Does GLM really perform badly at calling artifact-writing tools?

Short answer: no. Across two months of arena boards, GLM-5.2 and GLM-5.3 look like the worst models in the field at producing a report artifact — SYN 0 on the two workflows whose deliverable is a long document. Neither failure is a tool-calling deficit. Both are harness defects, and they stack.

What SYN actually measures

The Model Ability Card's SYN stat is fed by exactly two assertion types, `artifact_exists` and `artifact_contains`. Nothing else maps to the synthesis axis. So "low SYN" is never a soft judgement about writing quality — it means either *no artifact was produced* or *the artifact's body lacked a required string*.

Across 349 distinct scored trials the two modes split unevenly: 41 trials produced no artifact at all, against 262 individual content-check failures. The first mode is the one that looks like a capability story, and it is the one GLM dominates.

The first look, which is wrong

Ranked by SYN pass rate over every scored trial, GLM sits at the bottom:

model	SYN	pass	no artifact	thin body
glm-5-3	0.0%	0	2	7
glm-5-2	44.4%	24	6	24
…				
grok-4-6, gemini-3-7-flash, gpt-5-6-sol	100%	—	0	0

Read naively, that is a capability finding. It is not, and the reason it is not is that **a budget artifact must be excluded before any capability claim** — the desk learned this the hard way on 2026-08-19.

Defect 1 — a 4096-token ceiling nobody set

`langchain_anthropic.ChatAnthropic` fills in `max_tokens` when the caller leaves it unset, by looking up a per-model profile keyed on the exact model id:

```
if values.get("max_tokens") is None:
    profile = _get_default_model_profile(model) if model else {}
    values["max_tokens"] = profile.get("max_output_tokens", _FALLBACK_MAX_OUTPUT_TOKENS)  # 4096
```

Every id routed through this desk's Anthropic path carries a ZenMux vendor prefix, and the prefix defeats the lookup:

model id	resolved ceiling
`z-ai/glm-5.3`, `z-ai/glm-5.2`	none → **4096**
`minimax/minimax-m3`, `meituan/longcat-2.0`	none → **4096**
`anthropic/claude-sonnet-5`	none → **4096**
`claude-sonnet-4-5-20250929` (unprefixed)	64000

So every model on that path ran at 4096 output tokens, including the Claude ids whose real ceiling is 64000 — a 15.6× under-allocation.

It is invisible by construction

A truncated turn is not an error. It returns:

```
content:           [{"type":"text","text":""}, {"type":"reasoning", …}]
response_metadata: {"model_name":"z-ai/glm-5.3", "stop_reason":"max_tokens"}
tool_calls:        []
status:            success
```

Empty text, no tool call, and a **successful** span. The arena's infra-blank gate corroborates a blank transcript with step *errors* before excluding a match — and there are none. So the harness scores a dead turn as a model failure. `stop_reason` is the only tell, and nothing read it.

It is exclusive to that path

Counting `stop_reason == "max_tokens"` across every arena LLM call:

model	calls	truncated	rate
glm-5-3	796	32	**4.0%**
claude-sonnet-4-6	1,028	25	2.4%
longcat-2-0	1,450	14	1.0%
claude-sonnet-5	1,746	15	0.9%
minimax-m3	1,799	12	0.7%
qwen-3-7-max / glm-5-2	1,744 / 2,114	5 / 6	0.3%
all 25 other models	~30,000	**0**	**0.0%**

Seven models truncate; they are exactly the seven that have been routed over the Anthropic protocol. GLM-5.3 has the highest rate in the field.

Why the artifact call dies first

A cap does not degrade an agent uniformly. It ablates whichever call needs the most output, and in a golden workflow that is unambiguously the report call — its argument *is* the deliverable. Measured over 354 successful `write_report_artifact` calls:

	tokens
median body	~1,595
p75	~2,526
p95	~3,851

The body **alone** exceeds 3,000 tokens in 16% of calls, leaving under 1,096 for the reasoning block a reasoning model emits first. Every other tool call in the workflow takes a handful of ids and numbers. So a global ceiling reads, on the scoreboard, as a specific inability to write reports.

The prediction that follows is size-dependence, and it holds. On the *same capped configuration*, GLM-5.2 passes SYN 3/3 on `risk-limit-breach-day`, whose report is smaller, while scoring 0/5 on `high-board-portfolio-review-day`.

The natural experiment: Run #114 → #115

GLM-5.3 ran all five workflows on 2026-08-19 before the fix, and again after it. Same model, same workflows, one trial each; the budget is the only change. `completion_tokens` maxed at exactly 4096 on all five #114 threads and reached 12,314 in #115.

workflow	#114 (4096)	#115 (uncapped)
risk-manager-control-day	OVR 67, SYN 0, obj 82.1	OVR 87, SYN 99, obj 100.0
high-board-portfolio-review-day	OVR 66, SYN 0, obj 68.6	OVR 69, SYN 79, obj 77.1
trader-rfq-booking-day	OVR 62, SYN 99, obj 65.1	OVR 91, SYN 99, obj 96.8
risk-limit-breach-day	OVR 87, SYN 99, obj 100.0	OVR 91, SYN 99, obj 100.0
ops-settlement-day	OVR 79, SYN 99, obj 90.9	OVR 81, SYN 99, obj 90.9
mean	**OVR 72.2 / obj 81.3**	**OVR 83.8 / obj 93.0**

+11.6 OVR and +11.6 objective points from one configuration line. In #114 the model made *zero* artifact-tool calls on the two failing workflows; in #115 it called the tool on all five.

Defect 2 — GLM-5.2's other cause

GLM-5.2's history splits into three regimes, and only the middle one looks like incompetence:

era	runs	condition	SYN
uncapped, OpenAI protocol	20, 33	healthy	**4/4, 4/4, 5/5, 5/5**
OpenAI protocol	51, 91	`Tool call ID is required for subagent invocation` ×9—15 per thread	**0/6, 0/6, 0/5, 0/5** (obj 12.0)
Anthropic protocol	93, 100	4096 cap	0/5 on the big report, **3/3** on the small one

In runs 51 and 91 the OpenAI-compatible gateway returned tool calls with an **empty string** as the id. deepagents' `task()` guards on it, so every persona delegation raised before a subagent started — no persona ever ran, and the objective score collapsed to 12.0 while the model reasoned perfectly well about its own failure. Pinning GLM-5.2 to the Anthropic protocol fixed that defect and delivered it straight into the 4096 ceiling.

On the flagship, uncapped and on a working protocol, GLM-5.2 scored SYN 4/4 twice, at objective 89.7 and 92.3. That is the control the whole question turns on.

Controlled A/B

Both GLM ids, both failing workflows, two trials, two budgets — the budget being the only difference. Arm A is Run #118 at 4096; arm B is Run #119 at 32768.

Arm A — budget 4096, complete (8/8 trials):

model	workflow	trial	artifact?	SYN	obj	truncated turns
glm-5-2	high-board	0 / 1	**no / no**	0/5, 0/5	60.0, 71.4	3, 1
glm-5-2	risk-manager-control-day	0 / 1	**no / no**	0/4, 0/4	69.2, 64.1	6, 4
glm-5-3	high-board	0 / 1	**no / no**	0/5, 0/5	65.7, 65.7	6, 4
glm-5-3	risk-manager-control-day	0 / 1	**no / no**	0/4, 0/4	82.1, 82.1	12, 7
			0/8 = 0%	**0/36 = 0%**	mean 70.0	**43 total**

Every thread clipped at exactly 4096. Not one of the sixteen artifact checks passed, across two models, two workflows and two trials.

Arm B — budget 32768, complete (8/8 trials):

model	workflow	trial	artifact?	SYN	obj	truncated turns
glm-5-2	high-board	0 / 1	**YES / YES**	4/5, 5/5	77.1, 80.0	0, 0
glm-5-2	risk-manager-control-day	0 / 1	**YES / YES**	4/4, 4/4	94.9, 89.7	0, 0
glm-5-3	high-board	0 / 1	**YES** / no	5/5, 0/5	88.6, 68.6	0, 0
glm-5-3	risk-manager-control-day	0 / 1	**YES / YES**	4/4, 4/4	94.9, 97.4	0, 0
			7/8 = 88%	**30/36 = 83%**	mean **86.4**	**0 total**

Arm B was cut short mid-run by a ZenMux `402 quota_exceeded`; three matches were swept to `invalid` and re-run with `--resume` at the same budget once the rolling window reopened. Not one turn in the completed arm truncated, and the peaks — 5,391 / 7,112 / 7,157 / 7,420 / 8,058 / 10,852 / 12,616 / 15,883 / **24,163** — sit far above the old ceiling. GLM-5.2 emitted nearly **six times** 4096 in a single turn once it was allowed to.

Against the predeclared criteria

criterion	threshold	result
arm B produces the artifact	⩾ 75%	**88%** (7/8)
arm A produces the artifact	⩽ 25%	**0%** (0/8)
arm A truncates, arm B does not	—	**43 vs 0**

H1 is supported on all three. Mean objective moves 70.0 → 86.4 on the budget alone, across both models and both workflows.

The one uncapped failure is real, and it is not GLM's

One arm-B trial still produced no artifact: glm-5-3 on high-board, trial 1. It carries **zero** truncated turns, so it is a genuine miss rather than a budget artifact — and it is worth reporting precisely because

it is the honest half of the answer. The model routed `generate-report`, gathered evidence over fourteen tool calls, and then wrote the whole report into the chat reply, announcing:

> "…drafted as Markdown by the high_board persona — **read-only: no report job, artifact, approval, or release was created**."

That is a deliberate conservative reading of its remit, not an inability to call the tool. It is also **not a GLM trait**: DeepSeek V4 Pro did exactly the same thing in Run #104, down to the parenthetical that the report was not persisted. Prose-instead-of-persist is a real failure mode on this benchmark, it costs the whole synthesis axis, and it is orthogonal to everything else in this study.

The cap also cost efficiency

Truncated turns are re-tried, so the cap made the agent slower as well as worse. On the flagship, capped GLM-5.3 spent 111 and 93 LLM calls to score 82.1; uncapped in Run #115 it spent 72 to score 100.0. A budget too small to finish the work is not a saving — it buys back nothing and costs EFF on the card.

What it cost the published boards

Every board on the public leaderboard contains at least one contestant whose turns were silently truncated:

board	workflow	damaged contestants
Run #20	risk-manager-control-day	qwen-3-7-max (5 truncated + 12 tool-id errors), minimax-m3 (12), longcat-2-0 (5), claude-sonnet-5 (3)
Run #33	trader-rfq-booking-day	longcat-2-0 (5), claude-sonnet-5 (1)
Run #94	high-board-portfolio-review-day	glm-5-2 (4), longcat-2-0 (2)
Run #101	risk-limit-breach-day	glm-5-2 (2), longcat-2-0 (2)

These boards understate those models. They are not being silently rewritten — a board is a measurement taken under stated conditions — but the condition is now stated.

The honest residual

Lifting the cap does not make GLM perfect. Uncapped on `high-board`, GLM-5.3 writes the artifact, names the governance framing, the Snowball composition and the governed valuation 238 — and still

omits the second required figure, 17.5. That is a genuine content gap worth one check, not a tool-calling defect: one point rather than five. Together with the prose-instead-of-persist trial above, that is the real, unglamorous size of GLM's remaining weakness on this axis.

Mitigation

1. **Never let a library choose a budget.** Set `max_tokens` explicitly on every protocol path. A default keyed on a model-id profile lookup fails silently for any id the library has never heard of — which is every id behind a gateway.
2. **Make truncation loud.** `stop_reason == "max_tokens"` is explicit in `response_metadata` and nothing in the harness read it. Blank-detection that corroborates with *errors* structurally cannot see a truncated turn, because truncation is a success.
3. **Score a truncated match, but flag it visibly.** The desk's call, and the opposite of what the existing transport-error gate does: an `invalid` match is excluded from board means, which would silently shrink Run #20 by four contestants and leave a reader wondering where they went. A visible flag keeps the field whole and puts the caveat where the number is. The cost is that a flagged number still sits on a public page, so the flag has to reach the card and the board row, not just the drilldown.
4. **Make the output budget a run variant, like reasoning effort.** Budget is currently a process-level env var with no column on `arena_run`, which is why this study had to run it as two separate runs that must never be merged. Promoting it mirrors what migration 0057/0058 did for effort — and it must join the **contestant key**, because `merge_runs` folds by `(workflow_id, model_id, reasoning_effort)` and a budget outside that key would average two operating regimes into one row. Two budgets are two regimes: this study measures an 11.6-point OVR gap between them.
5. **Size the budget to the per-model appetite, and lift it for everyone.** Appetite varies widely across the field; special-casing one model would simply relocate the bias.
6. **Diagnostic rule of thumb:** when a model appears unable to call one specific tool, check the output budget before capability. The tool with the biggest payload dies first, and on this desk that is always the one that writes the deliverable.

Method

Arena database, 349 distinct scored trials over runs #10–#116 for the field-wide figures, merged runs excluded (they re-contain their sources). The controlled A/B is Run **#118** (budget 4096) and Run **#119** (budget 32768), 2 models × 2 workflows × 2 trials each, run strictly sequentially and never merged — budget is not in the contestant key, so a merge would average the two regimes. Run #119 lost three matches mid-run to a ZenMux `402` and was completed with `--resume` at the same budget.

Truncation is counted from the trace database as LLM spans whose `response_metadata.stop_reason` is `max_tokens` — an explicit provider signal, not a token-count

threshold, because a capped turn and a turn that merely happens to be long are otherwise indistinguishable. Per-trial breakdowns are walked through `aggregate[]` rather than the folded top-level `objective`, which carries trial 0 only, and per-trial transcripts (`transcript.trial<N>.json`) rather than `transcript.json`, which is the last clean trial. Appetite percentiles exclude truncated observations, which are censored at the cap by construction.

The design — arms, scorer and all three pass criteria — was predeclared before any run in the A/B executed. No agent code was modified for this study; both arms use the existing `OPEN_OTC_AGENT_MAX_OUTPUT_TOKENS` seam.
