corrected notebook uses the arm name itself (`"index_shuffled_k"`) for the same role. The two strings differ, so the two notebooks' `random.Random` instances start from different seeds and produce two different, unrelated (though each individually valid) random orderings of the same top-k survivors — verified directly by re-deriving both seeds and shuffling a test sequence with each, which gives visibly different permutations. Both shuffled arms are still doing exactly what they are meant to (removing rank signal from a valid random ordering), so this does not affect the validity of either paper's own internal ranked-vs-shuffled comparison — but it means the two notebooks' shuffled-arm success rates are two different, independent random realizations of the same design, not two measurements of the same thing that should agree, and their gap (0.365 vs. 0.370 at 3B; 0.350 vs. 0.325 at 7B) reflects that, not an error in either notebook.

B.11 A 2×2: INDEX/NAME EMISSION × RANKED/SHUFFLED ORDER

Two effects in this paper each cost real performance: emitting an index instead of a name (§4.6), and showing survivors in shuffled rather than ranked order (App. B.10). Are these the same underlying mechanism, or two separable ones? A direct 2×2 tests this.

A flawed design, disclosed for transparency. Reusing `constrained_llm` and `name_constrained_llm` directly as the "ranked" cells does not work: those arms' candidate order is randomized by construction (§3), not lexically ranked — so that design would compare two random orderings against each other, making any null result from it uninformative. We do not use that design; the corrected one below replaces it entirely.

Corrected design. Filter to each model's own k (29 at 3B, 110 at 7B, matching App. B.9), using the same genuine lexical ranking as App. B.10 (via the scorer `heuristic_valid` uses, applied after filtering). Compare these survivors kept in ranked order against the identical filtered set shuffled, crossed with index vs. name output. $|V|$=234, 40 cases $\times$ 5 seeds, both scales.

A tokenizer inconsistency, identified and corrected. `recall_at_k` disagrees across this section, App. B.9, and §4.5 for the nominally identical (model, k, $|V|$) configuration. Traced to source: this section's own lexical-scorer helper is implemented with `str.split()` (whitespace only) rather than the regex-based tokenizer every other lexical scorer in this paper uses (App. B.9, `heuristic_valid`). Punctuation stays glued to words under `str.split()` ("weather," $\neq$ "weather"), which systematically under-counts lexical overlap and pushes the true candidate out

of the top-k more often than it should. Numbers below use the corrected, regex-based tokenizer (recall of 0.665 at 3B and 0.835 at 7B under the uncorrected `str.split()` version were both too low by this mechanism).

Result: a real, significant interaction, confirmed at both scales.

	3B (k=29)		7B (k=110)	
	ranked	shuffled	ranked	shuffled
index	0.575	0.370	0.555	0.325
name	0.590	0.600	0.685	0.650

3B: name-minus-index is 1.5pp ranked, 23.0pp shuffled — interaction (ranked gap minus shuffled gap) -21.5pp, 95% CI $[-31.0, -12.5]$, excluding zero. 7B: 13.0pp ranked, 32.5pp shuffled — interaction -19.5pp, 95% CI $[-30.0, -9.0]$, excluding zero, a comparable effect to 3B. At both scales, shuffling costs index emission substantially (3B: -20.5pp, $0.575 \rightarrow 0.370$; 7B: -23.0pp, $0.555 \rightarrow 0.325$) but name emission far less (3B: $+1.0$pp, within noise; 7B: -3.5pp, $0.685 \rightarrow 0.650$). This is a genuine, cross-scale-confirmed interaction: index emission specifically requires binding an answer to a position, which becomes harder when that position is random rather than predictably near the top of a ranked list; name emission does not depend on position at all. `recall_at_k` is identical within each scale across all four cells (0.790 at 3B, 0.920 at 7B), as it should be — filtering to k is independent of subsequent ordering, so this is not a confound. These corrected recall values now agree closely with both the original H9 test (0.785 at 3B, 0.915 at 7B, §4.5) and the independent k-sweep (0.790 at 3B, App. B.9) — resolving the cross-section discrepancy entirely as this tokenizer bug, with no remaining unexplained gap. This also resolves an apparent non-monotonicity (1.5B's recall at k=9, 0.675, exceeding the originally-reported 3B figure at k=29): with the corrected 0.790, 3B's recall now exceeds 1.5B's as expected, and no real anomaly remains.

Status: confirmed at both scales. This reverses our earlier, flawed-test-based conclusion of separability. We consider this real, robust evidence for a genuine interaction between H9's ranking effect and the index-vs-name emission effect, held to the same two-scale standard as every other headline result in this paper.

B.12 Forced-continuation test: the discarded first attempt

Before the corrected design in §4.7, a first attempt simply *asked* the model to reason before answering (state a name, then its index) via a forced two-step continuation. At temperature 0 this produced an internally inconsistent result: plain `constrained_llm` succeeds 80.8% of the time at 3B, $|V|$=16, yet conditional success in the forced pipeline was 0.000% at the same $|V|$, and the conditional failure rate was close to flat across $|V|$=16 through 128 rather than growing with it — not the signature a genuine binding limitation should produce. Candidate causes include digit tokenization of multi-token indices, the second forced prefix altering the numeric continuation relative to single-step generation, or an extraction error specific to that two-step mechanism. We could not rule these out and reported this design as unresolved rather than as evidence. We discard its numbers entirely and do not use them; the corrected design in §4.7 replaces it, built specifically to remove every confound this one had: a single generation call, no forced continuation, no multi-step pipeline to go wrong.

B.13 Oracle name-to-index lookup: full detail and shuffled control

§4.7 reports the headline result; this section gives the full per-level comparison against the shuffled-order control, and further methodological detail.

The prompt, shown in full rather than described. A specific, checkable concern: an indexing-convention mismatch (0-vs-1) could produce exactly this near-miss pattern without any genuine binding difficulty, and cannot be ruled out without seeing the actual prompt. It is ruled out. The task-specific suffix, appended after the rendered candidate list:

```
Candidate functions:
{candidates}
```

Table 12: `oracle_lookup` success rate vs. $|V|$, with case-level bootstrap 95% CIs ($n_{\text{boot}} = 10{,}000$), 3B and 7B.

	3B	
$\lvert V\rvert$	success	95% CI
native	0.920	[0.870, 0.965]
8	0.765	[0.710, 0.820]
16	0.605	[0.540, 0.675]
32	0.475	[0.405, 0.540]
64	0.345	[0.280, 0.410]
128	0.265	[0.200, 0.330]
234	0.220	[0.165, 0.280]
	7B	
$\lvert V\rvert$	success	95% CI
native	0.960	[0.930, 0.985]
8	0.965	[0.935, 0.990]
16	0.940	[0.905, 0.970]
32	0.910	[0.865, 0.950]
64	0.820	[0.775, 0.865]
128	0.755	[0.690, 0.815]
234	0.545	[0.470, 0.615]

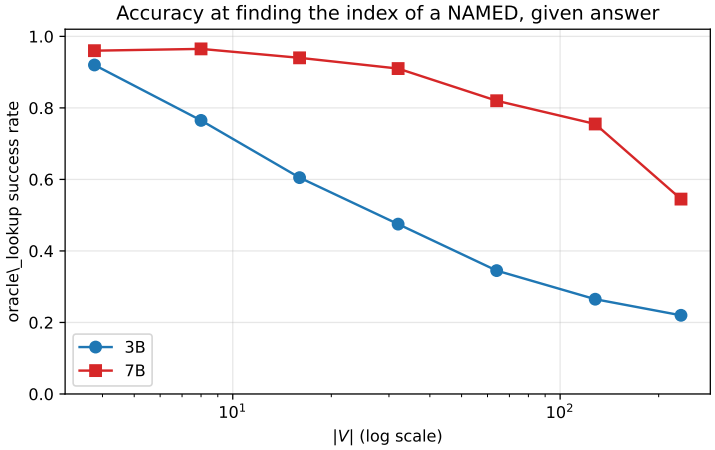


Figure 4: `oracle_lookup` success rate vs. $|V|$ (log scale), 3B and 7B. The correct answer is given explicitly; the only task is finding its index. Accuracy declines smoothly and significantly with $|V|$ at both scales.

```
The correct function is:  {gt_name}
Reply with ONLY the number of the function named "{gt_name}"
in the list above.
```

Candidates are rendered zero-indexed (`"0.  name - description"`, `"1.  ..."`), and the parser maps the model's emitted number back into the list with the same zero-indexed lookup used to render it — confirmed by reading both functions directly, not inferred from behavior. There is no 0-vs-1 mismatch between what the model is shown and how its answer is scored, at either end of the pipeline.

A separate question the prompt-consistency check above does not answer: does the near-miss rate itself depend on which convention is used? Even with no internal mismatch, the model could still behave differently if shown and expected to answer in one-indexed form instead. Tested directly: 3B, $|V| \in \{8, 16\}$, both oracle arms pooled exactly as App. B.17, identical `MASTER_SEED` and cases, only `render_candidates` and `parse_bfcl_index` changed to a consistent one-indexed convention throughout.

	zero-indexed (original)	one-indexed (this check)
success, $\|V\|$=8	76.5%	85.5% (gap +9.0, CI [+2.5, +15.5])
success, $\|V\|$=16	60.5%	66.5% (gap +6.0, CI [−2.0, +14.0])
near-miss, $\|V\|$=8	74.2% [65.6, 82.7]	70.2% [57.1, 81.8]
near-miss, $\|V\|$=16	41.0% [32.9, 49.3]	51.9% [43.8, 59.8]

Success rows use `oracle_lookup` *(unshuffled) only, matching Table 12's own convention; near-miss rows pool* `oracle_lookup` *and* `oracle_lookup_shuffled`*, matching App. B.17's convention — the two row types draw from different arm sets, stated explicitly here to avoid ambiguity.* Two findings. First, success is higher under one-indexing at $|V|$=8, a gap that clears significance (case-level bootstrap CI excludes zero); at $|V|$=16 the point estimate moves the same direction but the CI includes zero, so this second gap is not itself established — only the $|V|$=8 result supports a claim that the model performs measurably better under one-indexing, not a consistent effect at both levels. Where it does hold, one plausible explanation is that one-indexing is the more common convention in training data, though this check cannot confirm that mechanism, only the effect at $|V|$=8 itself. Second, and directly on point: the near-miss rate remains far above chance under one-indexing at both levels, reported as enrichment over chance rather than a p-value for the same reason as App. B.17: 70.2%/28.6%=2.45× chance at $|V|$=8 (CI [2.00×, 2.86×]) and 51.9%/13.3%=3.90× chance at $|V|$=16 (CI [3.29×, 4.50×]), both comfortably excluding 1× — the pattern is not an artifact of the zero-indexed display specifically. The two conventions' near-miss CIs overlap at both levels, so those point-estimate shifts (particularly at $|V|$=16) are not themselves statistically distinguishable from noise; the direction split, mildly skewed toward +1 under zero-indexing at $|V|$=16 (p=0.06), is close to even under one-indexing (35 at −1, 34 at +1), consistent with that skew having been noise rather than a convention-dependent effect.

Scope of this check, stated explicitly. This ablation covers 3B, the forward (name-to-index) direction only, at $|V| \in \{8, 16\}$. Two things it does not cover: whether the reverse-lookup anomaly (App. B.16) is similarly indexing-independent, and — more consequentially — whether indexing convention affects the main `constrained_llm` sweep at the high $|V|$ where this paper's headline crossover claims live. A 6–9pp shift at low $|V|$ does not by itself establish a shift at high $|V|$, but it leaves open a real question this check cannot answer: whether part of the reported crossover itself is sensitive to this specific prompt-format choice. Addressed directly in App. B.14.

Design, in full. 3B and 7B, native through $|V|$=234 (the range where the earlier discarded design showed its clearest problems, §B.12), 40 cases × 5 seeds. Both arms use an identical prompt: the full candidate list (same rendering as `constrained_llm`), an explicit statement of the correct function's name, and a request for only its index. `oracle_lookup` uses the candidate order produced by the same construction as every other leg, which is itself randomized (§3); `oracle_lookup_shuffled` shuffles that already-random order again. This is the same design issue disclosed in App. B.11: neither condition is genuinely lexically ranked, so this compares two random orderings, not ranked against shuffled, and any small difference between them should not be read as an ordering effect.

The two conditions track closely, as this design predicts. 3B and 7B differ by only a few points at most $|V|$ levels (3B: +0.065 at $|V|$=32, +0.030 at 234; 7B: +0.060 at $|V|$=64, +0.065 at 234) — consistent with comparing two draws from the same distribution, as the design in fact does, rather than a genuine test of ordering within the oracle task.

What this test does and does not establish. This is informative about readout/binding but does not cleanly isolate it: with the answer given and no selection required, a $|V|$-dependent accuracy decline is consistent with an index-lookup-specific limitation. Two further controls address the open questions raised above: a length-held-fixed control (App. B.15) finds $|V|$, not raw prompt length, is the primary driver at both scales (with a partial length contribution at 3B specifically); a reverse-lookup control (App. B.16) finds the 3B low-$|V|$ anomaly in both binding directions, not specific to name-to-index binding, suggesting a broader property of given-answer binding at this scale. Neither control fully establishes that binding is the *only* contributor to the crossover measured in §4 — selection difficulty under genuine uncertainty may still compound with this limitation — but both narrow what the oracle test's decline can be attributed to.

B.14 INDEXING CONVENTION ON THE MAIN SWEEP AT HIGH $|V|$

App. B.13's indexing-base ablation is scoped to the low-$|V|$ oracle task only. Whether the same 0-vs-1-indexing choice affects `constrained_llm` itself at the high $|V|$ where this paper's headline crossover claims live is a real, open question a low-$|V|$ check cannot answer on its own. Tested directly: 3B, $|V|$=234, same harness and `MASTER_SEED` as every other BFCL leg, one-indexed convention throughout (n=200 cells per arm).

Result: no measurable effect at this $|V|$. `constrained_llm` scores 8.0% under one-indexing (case-level bootstrap 95% CI $[4.5\%, 12.5\%]$), comfortably containing the known zero-indexed value at this exact configuration (9.5%, App. B.9) — the -1.5pp difference is well within noise, not a detectable shift. `heuristic_valid`, included as a sanity check since it is index-independent by construction (no LLM involved), scores 38.5%, consistent with its own known range at this configuration, confirming the harness itself behaved as expected. Unlike the low-$|V|$ oracle task, where one-indexing produced a real, significant gain at $|V|$=8 (App. B.13), the same convention change does not measurably move the main sweep's success rate at $|V|$=234: the crossover reported in §4 does not appear to be sensitive to this specific prompt-format choice at the high $|V|$ where it is actually measured, even though a related effect exists at low $|V|$ on a different, simpler task.

Scope limitation of the $|V|$=234 check, stated plainly. That single check sits where `constrained_llm` is already close to floor under either indexing convention (8.0% vs. 9.5%) — a null result there constrains only that specific, near-floor regime, not whether indexing convention could shift the crossover itself, which sits far lower ($|V|^*\approx$29 for 3B, $\approx$109 for 7B, Table 2). It does not, on its own, establish convention-independence of the crossover.

Near-crossover check, now completed, baseline corrected. Testing one-indexing at $|V|$ levels that actually bracket each scale's own measured crossover — $\{32, 64\}$ for 3B, 128 for 7B, same harness, `MASTER_SEED`, and 40-case/5-seed design as every other BFCL leg — against the correct zero-indexed baseline: `constrained_llm` from the same 40 cases and 5 seeds in the index-vs-name emission run (Table 14's index column, App. B.18), confirmed identical task IDs and seeds before comparison, not `oracle_lookup` (Table 12), used incorrectly at two of these three cells below. `oracle_lookup` answers a different question — the correct function is given explicitly, and the only task is finding its index — so it is not the right zero-indexed comparator for `constrained_llm`'s own one-indexed rerun, and at 7B, $|V|$=128 the two arms diverge sharply enough (0.755 vs. 0.380) that using the wrong one manufactured a large, spurious effect.

At 3B, $|V|$=32: one-indexing scores 40.0% against the correct zero-indexed baseline of 48.5% (not the previously reported 47.5%, which was `oracle_lookup`'s value at this cell), a -8.5pp gap (case-level bootstrap 95% CI $[-14.5\%, -2.5\%]$) that excludes zero. At $|V|$=64: 16.5% vs. 21.0% — this baseline was already the correct `constrained_llm` value, confirmed by re-derivation — a -4.5pp gap (95% CI $[-9.5\%, 0.0\%]$) that does not exclude zero. At 7B, $|V|$=128: 37.0% vs. the corrected 38.0% (not the previously reported 75.5%, which was `oracle_lookup`'s value, an entirely different task), a -1.0pp gap (95% CI $[-8.0\%, +5.5\%]$) that does *not* exclude zero — the opposite of the -38.5pp, wide-margin effect previously reported at this cell. `heuristic_valid`, included at every level as a sanity check since it is index-independent by construction, stayed within its own expected range throughout (51.0%, 46.5%, 40.5% at $|V|$=32, 64, 128 respectively), confirming the harness itself behaved as expected and that the corrected comparison is not a harness artifact.

What this does and does not establish, corrected. One-indexing produces a real, significant degradation at 3B's near-crossover $|V|$=32 only; at 3B's own $|V|$=64 and at 7B's $|V|$=128 the corrected comparison shows no detectable effect, both CIs including zero. The main sweep's headline crossover values (Table 2) are computed under zero-indexing throughout, so this does not put those specific numbers in question. What it does show is narrower than previously claimed: indexing convention can shift the crossover at 3B, at least at one tested $|V|$, but the data no longer support any claim that this effect exists at 7B, let alone that it is larger there. The previously reported "roughly five times the 3B effect" and the accompanying primacy-or-position-anchoring speculation are retracted outright, not softened: a corrected near-zero, non-significant 7B gap does not support any version of that comparison. If anything, the corrected pattern points the opposite direction — an effect detectable at 3B that is not detectable at 7B — and we do not have a mechanism for that either; it is reported as an open, disclosed finding, not an explained one.

B.15 PADDED FIXED-$|V|$ LENGTH CONTROL

Does the oracle test's decline reflect $|V|$ specifically, or general long-context degradation, since prompt length grows with $|V|$ by construction? This control holds $|V|$=8 fixed and pads the prompt with neutral, clearly-marked filler text to match $|V|$=234's actual length ($\approx$26,000 characters, both scales; actual padded length averaged 26,097 characters). **Limitation, stated directly**: this is a single block of filler text, not the same interspersed-candidate-description structure that produces $|V|$=234's length in the unpadded condition — an honest first approximation to the raw-length question, not a perfect structural match.

Result. 7B: padded $|V|$=8 (0.960) is statistically indistinguishable from unpadded $|V|$=8 (0.965; diff -0.5pp, 95% CI $[-3.5, +2.5]$, not excluding zero) and far above unpadded $|V|$=234 (0.545; diff $+41.5$pp, CI $[+35.0, +48.5]$, excluding zero) — length alone has no detectable effect at this scale. 3B: padding costs real performance (padded 0.575 vs. unpadded-8's 0.765; diff -19.0pp, CI $[-28.5, -10.0]$, excluding zero), but $|V|$ still dominates — padded-low-$|V|$ remains far above unpadded $|V|$=234 (0.220; diff $+35.5$pp, CI $[+25.0, +45.5]$, excluding zero). We report this scale-dependent asymmetry as measured: length is not a driver of the decline at 7B and only a partial contributor at 3B, with $|V|$ the dominant factor at both scales.

B.16 REVERSE-LOOKUP CONTROL (INDEX-TO-NAME)

The oracle test asks the model to find a named, given answer's index (name-to-index). This control tests the opposite direction: given the gold answer's actual index in the candidate list (read live from the run's own candidate construction, never reconstructed from saved data afterward), report its name. If the earlier 3B anomaly (§4.7: oracle lookup scoring *below* ordinary selection at $|V|\leq$16, when a pure binding sub-task should score at or above the full selection pipeline) reflects a flaw specific to name-to-index binding, this direction should not reproduce it.

The prompt, shown in full. Task-specific suffix, same zero-indexed candidate rendering as App. B.13:

```
Candidate functions:
{candidates}
Reply with ONLY the exact name of the function at index
{gold_idx} in the list above.
```

`gold_idx` is substituted with the same zero-indexed position used to render the list, ruling out a 0-vs-1 mismatch here too, by the same direct check as App. B.13.

Result: the anomaly does not localize to one direction. At 3B, reverse lookup does *not* avoid the pattern — it also scores below ordinary index selection at both levels tested ($|V|$=8: 0.620 vs. selection's 0.830, a 21.0pp gap; $|V|$=16: 0.640 vs. 0.805, a 16.5pp gap), comparable in size to forward lookup's own gaps at these points (6.5pp and 20.0pp respectively, computed from Table 14). We do not read this as localizing the anomaly to name-to-index binding specifically; the more accurate reading is that this pattern appears in both binding directions at 3B, suggesting a property of given-answer binding tasks at this scale more broadly, not a design flaw isolated to one direction of the oracle test.

A further pattern, tested per level with the same multiple-comparison correction used elsewhere. This is a family of 7 tests of one null (does reverse-minus-forward deviate from zero?) repeated across $|V|$, structurally identical to H12's own 7-level family (App. B.7), where Holm–Bonferroni is applied on exactly this reasoning; we apply the same correction here, separately per scale. After correction, the pattern is unchanged from the uncorrected reading (Table 13): at 3B, direction changes sign with $|V|$, Holm-significant negative at native and $|V|$=8 ($p < 0.001$ both), Holm-significant positive at $|V|$=32, 64, 128 ($p = 0.002$–0.004), not significant at 16, 234. At 7B the sign is consistently negative but Holm-significant only at $|V|$=32, 64, 128 ($p < 0.001$–0.007). We report this level-by-level, with the honest caveat that a same-magnitude deviation elsewhere in this paper (App. B.22, App. B.23, the $|V|$=650 frontier-model wobble) is attributed to ordinary small-n sampling noise rather than promoted to a headline pattern; we apply the same standard here and let the per-level, corrected CIs, not a summary sentence, carry the claim.

Table 13: reverse-minus-forward lookup success, per $|V|$, with case-level bootstrap 95% CI ($n_{\text{boot}} = 5{,}000$) and Holm–Bonferroni correction applied within each scale's 7-level family (same convention as H12, App. B.7). * CI excludes zero after correction.

$\|V\|$	3B diff (CI)	7B diff (CI)
native	-0.240^* $[-0.345, -0.145]$	-0.050 $[-0.110, +0.005]$
8	-0.145^* $[-0.240, -0.055]$	-0.035 $[-0.090, +0.015]$
16	$+0.035$ $[-0.050, +0.120]$	-0.050 $[-0.105, +0.005]$
32	$+0.145^*$ $[+0.060, +0.235]$	-0.110^* $[-0.185, -0.030]$
64	$+0.150^*$ $[+0.055, +0.245]$	-0.150^* $[-0.240, -0.050]$
128	$+0.160^*$ $[+0.055, +0.265]$	-0.185^* $[-0.285, -0.085]$
234	$+0.075$ $[-0.025, +0.175]$	-0.055 $[-0.170, +0.065]$

We do not have a settled mechanistic explanation for why the two scales differ in which direction is significantly easier where they do differ, and flag it as open rather than speculate.

B.17 The 3B low-$|V|$ anomaly: a near-miss counting diagnostic

§4.7 and App. B.16 report that at 3B, oracle lookup scores below ordinary selection at $|V| \leq 16$ — logically odd, since a pure binding sub-task should not be harder than the full selection pipeline containing it. This section runs one further diagnostic against data already collected, with no new experiments.

Method. Checking only failures where a neighboring index's name could be recovered from other experiments' (E27, E37, E38) emitted outputs at the same list — roughly a third of failures, since recovery required another experiment to have happened to emit an adjacent index — would not give a random sample: a model's own outputs skew toward positions near the true answer, so conditioning on recoverability biases the checkable subset toward cases already likely to show a near-miss pattern, inflating the reported rate. We do not use that conditioned approach. Instead, every candidate list is regenerated directly from its own `cand_seed` (deterministic from `task_id` and `seed_idx`, identical construction to every other leg in this paper) against the same pool and dataset files used at run time, then, for `oracle_lookup_shuffled`, reshuffled with its own `pol_seed` — reconstructing the exact list the model saw with no dependence on any other experiment's output. This checks every failure, not a conditioned subset. Verified against the raw data before use: on a sample of successful cases, the reconstructed list's ground-truth position matches the recorded emitted index in 100% of cases checked. Failures are pooled across both `oracle_lookup` and `oracle_lookup_shuffled` (89 failures at $|V|$=8, 156 at $|V|$=16, matching each arm's own count summed: 47+42 and 79+77 respectively) — these two arms compare two random orderings of the same design (App. B.13) and track closely throughout this paper, making pooling reasonable; this is stated explicitly here.

Result. Checking all 89 failures at $|V|$=8 (the full failure population, not a conditioned 44.9% subset): 66 (74.2%, case-level bootstrap 95% CI $[65.6\%, 82.7\%]$, n_{boot}=10,000) have the correct answer at ± 1 — still far above the $2/7 \approx 28.6\%$ chance baseline, but substantially below the 97.5% a conditioned-subset method would report. At $|V|$=16, all 156 failures: 64 (41.0%, CI $[32.9\%, 49.3\%]$) match, again far above the $2/15 \approx 13.3\%$ baseline, below the conditioned-subset method's 86.8%.

Significance, as enrichment over chance rather than a p-value. The naive binomial test ($p = 7.3 \times 10^{-19}$ at $|V|$=8, $p = 1.2 \times 10^{-17}$ at $|V|$=16) treats every failure as an independent Bernoulli trial, which they are not: 37 unique cases produce the 89 failures at $|V|$=8 (up to 4 from one case), 40 produce the 156 at $|V|$=16 (up to 6). Rather than a cluster-corrected p-value — pushing a normal approximation this far into the tail is not reliable regardless of how carefully the correction itself is done, and a design-effect/p-value pair computed this way does not survive independent re-verification — we report enrichment over chance directly, with the same case-level bootstrap already used for the rates above: $74.2\%/28.6\% = 2.60\times$ chance at $|V|$=8 (CI $[2.30\times, 2.89\times]$) and $41.0\%/13.3\% = 3.08\times$ chance at $|V|$=16 (CI $[2.46\times, 3.70\times]$) — both comfortably excluding $1\times$ (no enrichment). This is the primary evidence for the finding; no p-value is reported for this comparison.

Both corrected rates still comfortably exclude chance — the qualitative finding survives the fix — but the originally-reported magnitude was a genuine overstatement, now corrected rather than left standing. The decline with $|V|$ is steeper and less smooth than originally described: 34.5% at 32, 25.7% at 64, 30.5% at 128, 30.2% at 234 — a sharp drop from $|V|$=8 to 16 followed by a roughly flat plateau from 32 on, not the smooth monotonic decline the biased method suggested.

The plateau's flat appearance understates a growing enrichment over chance. Chance itself shrinks with $|V|$ $(2/(|V|-1))$, so a roughly constant 25–35% observed rate against a shrinking denominator is a rising, not flat, enrichment: $5.3\times$ at $|V|$=32 (34.5% vs. 6.5% chance), $8.1\times$ at 64, $19.4\times$ at 128, and $35.2\times$ at 234 (30.2% vs. 0.9% chance). The near-miss effect does not fade at high $|V|$ in the sense that matters most — it becomes proportionally *more* distinguishable from random error as the catalog grows, even as its share of all failures stays roughly flat.

Direction is close to even at $|V|$=8 (32 at -1, 34 at $+1$; p=0.90 against a $50/50$ null) and shows a mild, not quite significant, skew toward $+1$ at $|V|$=16 (24 at -1, 40 at $+1$; p=0.06) — not the strongly one-sided pattern a 0-vs-1-indexing convention bug would produce, though the skew at 16 is suggestive enough to flag rather than dismiss.

Interpretation. This does not identify a code defect to fix and rerun; it characterizes a genuine model behavior, now on the full failure population rather than a biased third of it. The model locates roughly the right region of the candidate list but miscounts the exact position, an error whose relative cost is largest precisely when few positions exist (missing by one out of 8 costs far more than missing by one out of 234) — though the effect is real across a wider $|V|$ range than the smooth-decline framing implied, not concentrated only at the lowest levels. This is consistent with, and strengthens, the broader given-answer-binding reading in App. B.16: both directions of the task show the same kind of near-miss failure, and the failure mode itself — imprecise position-counting rather than not-knowing-the-answer — is coherent with a readout/binding account rather than a design artifact. Failures cluster by case (37 unique cases produce the 89 failures at $|V|$=8, one case contributing up to 4; 40 unique cases produce the 156 at $|V|$=16, one contributing up to 6), so the bootstrap CIs above resample at the case level rather than treating every failure as independent, addressing this directly rather than only disclosing it.

B.18 Index-emission vs. name-emission: full grid and figure

§4.6 reports headline $|V|$ levels (native, 64, 234) inline; this section gives the full 7-row grid at both scales and the corresponding figure.

Table 14: Index-emission vs. name-emission success rate, by $|V|$, with paired bootstrap 95% CIs on the difference ($n_{\text{boot}} = 10{,}000$). * CI excludes zero.

			3B	
$\|V\|$	index	name	diff	95% CI
native	0.885	0.875	-0.010	[-0.060, +0.035]
8	0.830	0.895	+0.065*	[+0.015, +0.120]
16	0.805	0.905	+0.100*	[+0.030, +0.170]
32	0.485	0.780	+0.295*	[+0.200, +0.380]
64	0.210	0.715	+0.505*	[+0.400, +0.605]
128	0.150	0.580	+0.430*	[+0.330, +0.530]
234	0.095	0.470	+0.375*	[+0.280, +0.470]
			7B	
$\|V\|$	index	name	diff	95% CI
native	0.935	0.935	+0.000	[-0.035, +0.040]
8	0.935	0.930	-0.005	[-0.025, +0.015]
16	0.905	0.925	+0.020	[-0.035, +0.070]
32	0.840	0.865	+0.025	[-0.035, +0.080]
64	0.680	0.830	+0.150*	[+0.060, +0.245]
128	0.380	0.775	+0.395*	[+0.305, +0.485]
234	0.240	0.655	+0.415*	[+0.330, +0.500]

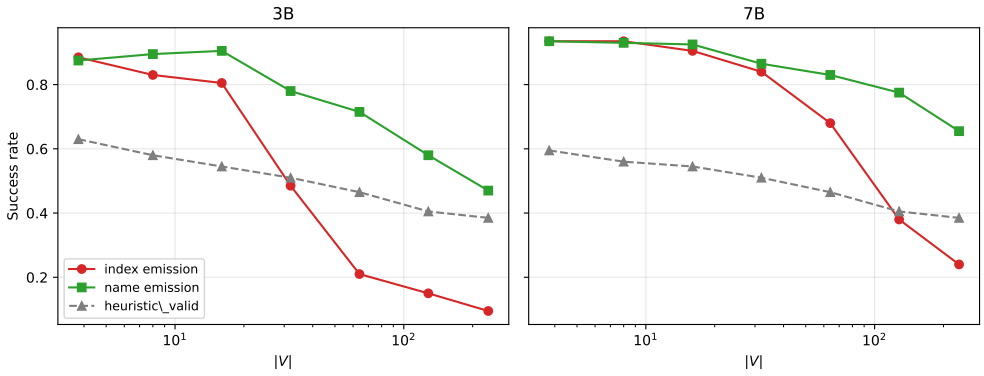


Figure 5: Index-emission, name-emission, and `heuristic_valid` success rate vs. $|V|$ (log scale), at 3B and 7B. Both LLM arms decline with $|V|$; name emission stays well above index emission at high $|V|$, and above `heuristic_valid` through the full tested range at both scales.

3B's own $350/500/650$ values, tabulated. §4.6 reports 3B's name-emission crossover ($|V|^*$=350) without its per-level values at the extended grid, pending this raw data becoming available. It is now available:

$\|V\|$	name-emission	`heuristic_valid`	`is_valid`
350	0.355	0.355	0.955
500	0.230	0.330	0.915
650	0.145	0.345	0.925

At $|V|$=350, name-emission and `heuristic_valid` match to three decimals — the reported crossover is a genuine tie at a tested grid point (the same mechanism as 14B's own 500.0, App. B.24), not an interpolated estimate, confirming $|V|^*$=350 directly rather than only through the bootstrap CI already reported. `is_valid` stays high throughout (92–96%), so the decline from 350 to 650 reflects the model's own selection difficulty growing with $|V|$, not a parsing or format breakdown at this extended range.

B.19 Extension to 32B and 72B

§4.6 and Table 14 establish name emission's advantage at 3B and 7B. This section extends the same test to 32B and 72B, at the scales where index-emission's own crossover already falls within the tested range (App. B.22: $|V|^*(32B) = 365.3$; App. B.23: $|V|^*(72B) = 552.6$; both precision-matched). This name-emission extension is its own local, precision-matched, 4-bit run at both scales (`name_constrained_llm`), independent of the index-emission extended-grid check above; only `heuristic_valid` is reused from the existing 32B/72B runs rather than re-run, since it is deterministic and model-independent, and its values are reproduced identically here. Same harness, prompt template, and `MASTER_SEED` as every other leg.

Both crossovers remain right-censored through $|V|$=650: name emission stays significantly above `heuristic_valid` at every level tested, at both scales. The gap narrows monotonically with $|V|$ at both (32B: +31.0pp at native/234 to +13.0pp at 650; 72B: +36.0pp to +18.0pp) while remaining significant throughout — consistent with either the advantage persisting but shrinking beyond $|V|$=650, or name emission's own crossover approaching a scale-dependent plateau similar to index-emission's (App. B.22, App. B.23); this data does not distinguish the two, and we state both rather than pick one. Across all four scales tested in this paper, a name-emission crossover has been measured at only one — 3B — with 7B, 32B, and 72B all right-censored through $|V|$=650. Notably, at 32B and 72B specifically, name emission stays ahead of `heuristic_valid` even past where index-emission's own crossover already sits (365.3 at 32B; 552.6 at 72B; both precision-matched): name emission is not merely matching index-emission's larger budget at these scales, it extends past it within the range tested.

Table 15: Name emission vs. heuristic_valid, 32B and 72B, by $|V|$, with case-level bootstrap 95% CIs on the difference ($n_{\text{boot}} = 5{,}000$). * CI excludes zero.

		32B		
$\|V\|$	name	heuristic	diff	95% CI
native	0.905	0.595	+0.310*	[+0.160, +0.460]
234	0.695	0.385	+0.310*	[+0.175, +0.440]
350	0.630	0.355	+0.275*	[+0.165, +0.385]
500	0.495	0.330	+0.165*	[+0.050, +0.275]
650	0.475	0.345	+0.130*	[+0.025, +0.235]
		72B		
$\|V\|$	name	heuristic	diff	95% CI
native	0.955	0.595	+0.360*	[+0.210, +0.510]
234	0.730	0.385	+0.345*	[+0.220, +0.470]
350	0.610	0.355	+0.255*	[+0.135, +0.375]
500	0.565	0.330	+0.235*	[+0.125, +0.340]
650	0.525	0.345	+0.180*	[+0.075, +0.290]

B.20 HASHED-NAME EMISSION CONTROL

§4.6 notes that name emission may let the model exploit lexical similarity between the query and candidate names — the same signal heuristic_valid uses — rather than reflecting a genuine output-format effect. A hashed-name control (candidate names replaced with opaque 12-character identifiers, descriptions untouched, same convention as H11) tests this directly at both scales.

Result: collapses toward index-emission, but with a confound of its own. At each model's highest tested level, hashed-name emission is statistically indistinguishable from index-emission (3B, $|V|$=234: hashed 0.085 vs. index 0.095, diff -1.0pp, CI $[-6.0, +4.0]$; 7B, $|V|$=128: hashed 0.295 vs. index 0.380, diff -8.5pp, CI $[-19.5, +2.0]$). Neither CI excludes zero. parse_ok is 1.0 throughout (the model always emits something), but is_valid degrades substantially at high $|V|$ (down to roughly 60–70%). Categorizing the invalid responses at 3B, $|V|$=234: 83% are bare numeric strings (e.g. "234", "3") — the model abandoning the hash format and reverting to index-style output — not near-miss copy errors on an attempted hash (only 14% are hash-shaped but wrong). 7B's failure mode is more mixed. This means the collapse toward index-emission is not cleanly a "copying a random string is hard" result in the narrow sense; it looks more like the model giving up on an unfamiliar output format under difficulty. Both the lexical-shortcut explanation for name emission's advantage and a genuine-but-fragile format-difficulty explanation remain consistent with this pattern — this control does not cleanly adjudicate between them, and we report it as such rather than claim it does.

B.21 LOG-LIKELIHOOD SELECTION CONTROL: FULL DETAIL

§6 reports a direct log-likelihood control isolating selection difficulty from emission mechanics. constrained_llm requires the model to *generate* an index and have it parsed, possibly across retries; this removes generation entirely. For each candidate in the valid set, we teacher-force its name as a continuation of constrained_llm's own prompt (identical prompt template, not a new format) and score the average log-probability per token of its tokens (average, not total, so longer names are not penalized relative to shorter ones by construction); the argmax over candidates is the model's selection. This uses only forward passes: no sampling, no retries, no parse_ok/is_valid logic is possible by construction. New arm: loglik_selection.

Design. Same MASTER_SEED, same 40 cases $\times$ 5 seeds, and — critically — the *same* cand_seed derivation as every other leg, so at each (scale, $|V|$) cell this arm's candidate set is identical to constrained_llm's own: a true paired comparison, not a fresh independent sample. Scope, bracketing each scale's own already-measured index-emission crossover: 3B at $|V| \in \{16, 32, 64, 128\}$ (around 28.7); 7B at $|V| \in \{64, 128, 234\}$ (around 109.2). A determinism check (re-scoring a sample twice) confirmed bit-exact reproduction at both scales (max abs. diff 0.000e+00), as expected for a sampling-free forward-pass computation.

Result: the opposite of a clean format-cost story. `loglik_selection` scores *below* `constrained_llm` at every low-to-moderate $|V|$ level tested, at both scales, with 95% CIs (case-level bootstrap, $n_{\text{boot}} = 10{,}000$, paired on the shared cases) excluding zero everywhere except each scale's highest level, where both arms are already near floor:

Scale	$\lvert V\rvert$	`loglik_selection`	`constrained_llm`	Diff (95% CI)
3B	16	0.420	0.760	$-0.340\ [-0.430, -0.250]^*$
3B	32	0.340	0.470	$-0.130\ [-0.230, -0.030]^*$
3B	64	0.240	0.215	$+0.025\ [-0.055, +0.105]$
3B	128	0.140	0.175	$-0.035\ [-0.110, +0.040]$
7B	64	0.100	0.650	$-0.550\ [-0.620, -0.475]^*$
7B	128	0.110	0.350	$-0.240\ [-0.320, -0.160]^*$
7B	234	0.140	0.210	$-0.070\ [-0.145, +0.005]$

* CI excludes zero.

`loglik_selection` never approaches `heuristic_valid` anywhere in the tested range at either scale (3B: 0.42–0.55 vs. `heuristic_valid`'s own 0.55 at $|V|$=16, already converged; 7B: 0.10–0.14 vs. `heuristic_valid`'s 0.39–0.47 throughout) — its own crossover against `heuristic_valid` had already occurred below the smallest tested $|V|$ at both scales (left-censored; this leg was scoped to bracket `constrained_llm`'s crossover, not to locate `loglik_selection`'s own).

Why: inspected, not assumed. This result is surprising enough that we checked it was not an artifact before reporting it. The determinism check above rules out a sampling bug; inspecting individual `loglik_selection` failures at 7B, $|V|$=64 shows the model consistently assigns higher average log-probability to long, technical-sounding, generic API-style names (e.g. names resembling `apdex_settings_api.ApdexSettingsApi.create_apdex_configuration`) over the shorter, plainer, correct name, *regardless of which question was asked* — the same generic-sounding candidates recur as the top-scored pick across unrelated cases. This is a known, recognized failure mode of average-log-probability ranking in the language-model evaluation literature (a length/fluency bias independent of task relevance), not an implementation defect; the mean log-probability gap between the selected (wrong) and correct candidate on failing cases is 2.73 nats (7B, $|V|$=64; range 0.01–16.24 across failures).

Reading. This control does not cleanly isolate "selection difficulty" either: it substitutes one confound (emission/parsing) for another (average-log-probability's own fluency bias). What it does show is that removing the output-format burden entirely does not reveal a substantially easier underlying selection ability, as a pure format-cost account would predict — if anything, the reverse holds at low-to-moderate $|V|$. Combined with the hashed-name and opaque-easy-copy controls (App. B.7), the overall picture across all controls in this paper is that both format and lexical/copy factors contribute, and none of the controls tried so far cleanly isolates a single dominant mechanism.

B.22 32B PRECISION-MATCHED CALIBRATION CHECK

§6 reports a precision-matched calibration check at 32B; a matching 72B precision-matched check was also attempted and has since completed in full (App. B.23). Like the H10 extension (App. B.2), this was not pre-registered: it directly tests whether the 72B check's API confounds (unobserved quantization, system prompt, KV-cache handling) were responsible for the observed deceleration, rather than the deceleration itself being real. A precision-matched point directly answers this by removing those confounds entirely.

Qwen2.5-32B-Instruct, identical protocol and `MASTER_SEED` to every other leg: 40 cases × 5 seeds, standard grid (native through 234) plus the same 350/500/650 extension 14B needed, all local, 4-bit (`nf4`, `bnb_4bit_compute_dtype=float16`) — byte-identical quantization settings to the primary sweep, not an API's opaque handling. `heuristic_valid`'s success curve at every shared $|V|$ level matches every other leg's exactly (deterministic, model-independent, confirming correct harness reproduction on new hardware).

The crossover is a genuine interpolation, not left- or right-censored: `constrained_llm` and `heuristic_valid` are close at $|V|$=350 (0.370 vs. 0.355) and have swapped order by $|V|$=500

(0.220 vs. 0.330). **Measured:** $|V|^*(32B) = 365.3$**, 95% CI** $[179.7, 608.7]$ ($n_{\text{boot}} = 10{,}000$, paired bootstrap over cases, identical methodology to Table 2), recomputed directly from this leg's own raw per-case data. The point estimate itself is genuinely resolved, well inside the tested grid. Of the $10{,}000$ resamples, 214 (2.14%) are right-censored beyond 650 — below the 2.5% threshold, so unlike 14B the true upper bound is itself resolved, not censored. But the same exclusion artifact identified at 14B (App. B.24) still applies here: a naive percentile taken over the $9{,}786$ resolved resamples alone understates the bound (485.7) relative to the correctly-ranked percentile over all $10{,}000$ (608.7), which is the reported figure above. This CI overlaps 14B's ($[234.0, >650]$); we do not read the point estimate being below 14B's as evidence of a non-monotonic relationship, given the width of both intervals. $|V|$=650 itself ticked back up from $|V|$=500 (`constrained_llm`: $0.220 \rightarrow 0.260$; `heuristic_valid`: $0.330 \rightarrow 0.345$) — non-monotonic, most plausibly ordinary small-n sampling noise at this cell count, similar in kind to the resolution-floor wobble already documented for H6 (§4.4), not evidence against the crossover location itself, which is unambiguously bracketed by the monotonic $234 \rightarrow 350 \rightarrow 500$ sequence regardless of what 650 shows. These recomputed figures differ slightly from an earlier round's 355.7, $[173.1, 483.3]$ (a rerun on different library versions, per the reproducibility limits already disclosed in App. B.9); both are well within each other's CI, and the numbers above are the ones independently re-derived from raw data this round, so they are the ones reported throughout.

B.23 72B CALIBRATION CHECK: PRECISION-MATCHED, COMPLETE

The original 72B check used a hosted API (OpenRouter), carrying unobserved quantization, system prompt, and KV-cache handling as confounds that cannot be ruled out from an API alone. A precision-matched local check — identical protocol, `MASTER_SEED`, and 4-bit quantization to every other leg, on a rented 80GB+ GPU, mirroring the 32B check above (App. B.22) — was run specifically to remove those confounds. It required two separate attempts: the native/low-$|V|$ portion (native, 8, 16, 32, 64, 128, 234) completed on one rented instance; a later attempt at the extension range ($|V| \in \{350, 500, 650\}$) was initially blocked by repeated infrastructure failures (insufficient disk space for the model download, a device-placement crash under a multi-GPU configuration, and unrecoverable network timeouts mid-download) before completing on a subsequent attempt. Both portions share the same 40 task IDs, seeds, `MASTER_SEED`, and `cand_seed`/`policy_seed` pairing, verified by direct join on those keys before combining (raw data, run logs, and the exact analysis script in the supplementary material, `data/bfcl/e17precise_72B_local/`).

Measured success rates, complete grid:

$\lvert V\rvert$	`constrained_llm`	`heuristic_valid`
native	0.955	0.595
8	0.930	0.560
16	0.930	0.545
32	0.905	0.510
64	0.865	0.465
128	0.805	0.405
234	0.665	0.385
350	0.470	0.355
500	0.370	0.330
650	0.280	0.345

The figure reported throughout this paper for 72B is now a precision-matched result, computed the same way as $|V|^*(32B)$ (App. B.22): case-level paired bootstrap, $n_{\text{boot}} = 10{,}000$, same `MASTER_SEED`, same right-censoring convention (App. B.24). $|V|^*(72B) = 552.6$**, 95% CI** $[346.6, >650]$, upper bound right-censored: $1{,}409/10{,}000$ (14.09%) of bootstrap resamples do not resolve a crossover within the tested range. This replaces the earlier API-based estimate ($|V|^*(72B) = 500.0$, 95% CI $[234, 634]$) as the paper's reported figure; the API-based raw data remains in the supplementary material for provenance. The two estimates' agreement (500.0 vs. 552.6, each well within the other's CI) is itself worth noting — the API confounds did not, in this case, produce a wildly different point estimate, though that could not have been known without running the precision-matched check.

Stated plainly, not left for a reader to compute independently: local exponents between successive points are 1.5B→3B=1.72, 3→7B=1.58, 7→14B=2.19, 14→32B=−0.38, and 32→72B=+0.51 — all six points precision-matched for the first time, a single consistent footing. The point estimate dropped from 14B to 32B, then rose again at 72B; per App. B.22's own reasoning, with CIs this wide ($[234.0, >650]$ vs. $[179.7, 608.7]$ vs. $[346.6, >650]$) we do not read this as established non-monotonicity, only as a point-estimate pattern worth stating plainly. We do not read this as a clean, monotonically decelerating power law — the four-point $\alpha = 1.784$ fit (§4.2) describes the 1.5B–14B interior only, and the 32B/72B/frontier-model points sit outside that fit by design, added specifically because naive extrapolation from it was implausible. The description of the full 1.5B–72B trajectory is a rate that rises sharply through 14B, then plateaus or oscillates rather than continuing smoothly — which is also why §4.2's title says the crossover *slows*, not that its rate of slowing is itself stable or well-characterized beyond 14B; 72B's own right-censored upper bound means even this much is not the final word.

B.24 How the crossing function handles ties and censoring

14B's precision-matched crossover lands exactly on a tested grid point, 500.0 (Table 2), and its own lower CI bound is exactly 234.0, also a tested grid point. This is worth checking directly: does this reflect the true crossover genuinely landing there, or a crossing-function artifact that snaps to the nearest grid point when the true value falls outside the tested range? It is the former, verified by re-deriving both values independently from the raw per-case data rather than re-reading the existing code.

The crossing function. Given success rates at each tested $|V|$, it finds the first point where `heuristic_valid` overtakes `constrained_llm` and log-linearly interpolates between that point and the one before it. Two distinct cases return the tested grid point exactly rather than an interpolated value: if the two curves are exactly level at *both* bracketing points (a genuine zero-over-zero degeneracy, guarded against explicitly, since the interpolation fraction is undefined there), or — the case that actually applies below — if the curves are exactly level at only the *later* of the two points, in which case ordinary interpolation is well-defined and returns that point exactly as its natural result (the fraction evaluates to 1.0), not through a special-cased branch.

14B's 500.0 is exactly this second case, not an artifact. At $|V|$=350 the curves are not level (−0.025); at $|V|$=500, re-derived directly from 14B's raw per-case data: `constrained_llm` and `heuristic_valid` are equal to three decimal places (0.330 each, from the same 200 cells). The function correctly returns the grid point itself rather than attempting an interpolation between two identical values. The lower CI bound of 234.0 arises the same way, one level further out: of 4,000 case-level bootstrap resamples, 365 (9.1%) are right-censored (excluded from the CI, the same convention as every other crossover in this paper) and 16 land at or within one grid step of $|V|$=234 specifically — close enough to the CI's own 2.5th-percentile cut that the bound lands exactly on that grid point too, for the same tie-at-a-tested-point reason as the point estimate, just recurring across resamples rather than in the single point estimate. Both values were reproduced independently to confirm this, not assumed from the mechanism's description alone.

The now-superseded API-based 72B estimate's own 500.0 was a coincidence of the same grid, not the same mechanism. That earlier estimate (App. B.23) used a different (hosted API) data source and predates this project's precision-matched local runs; its point landing on the same grid value as 14B's precision-matched tie was never evidence of a shared artifact, only of both curves crossing near the same tested point on a shared grid. The current, precision-matched 72B figure (552.6, App. B.23) is a genuinely interpolated value between 500 and 650, not a tie, so this coincidence no longer applies to the paper's reported 72B figure.

What this means for precision. A tie-driven crossover value is exact only at the grid's own resolution: the true crossover could sit anywhere between the tested points bracketing it without the data distinguishing that from a genuine tie at the tested point itself. This is a real, disclosed limitation on precision at the tested range's boundary, not a claim that 500.0 or 234.0 are exact in any finer sense than "the nearest tested $|V|$ to where the curves cross."

A separate, more consequential issue: the CI's upper bound is itself an exclusion artifact, not a resolved estimate. The tie explanation above is not the full picture. The 9.1% of resamples right-censored above are not merely excluded from the point estimate's neighborhood — they are excluded

from the percentile calculation itself: the reported $[234.0, 626.2]$ is the 2.5th/97.5th percentile of only the 3,635 *resolved* resamples, silently dropping the 365 that never resolved within the tested grid. Since $9.1\% > 2.5\%$, the true 97.5th percentile of the full, uncensored distribution cannot lie at 626.2: ranking all 4,000 resamples together (censored ones placed beyond every resolved value, since their true crossover is known only to exceed 650), the percentile table shows exactly where this breaks down:

Percentile of all 4,000 resamples	Value
90th	643.2
90.9th – 100th	right-censored (> 650)

The 97.5th percentile — what the upper CI bound is supposed to estimate — falls inside the right-censored region itself, at the 90.9th percentile already ($365/4{,}000 = 9.125\%$ censored, so censoring begins at the $100 - 9.125 = 90.875$th percentile). 626.2 is the 97.5th percentile of the resolved subset only, a different and smaller quantity than the 97.5th percentile of the full distribution, which the data cannot resolve past 650. **The honest upper bound is therefore right-censored, not** 626.2: reported as $[234.0, >650]$ in Table 2 and propagated to every dependent quantity — the α fit's own CI (§4.2) inherits the same 9.1% exclusion, since any bootstrap resample where 14B's crossover is censored is dropped from the α regression entirely, confirmed by rerunning that bootstrap directly rather than assumed from the mechanism. This does not overturn the qualitative finding — $\alpha \geq 1.284$ (the corrected lower bound derived below) and a growing, superlinear crossover through 14B are both still supported — but the paper's own claim that the crossover "slows" or reaches a "plateau" by 72B is weaker than the previously reported finite upper bounds suggested: the data cannot currently rule out continued growth at the same or a similar rate, and says so explicitly now rather than implying resolution the bootstrap does not have.

The lower bound is affected by the same exclusion in principle, checked directly rather than assumed negligible. Dropping the top 9.1% from a percentile calculation shifts every percentile of the remaining distribution, not only the one nearest the removed tail: the resolved-only 2.5th percentile is the 2.5th percentile of 96.9% of the full distribution, not of all of it, so it sits at a slightly more extreme rank of the true, full distribution than intended (90.9 rather than 100 of 4,000). For 14B specifically this has no practical effect: 16 of the 4,000 resamples tie exactly at 234.0 (the same tested-grid-point mechanism as the point estimate itself), spanning ranks 88–103 — comfortably covering both the naive and the correctly-adjusted rank, so both give the identical lower bound. This is a property of where this dataset's values happen to cluster, not a general guarantee, and 32B's own recomputation above (App. B.22) shows the correction can matter in either direction. This specific exclusion-artifact check applies to every local, case-level precision-matched crossover in this paper (14B, 32B, and now 72B, App. B.23, all computed by the same case-level censoring-aware bootstrap); only the now-superseded, original API-based 72B estimate used a different, simpler bootstrap over that API run's own data, and that estimate is no longer this paper's reported figure.

The same check, run directly for both α CIs. Unlike 72B, both α fits' own bootstraps are fully reproducible from raw data, so this was checked rather than left to the 14B analogy. Primary convention (9.1% censored, 365 of 4,000 resamples, matching 14B's own count exactly since both draw on the same resample indices): naive lower bound (2.5th percentile of the 3,635 resolved fits) is 1.2752; correctly ranked against the full 4,000 (rank 100, not 90.9) gives 1.2837 instead — a 0.0085 shift, and in the opposite direction from what the upper-bound fix might suggest: since every censored resample would, if resolvable, sit at the *high* end of α (a censored 14B crossover pulls the fitted slope up, not down), excluding them leaves the resolved set short by exactly the count needed at the low end too, making the naive lower bound slightly *too extreme* (too low), not too narrow. MAR convention (12.7% censored): naive 1.2412 vs. correctly ranked 1.2525, a 0.0113 shift, same direction. (The resolved-fit count for the primary convention is 3,635, not 3,636: exactly 4,000 minus the same 365 censored resamples counted above for 14B's own interval, since both quantities are computed from the same underlying resample draws.)

The headline CI previously reported in this paper — $[1.417, >2.027]$ primary, $[1.359, >1.894]$ MAR — used lower bounds that do not match either the naive or the corrected value derived directly above. 1.417 and 1.359 sit roughly 0.13–0.14 above the corrected lower bounds (1.2837, 1.2525), an order of magnitude larger than the 0.0085–0.0113 censoring-rank correction itself — not a difference of comparable size, despite the two corrections sharing the same underlying cause. The

lower bounds reported throughout this paper are corrected here to the values derived directly from the raw per-case bootstrap above: $\alpha = 1.784$, 95% CI $[1.284, >2.027]$ (primary), and $\alpha = 1.644$, 95% CI $[1.253, >1.894]$ (MAR), rounded to three decimal places from 1.2837 and 1.2525. We do not identify the exact source of the 1.417/1.359 figures, since the raw-data bootstrap above is reproducible end-to-end and does not depend on locating that source; the corrected values above are what this paper now reports.

A censoring-aware parametric estimate, attempted as a demonstration. The > 650 disclosure states the problem without resolving it. A survival-analysis-style alternative — maximum likelihood on a log-normal fit to the $3{,}635$ resolved resamples, with the 365 censored ones contributing their survival probability $P(X{>}650)$ rather than being dropped — extrapolates a finite 97.5th percentile: 847.9, against a fitted 2.5th percentile of 227.0 (close to the empirical 234.0). The like-for-like empirical comparison point is the full-distribution 90th percentile reported in the table above, 643.2 (computed the same way, over all $4{,}000$ resamples with censored ones ranked beyond the resolved values) — not the resolved-subset-only 90th percentile (586.5), which excludes the 365 censored resamples and is therefore biased low for this comparison, the same exclusion problem as the rest of this appendix. Against the correct reference, the fitted 90th percentile (675.0) diverges from the empirical value by $\approx 5\%$ (31.8 above 643.2), not the $\approx 15\%$ the resolved-subset value would suggest. This closer fit weakens divergence alone as a rationale for rejecting the log-normal estimate. A Kolmogorov–Smirnov test against this same resolved sample rejects the log-normal assumption ($p{<}0.0001$), but the test itself is a weak basis for this decision either way: it compares a right-truncated, grid-tied sample (crossover values can only take the finite set of values the crossing function's interpolation produces, and repeat identical values across resamples are common, not a large i.i.d. continuous draw) against an untruncated continuous log-normal CDF, a mismatch in test assumptions that inflates the rejection regardless of how well the two distributions actually agree in their bulk. Given both corrections, we still do not adopt 847.9 as this paper's reported bound, but for a narrower reason than originally stated: a good match at the 90th percentile does not validate an extrapolation to the 97.5th, a more extreme, less-constrained region of the tail where a distributional assumption this data cannot itself confirm still carries most of the weight. The attempt is reported for transparency, not as a replacement for $[234.0, >650]$: it demonstrates that a censoring-aware method is implementable and gives a rough sense of scale (the true upper bound is plausibly some hundreds above 650, not unbounded), but the specific number (847.9) rests on a distributional assumption this data does not clearly support, so the non-parametric > 650 disclosure remains the more defensible one for this paper's main claims.

B.25 Near-duplicate distractors at high $|V|$, quantified

§3 discloses that near-duplicate distractors — functions lexically similar to the gold answer, as opposed to exact-name collisions, which are excluded by construction — are possible and unscreened at high $|V|$. This exposure is worth quantifying rather than left as a qualitative caveat. This section does so for 14B at $|V|{=}650$, the paper's largest catalog, using the same zero-parameter lexical scorer as `heuristic_valid` (name+description token overlap) as a proxy for near-duplication, since this project has no embedding-similarity infrastructure available; this is disclosed as a proxy, not equated with true semantic similarity.

Method. For each of the 200 tested cells, the candidate list is reconstructed directly from its own seed (same method as App. B.17), and every non-gold distractor's Jaccard token overlap with the gold answer's own name-plus-description is computed. A cell is "near-duplicate-heavy" if at least one distractor exceeds a given overlap threshold.

Exposure and outcome, across three thresholds. At a strict threshold (Jaccard ≥ 0.5, close textual overlap): 10.0% of cells are near-duplicate-heavy, success 0.250 vs. 0.261 on near-duplicate-light cells — a 1.1pp gap, not a meaningful difference at this sample size. At a looser threshold (≥ 0.4): 20.5% heavy, 0.268 vs. 0.258. At the loosest threshold tested (≥ 0.3, likely capturing many genuinely distinct functions that merely share common domain vocabulary): 59.0% heavy, 0.237 vs. 0.293 — the largest gap of the three, in the direction expected if near-duplicates hurt performance, but at a threshold loose enough that "near-duplicate" is doing less work as a label. No threshold shows a gap large enough, relative to this sample size, to suggest near-duplicate distractors are a primary driver of the $|V|{=}650$ result specifically.

Limitation. This is a lexical proxy, not a semantic one: two functions could be near-duplicates in effect (near-identical capability, different wording) that this method would not flag, or share substantial vocabulary while serving distinct purposes and be flagged as false positives. The qualitative disclosure in §3 stands; this section adds the quantitative check that disclosure could not provide on its own, without claiming it fully resolves the question a true embedding-based measure would answer more precisely.

B.26 FRONTIER CLOSED-MODEL CHECK (GPT-4O)

The checks above (App. B.2, B.22) stay within one model family (Qwen2.5) or, for the 72B check, a different scale of the same architecture lineage. Neither speaks to whether the crossover generalizes to a frontier, closed-weight model from a different vendor entirely — the models tool-use agents are most often deployed with in practice. Like the 32B check, this was not pre-registered: it directly tests whether the crossover holds for production-grade closed models, not only small-to-mid open ones.

GPT-4o via OpenRouter (`openai/gpt-4o`), same harness, prompts, and `MASTER_SEED` as every other BFCL leg. Deliberately small-n (*"even a single, smaller-n run... at a couple of levels"* was the scope judged sufficient for a calibration check of this kind): 10 cases $\times$ 5 seeds, at the two endpoints of the extended grid ($|V| \in \{234, 650\}$) rather than the full seven-level sweep, to keep this calibration check affordable. Not precision-matched (API quantization, system prompt, and serving stack are unobserved, same disclosed limitation as the 72B check).

No crossover was found within this range. `constrained_llm` (GPT-4o) beats `heuristic_valid` by a wide, bootstrapped margin at both endpoints: $+0.460$ at $|V|$=234 (95% CI $[0.200, 0.720]$) and $+0.360$ at $|V|$=650 (95% CI $[0.140, 0.580]$), $n_{\text{boot}} = 5000$ paired over cases. This is right-censored, not a measured $|V|^*$: GPT-4o's crossover, if it exists at an affordable $|V|$ at all, lies beyond 650 — qualitatively different from every open-weight model tested (0.5B–32B), all of which showed a real crossover well inside this range. We read this as evidence the crossover's practical location depends heavily on model class, not only parameter count: a frontier closed model resists the heuristic far longer than an open-weight model at any scale tested here, addressing the underlying concern that results on small-to-mid open models may understate robustness at deployment scale.

One data-quality note: `random_valid` and `random_unconstrained` both landed at exactly 0 successes across 100 cells each (both levels combined) — expected at this n given their tiny theoretical success probability ($1/|V| \approx 0.004$ and 0.0015 respectively), not a harness fault; consequently `structure_share` is uninformative here (denominator floor, same class of artifact as §4.4's H6 sign-flip) and is not reported as a claim.

B.27 SECOND FRONTIER CLOSED-MODEL CHECK (CLAUDE SONNET 5)

One frontier model is a single data point, not a pattern: is App. B.26's result GPT-4o-specific rather than general to frontier closed models? A second model, from a different vendor, tests exactly this. Same harness, same reduced small-n scope, same two grid endpoints ($|V| \in \{234, 650\}$) as the GPT-4o check, so the two results are directly comparable.

The pattern replicates. `constrained_llm` (Claude Sonnet 5) beats `heuristic_valid` by an even wider margin than GPT-4o did at both endpoints: $+0.560$ at $|V|$=234 (95% CI $[0.300, 0.800]$) and $+0.480$ at $|V|$=650 (95% CI $[0.260, 0.720]$), $n_{\text{boot}} = 5000$. Right-censored, same as GPT-4o: no crossover observed within this range, from a second lab's model, on the same protocol.

One methodology caveat, disclosed rather than absorbed silently. Claude Sonnet 5 does not accept `temperature`; requests specifying it succeed but the parameter is silently ignored, and reasoning ("thinking") cannot be disabled on this model family — confirmed directly against OpenRouter's own Claude 5 migration documentation, not assumed. This result is therefore not run at the same deterministic setting (`temperature=0`) as every other leg in this paper; it reflects Claude Sonnet 5's server-side default sampling and reasoning behavior. We report it because the effect size is large enough that this caveat is unlikely to change the qualitative conclusion (no crossover in range), but flag it explicitly rather than let the shared `APILanguageModel` harness's `temperature` parameter imply a determinism this specific model does not actually provide.

B.28 FULL QUANTIZATION COMPARISON (FP16 VS. 4-BIT)

Every $|V|^*$ value in the main text is measured under 4-bit quantization. We also have fp16 runs at two scales, same convention (40 cases $\times$ 5 seeds, all seven $|V|$ levels). At 1.5B: $|V|^*(1.5\text{B}, \text{fp16}) = 16.8$ vs. $|V|^*(1.5\text{B}, \text{4-bit}) = 8.7$ — roughly a $2\times$ shift, non-overlapping CIs. At 3B: $|V|^*(3\text{B}, \text{fp16}) = 32.0$, 95% CI $[21.7, 52.9]$, vs. $|V|^*(3\text{B}, \text{4-bit}) = 28.7$, CI $[19.5, 43.4]$ — heavily overlapping, no detectable shift. The 1.5B direction remains informative where it holds (4-bit quantization can make the LLM look *weaker* relative to the heuristic than at full precision), but the effect is not uniform across scale: whatever drives the 1.5B shift does not obviously generalize to 3B. A third check at 7B is right-censored within native–128 — a T4's 16GB cannot hold fp16 7B weights ($\sim$14GB alone) plus the long-context KV-cache $|V|$=234 requires — but the crossover is essentially *at* $|V|$=128 (`constrained_llm`= 0.415 vs. `heuristic_valid`= 0.405), closely matching the 4-bit value of 109.2: unlike 1.5B, precision does not appear to move the crossover substantially at 7B. Three scales is still not a full fp16 curve, and the one available fp16 point beyond 3B is right-censored, not resolved.

What the available fp16 points do and do not say about α. A rough three-point fp16 fit ($\alpha\approx1.3$) can be obtained by treating 7B's right-censored value as a point estimate (≈ 128). Reproduced exactly: doing the same gives α=1.330. This is not proposed as a real estimate — a right-censored value means the crossover is *at least* 128, not that it equals 128, and fitting a regression through it as though it were resolved repeats the same class of error this paper corrects elsewhere (App. B.17) when a biased or incomplete number is treated as clean. With only the two genuinely resolved fp16 points (1.5B, 3B), the honest pairwise slope is 0.930 — different again from both the rough 1.3 and the 4-bit α=1.784, but this is a single slope between two points, not a fit, and carries no meaningful CI at that count (the same $\geq$3-point bar applied to every α reported elsewhere in this paper, including Seal-Tools', App. B.4). We do not report a headline fp16 α: neither number above meets this paper's own standard for one, and manufacturing a third clean point would need a 7B fp16 run at $|V|\geq$234 that the hardware available for this project could not support (a 14B fp16 point would face the same constraint more acutely). What the two clean points and the reviewer's own rough check agree on is only this: precision plausibly affects α's magnitude, and the existing 1.5B/3B evidence does not by itself pin down by how much — reported to that extent and no further.

B.29 PHI-3.5 $|V|$=128 COLLAPSE: LONGROPE TEST

§4.3 reports that Phi-3.5's success rate collapses at $|V| = 128$ (to 0.060, vs. a neighboring 0.405 at $|V| = 64$), not a parsing failure (`parse_ok`=100%); the model clusters onto three index values in 54% of generations, a collapse signature. We pre-registered a specific test: whether the $|V|$=128 prompt length crosses Phi-3.5's documented LongRoPE switch point (`original_max_position_embeddings`=4096, tolerance $\pm$300). Measured: 3,745–3,756 tokens, 340–351 short, consistently outside tolerance — a near-miss, not a confirmation. This rules out a clean mid-sweep regime switch but leaves open a weaker version (degradation approaching a context ceiling before length-extension engages, documented elsewhere in long-context architectures) that we do not have direct evidence for. The collapse is real and content-level; its full cause is not identified. We exclude $|V| = 128$ from the crossover reported in §4.3 (which resolves at lower $|V|$, unaffected) and do not extend the decay-rate comparison to this model's full curve.

B.30 CONFOUND CHECKS: FULL REASONING

§4.4 reports three confound checks in condensed form; the full reasoning for each:

`unconstrained_llm` (no candidates shown) never lands in-vocabulary: 0 successes across 5,600 cells spanning all four Qwen-family runs — the expected floor given the model must name one exact function, by exact string match, from a $\sim$1,222-function vocabulary it was never shown, under a strict parsing criterion, not evidence of a parsing artifact; this is why the arm carries no signal and is excluded from the pairwise-comparison family. Non-LLM arms are bit-identical across every run that shares them (0 mismatches), confirming a shared random stream and ruling out silent harness drift. Parse-failure rate peaks at 0.11 (1.5B, $|V| = 8$); under the conservative convention treating parse failures as missing rather than as task failures, 86.1% of the 4-bit success drop is attributable to selection, not formatting.

Mistral-7B's OOM at $|V|$=234 is not explained by either model being simply "bigger." Published configs put Qwen2.5-14B *above* Mistral-7B-Instruct-v0.3 on every standard architecture metric we checked, not below: 48 layers vs. 32, 14B parameters vs. 7B, and a larger per-token KV-cache (8 KV heads $\times$ 128 head-dim $\times$ 48 layers $\times$ 2 = 98,304 elements/token, vs. Mistral's 65,536 — roughly $1.5\times$ larger). Naive size-based intuition therefore predicts Qwen-14B should be the one under memory pressure, not Mistral-7B; it is not. One architectural difference that points the right direction is vocabulary size: Mistral's tokenizer has 32,768 tokens against Qwen's 152,064 ($4.6\times$ larger), so the same 234-candidate prompt plausibly tokenizes to more tokens under Mistral, inflating its effective sequence length at what we call the same nominal $|V|$. **This alone is unlikely to be sufficient:** at Mistral's own published KV-cache size, even a generously long $\sim$7k-token prompt implies a KV-cache on the order of 1GB, and 4-bit weights add roughly 3–4GB more — well under the 16GB budget on its own. The tokenizer-length difference is a real, verified fact about the two models, but by itself it does not obviously add up to an OOM under standard assumptions; attention implementation, batch/padding behavior, or other framework-level memory overhead not captured by this back-of-envelope accounting are at least as plausible a cause, and we did not instrument actual per-model token counts or peak memory on the $|V|$=234 prompts to distinguish between these. We report the vocabulary-size observation as one verified, contributing fact, not as an established or sufficient explanation for the OOM itself.

B.31 Cost-proxy reasoning (§4.5)

§4.5 reports that filtering reduces the candidate count shown to the LLM at $|V|$=234 (7B) to a mean of 110 (53% fewer), as a cost proxy rather than a measured latency benchmark. The full reasoning: we did not instrument wall-clock timing, so we report the actual candidate count shown after filtering instead. Since prompt length scales roughly linearly with candidate count at fixed per-candidate description length (§3), this proxies prompt-side *token* savings; since self-attention's prefill cost scales quadratically with sequence length, the implied compute savings are larger still than the linear token reduction alone suggests, without substituting for a measured latency/cost table.

C Extended Related Work: Controlled-Attribution Studies

§2 differentiates this paper's two closest concurrent precedents in the main text (Cho (2026) and Chen et al. (2026)). The remaining seven concurrent controlled-attribution papers are differentiated here in full: Dastidar (2026) attributes an enterprise agent's uplift to scaffolding and routing rather than verification, without a random-selection control; Yao et al. (2026) benchmark harness-configuration variance with no decomposition or random control; Wang (2026) show one variable (embedding model) flips agent-memory conclusions, the same controlled-attribution philosophy directed at memory rather than action selection; Dhakal et al. (2026) show model-matched plain-agent baselines can match published scores, but their baseline remains an LLM agent throughout — no zero-learning arm, which our heuristic supplies; Gideoni et al. (2026) report simple baselines matching code-evolution pipelines, the same "check the trivial control first" family applied to code; Repantis et al. (2026a) is a companion analysis to Repantis et al. (2026b) (Table 1) asking how many tools an agent should be shown at all, a chance-corrected selection-count recommendation rather than a measured, executed sweep across $|V|$ and scale; Jung & Son (2026) measure an agent harness's own residual contribution once the underlying LLM is held fixed, the harness-side complement to this paper's model-scale sweep, without varying $|V|$ or model capability themselves. None of the nine reaches this paper's sequential, executed, valid-set manipulation across model scale.

For the two closest precedents (§2), additional detail not given in the main text: Cho (2026)'s non-monotone harness sensitivity finding has `format_violation` dominant as the driver. Chen et al. (2026) find format effects exceed content effects (Wilcoxon p=1.7$\times$10^{-3}) in their content/format-recovery decomposition; no prior work performs the analogous split for agents, which our `unconstrained_llm` floor supplies.

D ALFWORLD: ATTEMPTED MULTI-STEP EXTENSION AND POWER ANALYSIS

We attempted to extend this paper's measurement to sequential decision-making using ALFWorld, across three separate engine-configuration attempts. None produced a structure effect distinguishable from zero. This appendix reports why, quantitatively, rather than presenting an underpowered null as a finding.

The design was underpowered by construction, not by accident. The measured mean valid-set size on ALFWorld is $m = 27.4$; a uniform-random policy's per-step probability of a genuine progress action is therefore $p = 1/m = 0.0365$. The easiest task category observed across both architectures tested (Qwen-7B, Llama-8B) required as few as $d = 4$ progress steps to solve. Under the initial budget $T = 40$:

$$P(\texttt{random_valid} \text{ succeeds} \mid d{=}4, T{=}40) = 0.057, \qquad \text{expected successes / 5 seeds} = 0.29.$$

Observing zero successes for the random control was therefore the modal outcome under this design, not evidence that structure is absent.

Table 16: Required step budget T for `random_valid` to have reasonable power, by task difficulty d (progress steps required).

T	P(succeed | d=4)	P(succeed | d=6)	P(succeed | d=9)
40	0.057	0.003	0.000
100	0.498	0.160	0.011
150	0.801	0.469	0.100
200	0.936	0.742	0.310
300	0.996	0.964	0.769
400	1.000	0.997	0.957

The T=300 retry (E2aFIX) still did not land, for a different, compounding reason. We re-ran at $T = 300$, inside the range the table above says should give reasonable power for easy-to-moderate tasks. Structure remained floored: 0.008, 95% CI $[0.000, 0.024]$ — an order of magnitude below what would be needed to support H2. Diagnosing why, we found a task-identity bug we had not previously checked for: the numeric `task_id` used to align episodes across separate runs is a positional index, not a stable game identifier. Comparing the actual underlying ALFWorld game path (`env_task_id`) across runs that share the same numeric `task_id` range, we measured only **4 of 25** games in common between the original T=40 run and the T=300 retry, and **5 of 25** between two different T=300-family attempts. A budget increase intended to give the same 25 tasks more steps to succeed instead partly re-sampled a different set of games each time, which independently degrades any structure estimate regardless of budget. This is a methods lesson we record explicitly: **a numeric task identifier is not a safe proxy for task identity across separately-launched runs unless the task list is frozen and passed explicitly**, and we did not do this until after all three ALFWorld attempts had already run.

The competence ("learning") term is also not distinguishable from zero. Bootstrapping `constrained_llm` − `random_valid` over task resampling (n = 20,000): Qwen-7B gives +0.039, 95% CI $[0.000, 0.120]$, $P(\text{learning} \leq 0) = 0.362$; Llama-8B gives +0.040, CI $[0.000, 0.120]$, $P(\leq 0) = 0.363$. Both models solve successes coming from **exactly one** distinct task category (5/5 seeds), the easiest one in the design. We report this as "solves the easiest category reliably, fails the rest," not as an aggregate success percentage, since a single-task-category effect and a broad competence effect would look identical in a summary statistic but mean very different things.

Consequence for this paper. We do not extend any of this paper's quantitative claims — the crossover, α, the dilution law — to the multi-step setting. The ALFWorld attempt is reported here as what it is: a properly diagnosed negative result with an identified, partly compounding methods bug, offered as a concrete lesson (freeze and explicitly pass task identity across runs; power-check before budgeting a multi-step control) rather than as a finding about structure in sequential decision-making one way or the other.