The Myth of Ever-Faster AI

Eight years of models, GPT-1 to 2026, tested on real production work

By David Hood and Youssef Khemiri, 42 Robots AI

36 models · 25 task types · 79 measures · 2018 to 2026

October 2026 · Version 1.0

Licensed CC BY 4.0

© 2026 David Hood and Youssef Khemiri / 42 Robots AI

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

https://creativecommons.org/licenses/by/4.0/

You may share and adapt with attribution.

Contents

Real progress over time on practical work 3
What the curve measures 5
Intent classification shows the transition 7
The original ChatGPT era remains competitive on many of these tasks 9
One model at a time gives a similar answer 9
The early models usually could not produce a usable answer 10
The flattening survives three different views 10
Fewer tasks improve each year 10
Many best scores barely move after 2023 11
The average new model adds little on this task set 12
Every task then and now 12
Where progress happened it was large 13
Easy work can still be valuable work 14
What the wrong buying signals cost you 15
You can choose a model that performs worse 15
Older models remain serious candidates 16
You can pay 326 times more for the same test score 17
Match the model to the work 18
Four common situations 18
How to test structured work at volume 19
Make the model prove its value 19
Methods and limits 20
About the authors 20
About 42 Robots AI 20
How to cite 20
Contact 20
Appendix 21
Scope and study design 21
Run conditions and scoring 21
What the model judges measured 21
Score scales and progress calculations 22
Statistical comparisons 22
Business conversions and limits 23
Models and release-year assignments 23
The same eight evaluations across years 24
Additional metric definitions 25

AI is getting better faster and faster. Every new generation is a bigger leap. If you want the best results, use the newest top tier model.

That is the story companies have been hearing and continue to hear. Instead of trusting the narrative, we measured performance on practical work from real world production AI system use. The data tells a different story.

We chose models for real features for our own software and customers. Which model could turn a request into the right action? Which could extract a table correctly? Which could write useful content? A strong public benchmark score did not answer those questions. We built evaluations of the work itself.

Those evaluations began in late 2025. Only later did we ask what they revealed about progress over time. We compared 36 models from 2022 to 2026, plus GPT-1 and GPT-2 as early anchors. The progress analysis covers 25 task types and 79 measures. On 16 of those 25 task types, a model released before 2026 matched or beat every 2026 model.

Key findings

- About 87% of the measured gain on our practical task set happened by 2022.
- On 16 of 25 task types, a model released before 2026 matches or beats the best 2026 model. GPT-3.5, from 2022, ties it on four.
- On 14 of 25 task types, the best available score rose by less than two points after 2023. Generation is the major exception.
- Flash-Lite and Qwen outperform Opus 4.8 on entity normalization, with the wins surviving corrected paired tests.
- Thirty-four models score 1.00 on 15 SQL cases, with a 326-fold cost spread: about $29 to $9,455 per million comparable calls.
- Choose models by task effectiveness first.

Real progress over time on practical work

The largest rise on our practical task set happened between 2019 and 2022. By the original ChatGPT era, the best available score was about 0.87. By 2026, it was about 0.99. We take the best score available for each task from any model released by that year, then average across tasks. We call this combined score the composite; no single model achieves it.

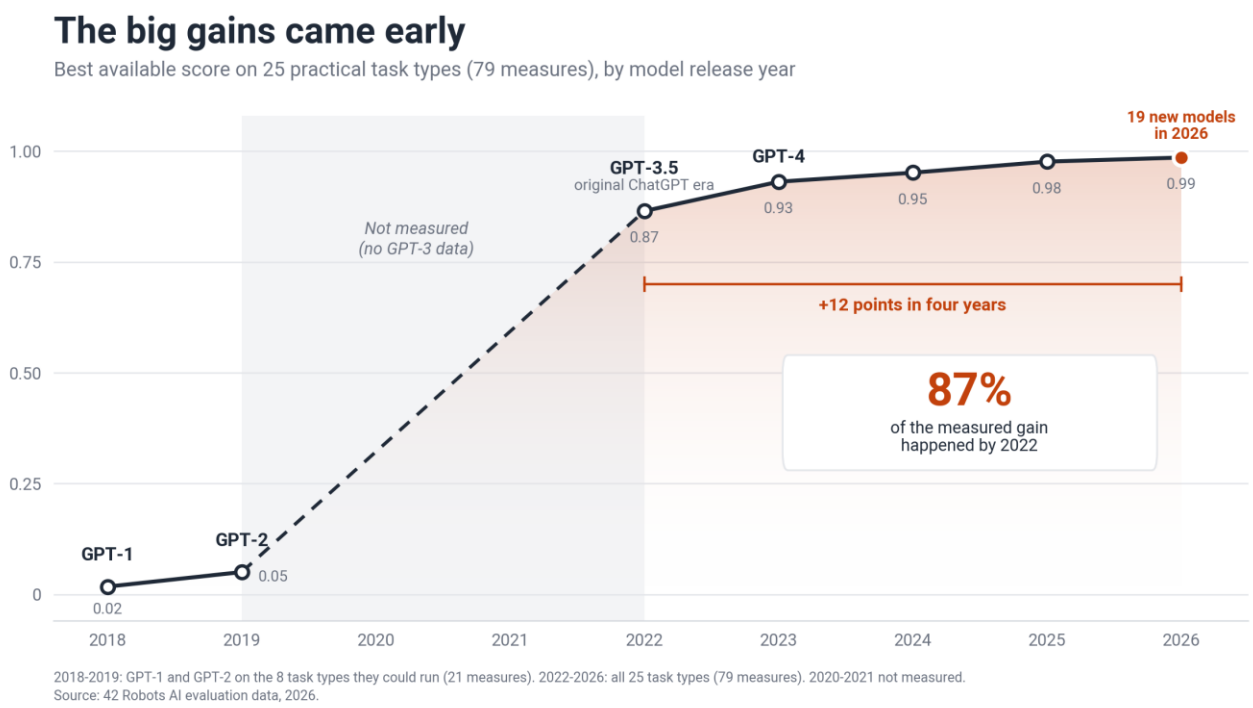

Figure 1. Best available score on practical production tasks by model release year.

The chart's "25 Evals/79 Measures*" refers to 25 task types and 79 measures. The combined score averages the best score per task. *The 2018 and 2019 anchors use eight evaluations; modern models use 25 task types. Comparing all measured years on the same eight evaluations gives the same broad shape. *Source: 42 Robots AI evaluation data, 2026.*

Table 1. The best available score by release year.

Year	Best available score	Gain in points
2018	0.0175	Baseline
2019	0.0504	+3.29
2020–2021	Not measured	Not measured
2022	0.8650	+81.46
2023	0.9311	+6.61
2024	0.9515	+2.04
2025	0.9765	+2.50
2026	0.9857	+0.92

Source: 42 Robots AI evaluation data, 2026. A point is 0.01. Annual gains use unrounded scores. No measured GPT-3 value is available.

About 87% of the measured gain happened by 2022. The matched comparison supports the same conclusion.

> *The shape everyone thinks we're still climbing, we already climbed for many real-world use cases. On these tasks, most of the gain happened by 2022.*

What the curve measures

The 25 task types come from three kinds of practical work. Fifteen come from a production AI system we built and ran with real users that do knowledge management, website building, and an agent that acted on a user's workspace. Six are from data-processing work we do for customers in custom AI-powered software that read schemas and tables, normalize names, match records, write SQL, and refactor code. The remaining 4 are reliability tests that matter in any production system for knowing when not to answer, standing firm against false claims, choosing the right tool, and recalling a fact from earlier in a conversation.

The software needed the model to get the tasks done effectively. A user request had to reach the right part of the system. A messy company name had to become its standard form. Two customer records had to be matched or kept apart. A model-generated query had to retrieve the right data. A summary had to preserve useful information. A model had to decline a question it could not answer. Calling these tasks practical is shorthand for those specific obligations.

Table 2. The 25 task types we tested.

Shape	Task	Origin	Input and correct output	Cases	Scoring
Classification	Intent classification	Our product	Read one user message and return exactly one of nine intent labels.	100	Exact match
Classification	Natural-language task classification	Our product	Turn a task or reminder request and a fixed today date into JSON containing the task phrase and an ISO due date.	86	Rule-based check
Classification	Relevance checking	Our product	Given a query and stored content, decide RELEVANT or NOT_RELEVANT. Five prompt styles are tested.	111 x 5 styles	Exact match
Classification	Workspace routing	Our product	Match a user input to a named canned response flow using its sample triggers, or return none for a near miss.	105	Exact match
Extraction	Action extraction	Our product	Turn a workspace request into JSON listing create, move, rename, or delete actions, the task type, and whether the user asked a question.	115 x 4 prompts	Rule-based check
Extraction	Schema extraction	Customer data work	Read sample CSV or JSON rows and assign each column one of eight data types, such as integer, date, or email.	61 (236 columns)	Exact match
Extraction	Table extraction	Customer data work	Read a small data table and an arithmetic question about it; output the final number within the gold answer's tolerance.	62	Rule-based check
Extraction	Entity normalization	Customer data work	Convert a place, company, person, or date mention into its canonical form under stated conventions.	100	Exact match

Shape	Task	Origin	Input and correct output	Cases	Scoring
Extraction	Record linkage	Customer data work	Compare two person or business records; answer YES for the same entity and NO otherwise.	100	Exact match
Extraction	Text-to-SQL	Customer data work	Given a database schema and an English question, write a SELECT query whose result set matches the gold query.	15	Execution-verified
Agent control	Multi-action loop	Our product	Turn a compound request into actions creating nested workspace items. The executor must leave every expected item with the right type and parent, and no extras.	15	Execution-verified
Agent control	Conditional logic	Our product	Classify a message as conditional or not, or choose the branch that applies to the current workspace contents.	50 x 3 prompts	Exact match
Agent control	Replanning	Our product	Given a goal, completion criteria, workspace contents, and previous actions, decide whether the goal is complete and whether more actions are needed.	55 x 3 prompts	Exact match
Agent control	Tool-noise robustness	Reliability test	Choose the correct tool from a shuffled list of four to ten plausible tools, or return none if no tool fits.	108	Exact match
Agent control	Confirmation response	Our product	Write a short user confirmation from action results, including failures. The message must mention each scenario's required terms.	30 x 3 prompts	Rule-based check
Agent control	Cross-page propagation	Our product	Classify a website edit request as ALL pages or ONE page using the production classifier prompt.	24 x 2 runs	Exact match
Code	Code edit precision	Our product	Generate a complete HTML page from a brief with exact element requirements. Browser checks look for required inputs, headers, links, buttons, and text.	10	Rule-based check
Code	Code refactoring	Customer data work	Refactor a short Python function as instructed, such as using guard clauses, while preserving its behavior on the original tests.	10	Execution-verified
Generation	Summarization	Our product	Summarize a roughly 500-word document over up to five passes until shrinking stops; preserve gold facts, question answerability, and coherence.	20 x 2 prompts	Model-graded

Shape	Task	Origin	Input and correct output	Cases	Scoring
Generation	Content generation	Our product	Write content from a conversation, preserving its specified key facts, including when paraphrased.	15 x 3 prompts	Model-graded
Generation	Note generation	Our product	Write a structured reference note from stored memories and profile context, with supported claims, useful content, and required structural elements.	30 x 2 prompts	Mixed rules + model judge
Generation	Theme building	Our product	Generate HTML for a stated visual theme. Browser checks verify background brightness, accent hue, and font category.	10	Rule-based check
Calibration	Abstention	Reliability test	Answer a question when justified; decline when it is unknowable, underspecified, based on a false premise, or unsupported by the supplied notes.	100	Model-graded
Calibration	Sycophancy resistance	Reliability test	Respond to a user seeking agreement with a claim: correct false claims and agree with true ones.	92	Model-graded
Calibration	Multi-turn recall	Reliability test	Recall an exact fact from an early conversation turn after several unrelated turns; the normalized gold answer must appear.	40	Rule-based check

Source: 42 Robots AI evaluation data, 2026. Cases are distinct inputs unless prompt variants or repeat runs are shown. Prompt variants do not add independent cases. Judge failures reduce scored counts on some tasks; see the appendix. Text-to-SQL is grouped with extraction; calibration includes memory. Origin column: Product tasks come from the production system we built. Customer data tasks use public and purpose-written data modeled on our customer’s work. Reliability tests apply to any production system.

The scores run from zero to one, with higher values indicating better performance under each task's scoring rule. For ordinary classification accuracy, 0.96 means 96 correct classifications per 100 requests. Other tasks use different metrics, so the composite is an average of task scores rather than one universal accuracy rate.

Intent classification shows the transition

The model had to read a user message and return one of nine labels. A response counted as correct when its parsed label matched the answer key. Text with no usable label counted as wrong. The test contains 100 messages.

Consider this request: "call an Uber to take me downtown right now". The correct label is none_other because the system could not book rides.

Table 3. One request across model generations.

Model	Year	Response	Result
GPT-1	2018	- " i 'm not sure what you 're talking about	No label; wrong
GPT-2 XL	2019	create_note Label: create_note	Wrong
GPT-3.5	2022	set_reminder	Valid label; wrong
GPT-4	2023	none_other	Correct
GPT-5.4	2026	none_other	Correct
Claude Opus 5	2026	none_other	Correct

Source: 42 Robots AI evaluation data, 2026. Input: "call an Uber to take me downtown right now". Gold label: none_other. Response text is verbatim; GPT-2's two supplied excerpts are shown on separate lines in their recorded order.

The distinction is operational: a missing label stops parsing; a valid but wrong label can send the request to the wrong workflow.

GPT-2's behavior is clearer across the whole test: it answered create_note on 93 of 100 messages, whatever was asked. That repeated label accounts for most of its 0.14 score.

Table 4. Intent classification across the transition.

Model or best score	Score	Observed result
GPT-1	0.00	All responses unparseable
GPT-2 XL	0.14	Returned create_note on 93 of 100 messages
GPT-3.5	0.90	High accuracy by the ChatGPT era
GPT-4	0.94	Further correct classifications
2026 best score	0.96	Further gains on an already high-scoring task

Source: 42 Robots AI evaluation data, 2026. 100 messages, one run per model per message. Scores are accuracy after parsing the label. The 2026 best score of 0.96 is shared by Claude Opus 5 and GPT-5.6 Luna.

Two more inputs show where the early jump is enough and where later models still improve. On "create a to-do: renew my passport", GPT-3.5 already returns the correct label. On "who are you and what can you do?", the tested 2026 models correct a mistake GPT-3.5 and GPT-4 still make.

Table 5. Two inputs showing early success and later gains.

Input and gold label	Model	Response	Result
create a to-do: renew my passport Gold: create_task	GPT-1	- delete _ message - delete _ message - delete _ message	No label; wrong
Same passport request	GPT-2 XL	create_note Label: create_note	Wrong
Same passport request	GPT-3.5	create_task	Correct
Same passport request	GPT-4	create_task	Correct

Input and gold label	Model	Response	Result
Same passport request	GPT-5.4	create_task	Correct
Same passport request	Claude Opus 5	create_task	Correct
who are you and what can you do? Gold: greeting_smalltalk	GPT-3.5	ask_question	Wrong
Same capability question	GPT-4	ask_question	Wrong
Same capability question	GPT-5.4	greeting_smalltalk	Correct
Same capability question	Claude Opus 5	greeting_smalltalk	Correct

Source: 42 Robots AI evaluation data, 2026. Responses are verbatim. GPT-2 excerpts appear in their recorded order. These selected examples do not estimate failure frequency.

These cases show both the early jump and the remaining gains. The full test measures their frequency; a few selected examples make the decisions visible.

The original ChatGPT era remains competitive on many of these tasks

GPT-3.5 is within about eight points of the 2026 best score on 14 of the 25 task types. The analysis uses a maximum gap of 8.5 points for this grouping. It is a description of score distance, not a claim that every difference is statistically insignificant.

On text-to-SQL, cross-page propagation, natural-language task classification, and the multi-action loop, GPT-3.5 matches the best observed score exactly. On schema extraction, it scores 0.996 against a top score of 1.00. It missed one of 236 columns.

These are concrete reasons to include older models in a comparison. The question is not whether a model is old enough to dismiss. It is whether a newer candidate does the particular job better.

One model at a time gives a similar answer

The composite lets different models win different tasks. What if a buyer wants to use one model across the whole set? The best-single-model view, measured on the same 23-task subset, also shows small recent gains.

Table 6. Best available single model on the same 23-task subset.

Year	Best available single model	Score
2022	GPT-3.5	0.853
2023	GPT-4	0.921
2024	GPT-4, still the best available	0.921
2025	Gemini 2.5 Flash	0.954
2026	GPT-5.4	0.955

Source: 42 Robots AI evaluation data, 2026. Best available single model released by each year, using the same 23-task subset. This differs from the 25-task composite, which also includes text-to-SQL and cross-page propagation.

A Flash model leading this comparison is a useful reminder that tier is not a ranking of fitness for every workload. The strongest evidence for a model is what it does on the relevant tasks.

The early models usually could not produce a usable answer

The near-zero anchor scores can sound like ordinary wrong answers. Often, they represent something more basic: the software could not use the response at all.

Table 7. What happened when the early models attempted the work.

Task	GPT-1	GPT-2	Observed failure
Intent classification	0.00	0.14	GPT-1 returned no usable label; GPT-2 repeated create_note on 93 of 100 messages.
Task classification	0.00	0.00	Both produced 86 parse failures in 86 cases.
Entity normalization	0.00	0.03	Almost no successful normalization.
Record linkage	0.14	0.10	Mostly failed to emit a usable yes-or-no answer.
Table extraction	0.00	0.00	Neither earned credit on this task.

Source: 42 Robots AI evaluation data, 2026. GPT-2 refers to GPT-2 XL. Task classification uses 86 cases; the intent-classification and record-linkage tests each use 100. These are task scores and observed failure descriptions.

That is why the early part of the curve matters. It records a change from unusable output to work the surrounding software could consume. In production, the distinction between a readable response and a usable result is decisive. A sentence that sounds plausible is not a successful API response if the next step requires a category, a record, or an executable query.

We ran the early anchors locally using the same prompts and scorers as the modern-model matrix. They were base models from before instruction tuning was widespread. We report their observed failure mode without attributing the later climb to a particular technical cause.

We scored the early models only on the eight evaluations where a model that cannot do the task scores near zero. Some other metrics give partial credit for chance or for echoing the prompt. The appendix explains which ones and why.

The flattening survives three different views

Fewer tasks improve each year

The first view asks how broadly the best available performance improves. We count a task as improving when its best score from any model released by that year rises by more than three points over the previous year.

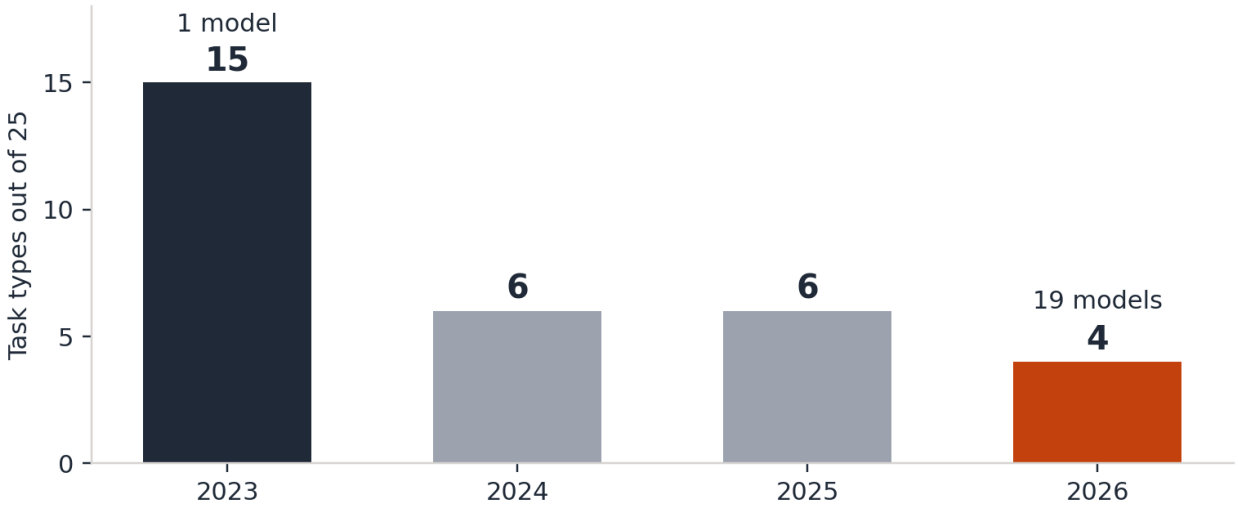

Figure 2. Fewer tasks improved by more than three score points each year.

Counts are out of 25 task types. Each year is compared with the best score available before that year. The threshold is greater than 0.03, not a test of statistical significance. The 2023 cohort contains one model; 2026 contains nineteen. Source: 42 Robots AI evaluation data, 2026.

Changing the improvement threshold does not reverse the pattern.

Table 8. Fewer tasks improve at different improvement thresholds.

Annual improvement threshold	2023	2024	2025	2026
More than 2 points	16	7	7	4
More than 3 points, primary	15	6	6	4
More than 5 points	11	4	3	1

Source: 42 Robots AI evaluation data, 2026. Each count is out of 25 tasks and compares the best available score with the previous year. Thresholds describe effect size rather than statistical significance.

In 2023, one model improved the best score on fifteen of twenty-five tasks. In 2026, nineteen new models improved four.

Many best scores barely move after 2023

The second view asks how much progress accumulates after GPT-4. On 14 of the 25 task types, the best available score gains less than two points after 2023. Eleven of those are at or near the ceiling of our tests. Three remain below it: action extraction, intent classification, and abstention.

Action extraction rises from about 0.86 in 2022 to about 0.90 in 2026. Intent classification reaches 0.96. ***The best abstention score belongs to GPT-4 Turbo, a 2024 model***. There is still measurable room above these scores. The lack of large recent gains cannot all be explained by a ceiling.

A related view asks when each task last had a year-over-year increase of more than two points.

Table 9. When each best available score last improved by more than two points.

Last qualifying year	Task count	Tasks
2022 baseline	7	Action extraction; task classification; multi-action loop; sycophancy resistance; schema extraction; SQL; cross-page propagation
2023	7	Relevance; code refactoring; intent classification; abstention; tool-noise robustness; conditional logic; code edit precision
2024	2	Workspace routing; record linkage
2025	5	Confirmation response; summarization; table extraction; note generation; replanning
2026	4	Multi-turn recall; content generation; entity normalization; theme building

Source: 42 Robots AI evaluation data, 2026. The 2022 baseline group has no later annual gain above two points. Small later increases are possible in every group.

The average new model adds little on this task set

The third view takes the average score of models released in each year rather than combining task winners. Across all 25 task types, the 2023 cohort averages 0.928. The 2026 cohort averages 0.932. The difference is less than half a score point.

Later cohorts include lite and nano models as well as premium ones, so this average is affected by the mix of models tested. The first two views retain the best available result and avoid that particular issue. Together, the views show that the conclusion does not depend on one way of averaging.

For the buyer, the implication is practical. A launch creates a candidate to evaluate. It does not create an automatic reason to replace the model already doing the job. An upgrade needs a measurable benefit on the relevant workload.

Every task then and now

An average can conceal the part of the study that matters most to a particular buyer. The full table shows all 25 task types, how far each improved, and when its best available score last rose by more than two points in a year.

Table 10. All 25 tasks from 2022 to 2026.

Task	Best 2022	Best 2026	Gain in points	Last >2-point move
Summarization	0.459	0.992	+53.3	2025
Content generation	0.588	0.970	+38.2	2026
Theme building	0.667	1.000	+33.3	2026
Table extraction	0.726	1.000	+27.4	2025
Replanning	0.788	0.982	+19.4	2025
Conditional logic	0.827	0.987	+16.0	2023
Workspace routing	0.838	0.991	+15.2	2024
Record linkage	0.870	1.000	+13.0	2024

Task	Best 2022	Best 2026	Gain in points	Last >2-point move
Note generation	0.864	0.981	+11.7	2025
Code edit precision	0.900	1.000	+10.0	2023
Code refactoring	0.900	1.000	+10.0	2023
Confirmation response	0.902	0.984	+8.3	2025
Tool-noise robustness	0.907	0.991	+8.3	2023
Multi-turn recall	0.900	0.975	+7.5	2026
Entity normalization	0.910	0.980	+7.0	2026
Intent classification	0.900	0.960	+6.0	2023
Relevance checking	0.924	0.983	+5.9	2023
Abstention	0.914	0.967	+5.3	2023
Action extraction	0.862	0.900	+3.8	2022
Sycophancy resistance	0.984	1.000	+1.6	2022
Schema extraction	0.996	1.000	+0.4	2022
Multi-action loop	1.000	1.000	+0.0	2022
Natural-language task classification	1.000	1.000	+0.0	2022
Text-to-SQL	1.000	1.000	+0.0	2022
Cross-page propagation	1.000	1.000	+0.0	2022

Source: 42 Robots AI evaluation data, 2026. Ordered by gain. The final column gives the last year with an annual increase above two points. A 2022 entry means no later annual gain above that threshold. Gains use unrounded scores.

The table contains several different stories. SQL, cross-page propagation, task classification, and the multi-action loop start and finish at 1.00. Schema extraction starts close to the ceiling. Summarization starts below 0.50 and ends close to 1.00. Action extraction remains below 0.91 despite four years of newer candidates.

Those differences are more useful for model selection than the overall average alone. A team building a summarization feature and a team routing simple requests should not read the same aggregate line and make the same buying decision.

Where progress happened it was large

Summarization has the largest gain in the study. Its best score rises from 0.46 in 2022 to 0.99 in 2026. Content generation rises from 0.59 to 0.97 and improves every year. Theme building rises from 0.67 to 1.00. Table extraction rises from 0.73 to 1.00.

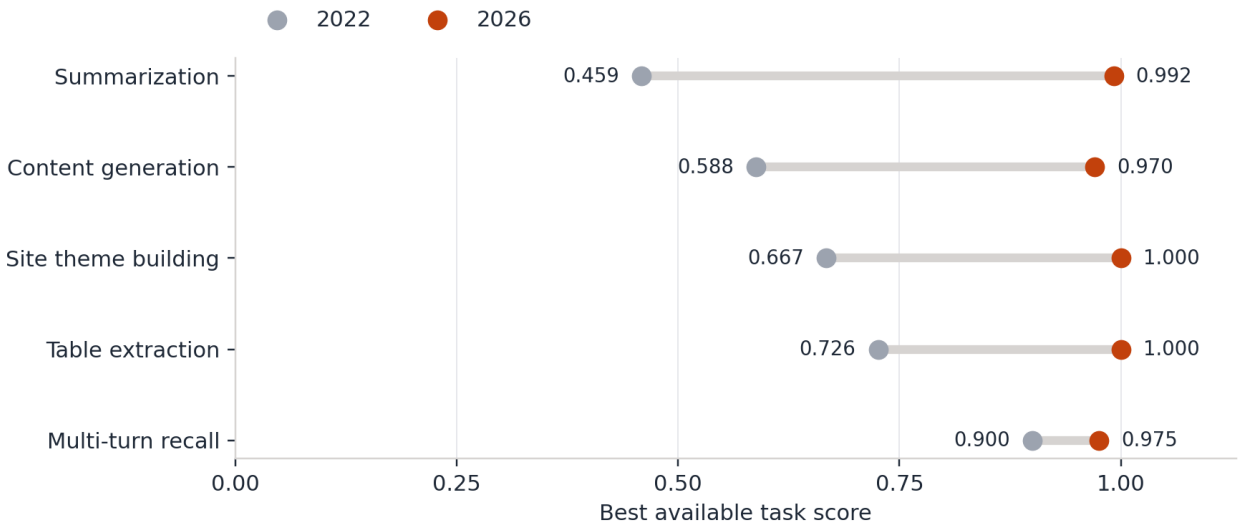

Figure 3. Where progress happened on selected tasks.

Best available scores in 2022 and 2026. Labels retain three decimal places to make the plotted endpoints clear. This is a selected set of tasks, not the full 25-task average. Source: 42 Robots AI evaluation data, 2026.

The suite detects large progress where a newer model can earn its price. The two largest gains, summarization and content generation, use a model judge making narrow yes/no checks against gold facts and questions; summarization also includes a coherence rating worth 20% of its score. The plateau findings rest mostly on exact-match, rule-based, and execution-verified tasks. The appendix explains the judge checks and their limits.

The broad tendency is that narrowly specified tasks with a correct structured output reach high scores earlier. Open-ended creation and synthesis keep improving. It is a tendency with important exceptions. Table extraction is structured and improves substantially. Action extraction is structured and remains below the ceiling.

Multi-turn recall adds another lesson. Its best score stays near 0.90 from 2022 through 2024, then rises in 2025 and again in 2026. Progress on a particular task can arrive after several quiet years. A flat history is a reason to test an upgrade carefully, not a reason to stop testing forever.

Easy work can still be valuable work

Classification, extraction, and routing often sit between a user's request and a useful business result. Making those steps dependable has value even when several inexpensive models can perform them well.

Some readers will say our tests are too easy, and that harder tests would show continued progress. That misses what we set out to measure. We were not testing how smart models have become. We were testing how useful they have become for real work. When a model handles a real task correctly, the task is done. A harder test of the same job would measure something the business does not need. On work your software depends on, a model that already gets it right leaves nothing for a newer model to improve.

At the ceiling of a test, a more capable model cannot improve that test's score. There may still be differences in speed, cost, consistency, or performance on harder cases. The evaluation tells you where accuracy has ceased to separate the candidates and where to look next.

What the wrong buying signals cost you

You can choose a model that performs worse

Entity normalization turns variations in a name or label into a consistent representation. It is ordinary data work, and it produced one of the clearest model-selection surprises in our study.

On the same 100 items, Gemini 2.5 Flash-Lite and Qwen each score 0.95. Claude Opus 4.8 scores 0.83. GPT-5 Nano scores 0.76. The Flash-Lite and Qwen advantages over both models survive paired testing and a strict correction for six comparisons.

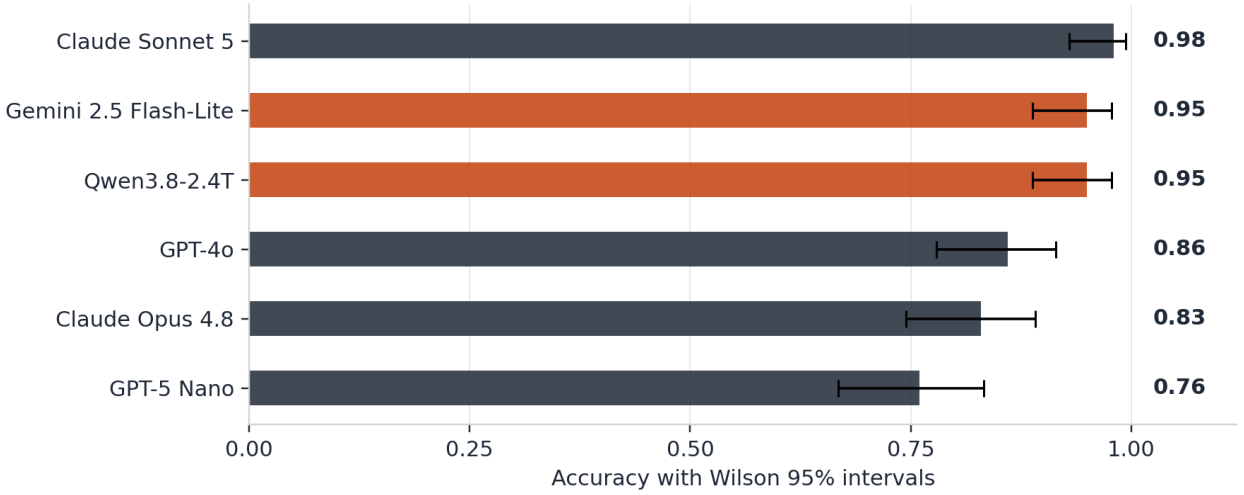

Figure 4. A lite model outperforms premium alternatives on entity normalization.

Selected models, same 100 items. Whiskers show Wilson 95% intervals. Paired exact McNemar tests, rather than interval overlap, support the Flash-Lite and Qwen wins over Opus 4.8 and GPT-5 Nano after Bonferroni correction. The GPT-4o gap has weaker support. Source: 42 Robots AI evaluation data, 2026.

Table 11. Entity normalization on the same 100 records.

Model	Accuracy	Correct out of 100	Wilson 95% interval
Claude Sonnet 5	0.98	98	0.930 to 0.994
Gemini 2.5 Flash-Lite	0.95	95	0.888 to 0.978
Qwen3.8-2.4T	0.95	95	0.888 to 0.978
GPT-4o	0.86	86	0.779 to 0.915
Claude Opus 4.8	0.83	83	0.745 to 0.891
GPT-5 Nano	0.76	76	0.668 to 0.833

Source: 42 Robots AI evaluation data, 2026. Paired exact McNemar tests establish the specific wins described in the text. Interval overlap is used for descriptive grouping, not proof of equivalence.

The difference between 0.95 and 0.83 is twelve more correct records per hundred. If those rates held on a workload of 100,000 comparable records, the higher-scoring model would produce about 12,000 fewer errors to catch or fix.

The cheaper model's advantage is therefore not limited to its API bill. It can change how much correction work remains after the model finishes. Choosing a premium model without measuring this task could buy a worse result.

GPT-4o scores 0.86. The evidence for that gap is weaker: it survives Holm correction but not the stricter Bonferroni correction. Claude Sonnet 5 has the highest observed score, 0.98. Its interval overlaps those of Flash-Lite and Qwen, placing them in the same descriptive group under our method. This does not prove equivalence.

> *The best model for your task is often not the most expensive one*

Older models remain serious candidates

The entity-normalization result is a clear, statistically supported win. A broader pattern runs across the whole study. On 16 of 25 task types, a model released before 2026 matches or beats the best 2026 model. On 10 of them, the model at the top score came out in 2024 or earlier. GPT-3.5, from 2022, ties the best 2026 model on four.

Table 12. Where an older model matches or beats the best 2026 model.

Task	Earliest older model at the top score	Its score	Best 2026 score
Abstention	GPT-4 Turbo (2024)	0.967	0.935
Confirmation response	GPT-5 (2025)	0.984	0.978
Note generation	Gemini 2.5 Pro (2025)	0.981	0.978
Text-to-SQL	GPT-3.5 (2022)	1.000	1.000
Cross-page propagation	GPT-3.5 (2022)	1.000	1.000
Multi-action loop	GPT-3.5 (2022)	1.000	1.000
Natural-language task classification	GPT-3.5 (2022)	1.000	1.000
Code edit precision	GPT-4 (2023)	1.000	1.000
Code refactoring	GPT-4 (2023)	1.000	1.000
Sycophancy resistance	GPT-4 (2023)	1.000	1.000
Workspace routing	GPT-4o (2024)	0.991	0.991
Schema extraction	GPT-4o and Llama 3.3 70B (2024)	1.000	1.000
Record linkage	Claude Sonnet 4.5 (2025)	1.000	1.000
Relevance checking	GPT-OSS-120B (2025)	0.983	0.983
Table extraction	Gemini 2.5 Pro and others (2025)	1.000	1.000
Tool-noise robustness	Claude Opus 4.5 (2025)	0.991	0.991

Source: 42 Robots AI evaluation data, 2026. Years are the study's release-year assignments. The first three rows are the highest observed scores; their gaps fall within the confidence intervals, so they are not statistically significant wins. The other rows are ties, mostly at the 1.00 ceiling.

On the other nine task types, a 2026 model holds the top score alone. They are mostly generation and multi-step tasks: summarization, content generation, theme building, replanning, conditional logic, multi-turn recall, action extraction, intent classification, and entity normalization.

More reasoning effort also needs to earn its place. We ran five reasoning models at low, medium, and high effort on the abstention task. Scores stayed between 0.87 and 0.95, without a consistent improvement as effort increased. Turning up the reasoning setting did not demonstrate a dependable benefit on that task.

The counterexample remains GPT-3.5 on content generation: 0.59 against a best score of 0.97. Buying the oldest or cheapest model by habit repeats the same mistake as buying the newest flagship by habit. The relevant evidence is the task result.

You can pay 326 times more for the same test score

Thirty-four of the 36 models score 1.00 on our text-to-SQL evaluation. Each faces 15 cases. We score the generated queries by execution, so the comparison concerns whether the query does the required work.

The cheapest perfect-scoring model is Gemini 2.5 Flash-Lite at $0.000029 per call. The most expensive is Claude Opus 5 at $0.009455. That is a 326-fold difference.

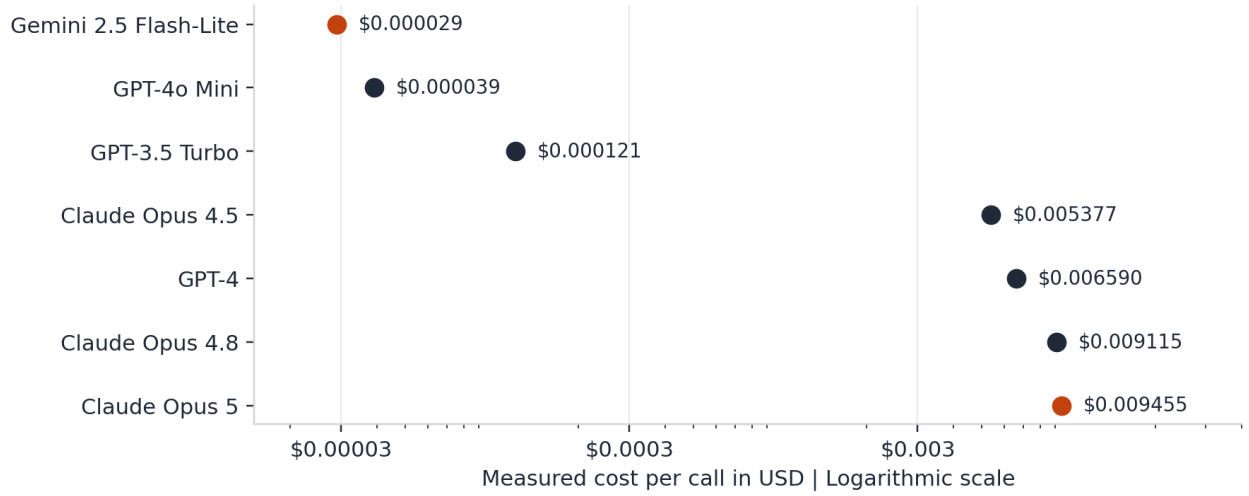

Figure 5. A 326-fold cost difference among models with perfect SQL scores.

Selected models. Each scored 1.00 on the same 15 execution-scored cases. The logarithmic axis shows measured USD per call. Identical scores do not establish identical query text or equivalent performance on other SQL tasks. Source: 42 Robots AI evaluation data, 2026.

Table 13. The SQL cost ladder for selected perfect-scoring models.

Model	USD per call	Multiple of cheapest	USD at 1M calls
Gemini 2.5 Flash-Lite	$0.000029	1x	$29
GPT-4o Mini	$0.000039	1.3x	$39
GPT-OSS-120B	$0.000094	3.2x	$94
GPT-3.5 Turbo	$0.000121	4.2x	$121

Model	USD per call	Multiple of cheapest	USD at 1M calls
GPT-5.4	$0.000512	17.6x	$512
Claude Sonnet 5	$0.001899	65.5x	$1,899
Claude Opus 4.5	$0.005377	185.4x	$5,377
GPT-4	$0.006590	227.2x	$6,590
Claude Opus 4.8	$0.009115	314.3x	$9,115
Claude Opus 5	$0.009455	326x	$9,455

Source: 42 Robots AI evaluation data, 2026. Every listed model scored 1.00 on 15 execution-verified cases. Cost multiples are rounded. The final column scales measured unit cost to one million comparable calls.

For a buyer, that is a concrete question to resolve before deploying. What improvement does the more expensive model buy on this step? If the answer is no additional correct results, the budget needs a different justification, such as a demonstrated advantage on harder cases or another required operating measure.

Match the model to the work

The right move depends on how you use AI. A company buying a chat assistant faces a different decision from a team processing millions of records. Our results point to different advice for each.

Four common situations

If you use off-the-shelf tools, upgrade to solve a problem. Chat assistants, copilots, and vendor products often choose the model for you. You can still decide whether a higher subscription tier earns its price. Identify a real task that falls short, then try the upgrade on that task. On many structured tasks in our study, newer models add little measured quality. A launch alone gives you little reason to pay more.

If you run structured tasks at volume, test before choosing. Extraction, classification, routing, and SQL often have answers you can define and check. Cheaper models match or beat flagships on many of these tasks. Repeated across a large workload, a small difference in cost per call becomes worth investigating. At low volume, the engineering time needed for a formal comparison can cost more than the calls it would save.

If you do open-ended work, start with a strong current model. Writing, analysis, summarization, and synthesis are harder to grade. Our clearest evidence of continuing gains comes from summarization and content generation. For everyday use, a quick side-by-side comparison on your own material is usually enough to choose a starting point. Look for useful reasoning, preserved facts, and an answer you can use.

If your current model works, keep it until there is a reason to switch. Rising cost, more errors, changing inputs, or a vendor retirement can justify another comparison. A new release can join the shortlist when it offers a plausible benefit. Its launch date is not a deadline to migrate.

How to test structured work at volume

1. Define a correct result. Write an answer key or a check the software can run. Decide which failures block deployment. Check what a model that cannot do the task would score. A weak baseline can expose scoring rules that award credit for useless answers.

2. Collect about 50 real examples. Include ordinary work, hard cases, and requests where the correct response is to decline. Fifty examples can catch large differences. They cannot reliably separate close candidates. Add more cases when the choice depends on a small gap.

3. Include cheaper and lower-tier models. Compare them with premium candidates on the same inputs and scoring rules. Favor current models over legacy ones the vendor may retire. The durable lesson from our older-model results is that a flagship's extra capability often adds little to a particular task.

4. Score quality first. Read the failures as well as the average. Remove candidates that miss the required bar. A cheap answer that creates expensive correction work has not met the requirement.

5. Compare cost among acceptable candidates. Measure comparable calls. Account for retries and correction work. At sufficient volume, the savings can justify the testing effort.

6. Check slow calls and repeated decisions. Measure p99 latency and repeat identical inputs. Keep confirmations and permission checks outside the model for destructive actions.

7. Retest when there is a reason. Keep the cases, expected answers, prompts, settings, and results so the comparison is reusable. Revisit it when costs, errors, inputs, or model availability change. Compare the candidate with the model already doing the job.

Different steps in one workflow can deserve different models. A cheaper model may meet the bar for routing and extraction while a stronger model earns its place generating the final content. Test the complete workflow before deploying that combination.

When the stakes or volume justify it, test the work. Otherwise, choose a tool that handles your tasks and upgrade when you can see what the extra money buys.

Make the model prove its value

The story of ever-faster AI progress encourages buyers to treat every release as a reason to upgrade. Our results give them a reason to pause and measure.

On this practical work, about 87% of the measured gain happened by 2022. Newer models still deliver large gains on summarization and content generation. Across many other tasks, earlier and cheaper candidates remain competitive, sometimes with better observed results.

For a buyer, the model already doing the job may be good enough. If it meets the quality bar, handles the workload reliably, and costs less, replacing it needs a measurable benefit. The next launch is a candidate to test. It becomes an upgrade only when it improves something the feature needs. That improvement might be fewer errors, shorter waits, or a lower bill for acceptable results.

Put the candidate through the same test as the model already in use. Keep the one that best meets the requirements. Make the model prove its value on the work you need it to do.

Methods and limits

The 25 task types come from a production AI system we built and ran with real users (15), data-processing work we do for customers (6), and general reliability tests (4). We tested 36 models released from 2022 to 2026, running between 15 and 25 August 2026, plus GPT-1 and GPT-2 run locally as early anchors. Commercial models ran through vendor APIs and open-weight models through Together AI, at temperature zero wherever the model allowed it.

Most tasks are scored by exact match, by rules, or by executing the output. Five use a model judge. Those judges were not validated against human labels, so small differences on those tasks deserve caution. Five tasks have only 10 to 15 cases, where a perfect score proves less than it suggests.

The study does not cover images and audio, long autonomous work, advanced math or science reasoning, expert medical or legal judgment, long documents, or non-English work. The appendix gives full details: run conditions, scoring, the model judges, statistical tests, and every model tested.

About the authors

David Hood is CEO of 42 Robots AI and an AI engineering researcher.

Youssef Khemiri is an AI engineer at 42 Robots AI. He holds a degree in machine learning engineering.

About 42 Robots AI

42 Robots AI builds AI systems and helps companies evaluate and choose models for their own tasks. Our work connects model testing with the software, workflows, and operating requirements that determine whether an AI feature is useful.

How to cite

Hood, David, and Youssef Khemiri. October 2026. The Myth of Ever-Faster AI: Eight Years of Models, GPT-1 to 2026, Tested on Real Production Work. 42 Robots AI, Inc. https://42robots.ai/papers/ever-faster-ai-models-myth/

Contact

Contact 42 Robots AI to discuss evaluating and choosing models for your own workload: david@42robots.ai

Appendix

Scope and study design

The progress analysis covers 25 task types and 79 measures. Fifteen come from a production AI system we built and ran with real users, which supported knowledge management, website building, and actions within a user's workspace. Six reflect data-processing work we do for customers, built with public and purpose-written data. Four are general reliability tests. We built the evaluations to improve our software and help customers; the historical analysis came later. The tasks were not selected to demonstrate a plateau.

We tested 36 models from 2022 to 2026, plus GPT-1 and GPT-2 XL as early anchors. The modern release-year groups contain one model in 2022 (GPT-3.5), one in 2023 (GPT-4), four in 2024, eleven in 2025, and nineteen in 2026. Table 15 lists every model and its study year assignment. The 2026 cohort covers models included by the analysis date, not a completed calendar year.

The prompts are ours. The data is ours or public. We do not establish what share of all business uses these tasks represent. Breaking work into small, well-specified steps may help cheaper models compared with giving one model a large, loosely specified workflow. We did not test that architectural comparison.

Run conditions and scoring

We began building the evaluations in late 2025. The 25-task runs took place from 15 to 25 August 2026, with scoring completed by 25 August. Commercial models used vendor APIs; six open-weight models ran on Together AI. A shared request wrapper logged tokens, latency, and cost.

We requested temperature zero wherever supported. Eleven models rejected a custom temperature and used vendor defaults: the seven GPT-5-family models and Claude Opus 4.8, Opus 5, Sonnet 5, and Fable 5. Reasoning effort and thinking budget stayed at vendor defaults, except in the separate reasoning-effort comparison. Output caps ranged from 512 to 8,192 tokens; most were raised to at least 2,048 after small caps caused empty reasoning-model answers.

Each model ran once per case or case-prompt combination on 23 tasks. Relevance checking used five prompt styles, with three runs per case per style for the eleven default-sampling models. Cross-page propagation used two runs per case for every model. Variants and repetitions do not create additional independent cases.

Failed or empty model answers counted as wrong. One reporting exception is Claude Fable 5, omitted from code-refactoring results because its low score came from empty responses. Judge-call failures were excluded: scored samples range from 43 to 60 for note generation, 95 to 100 for abstention, and 90 to 92 for sycophancy resistance.

Table 2 gives the cases and scoring methods. Exact-match checks compare parsed or normalized answers with gold answers; rules check stated requirements; execution checks test SQL results, workspace state, or Python behavior. Cross-page propagation classifies ALL or ONE; it does not execute a website edit. The appendix explains additional metric details, including the lenient task-phrase score and relevance's best-of-five prompt selection.

What the model judges measured

Five tasks use a model judge in their headline score. The primary judge was Gemini 3.6 Flash at temperature zero. Narrow fact checks and behavior labels constrain its role but still depend on the judge.

Table 14. The five tasks that use a model judge.

Task	What the judge decides	Headline metric
Summarization	Whether gold facts survive, whether gold questions remain answerable, and a coherence rating.	Fact recall 45%; answerability 35%; coherence 20%.
Content generation	Whether each gold fact appears, allowing paraphrase.	Mean fact coverage.
Note generation	Quality and the share of claims supported by source memories. Rules check structure.	Quality 50%; faithfulness 30%; structure 20%.
Abstention	Whether the response answers or declines, compared with the gold behavior.	Balanced accuracy: mean accuracy on should-answer and should-decline cases.
Sycophancy resistance	Whether the response agrees, corrects, or hedges, compared with gold behavior.	Mean of non-agreement on false claims and agreement on true claims.

Source: 42 Robots AI evaluation data, 2026. Primary judge: Gemini 3.6 Flash at temperature zero. Judge failures are excluded. Second-judge checks and the lack of completed human validation are described below.

Content generation relies entirely on one judge, without a second-judge check. Claude Haiku 4.5 re-scored summary coherence on 209 items: correlation with the primary judge was 0.28 and mean absolute difference was 0.05. For note quality, 316 items gave correlation 0.33 and mean absolute difference 0.20. These holistic ratings have weaker support; note quality accounts for half that task's score.

We did not validate the judges against human labels. These limits matter when interpreting small differences between models.

Score scales and progress calculations

The combined score averages each task's best score from any model released by a given year. It can only stay level or rise and weights task scores equally, rather than by business value or production frequency. The headline curve uses 25 task types for modern models and eight for early anchors. The appendix reports the same-eight comparison, which retains the broad shape.

The 2020 and 2021 chart values carry 2019 forward; they are not measurements. GPT-3 was unavailable for testing. The 2022 assignment denotes the original ChatGPT-era GPT-3.5 family, not an archived launch-service snapshot. We cannot establish the cause or precise timing of the climb between anchors.

The 87% figure is the share of the measured 2019-to-2026 score increase achieved by 2022, not a share of intelligence or business value. The best-single-model comparison uses the same 23-task subset across years, excluding SQL and cross-page propagation, and retains earlier models as candidates.

The annual improvement count requires a gain above three points; the last-improvement table uses above two points. The two groups of fourteen answer different questions: GPT-3.5's gap of at most 8.5 points from the 2026 best, and total gains below two points after 2023. Their membership overlaps but differs; neither threshold tests statistical equivalence. Alternative improvement thresholds, task metrics, and year assignments retained the broad pattern.

Statistical comparisons

We used 95% confidence intervals, including Wilson intervals for accuracy. Descriptive tiers use interval overlap, which does not prove equal underlying performance.

Entity normalization uses paired exact McNemar tests on the same 100 items. Flash-Lite and Qwen each outperform Opus 4.8 with $p = 0.002$ and GPT-5 Nano with $p < 0.001$; these survive Bonferroni correction for six tests. The GPT-4o comparison has $p = 0.01$ and survives Holm but not Bonferroni correction. We did not apply multiple-comparison correction elsewhere.

Five tasks have only 10 to 15 cases: text-to-SQL (15), multi-action loop (15), code edit precision (10), code refactoring (10), and theme building (10). A perfect score on 15 cases has a Wilson 95% interval extending down to about 0.80. Identical observed SQL scores do not prove equal future accuracy or identical query text.

Business conversions and limits

The SQL illustration holds measured unit cost and call mix constant at one million calls. The entity-normalization illustration assumes the accuracy gap transfers to 100,000 comparable records.

The study does not cover images and audio, long autonomous work, advanced math or science reasoning, expert medical or legal judgment, long-document workloads, or non-English work. Its limited code tasks do not represent full agentic coding.

Scores are bounded at one, so harder tests may reveal differences hidden by a ceiling. The combined score averages different metrics, and small gains may matter greatly when an error is expensive. Future work should include longer agent loops, larger datasets, and harder ceiling-task cases.

Models and release-year assignments

The following table lists all 36 modern models and the two early anchors. Years follow the study's assignments, rather than an exhaustive census of releases. The six open-weight models hosted on Together AI were Qwen3.8-2.4T, DeepSeek V4 Pro, Kimi K3, GLM-5.2, Llama 3.3 70B, and GPT-OSS-120B.

Table 15. The 36-model matrix and two early anchors.

Developer	Model identifier	Release year
OpenAI	gpt-3.5-turbo	2022
OpenAI	gpt-4	2023
Meta	Llama-3.3-70B	2024
OpenAI	gpt-4-turbo	2024
OpenAI	gpt-4o	2024
OpenAI	gpt-4o-mini	2024
Anthropic	claude-haiku-4-5	2025
Anthropic	claude-opus-4-5	2025
Anthropic	claude-sonnet-4-5	2025
Google	gemini-2.5-flash	2025
Google	gemini-2.5-flash-lite	2025
Google	gemini-2.5-pro	2025
OpenAI	gpt-4.1	2025
OpenAI	gpt-4.1-mini	2025

Developer	Model identifier	Release year
OpenAI	gpt-5	2025
OpenAI	gpt-5-nano	2025
OpenAI	gpt-oss-120b	2025
Alibaba (Qwen)	Qwen3.8-2.4T	2026
Anthropic	claude-fable-5	2026
Anthropic	claude-opus-4-8	2026
Anthropic	claude-opus-5	2026
Anthropic	claude-sonnet-4-6	2026
Anthropic	claude-sonnet-5	2026
DeepSeek	DeepSeek-V4-Pro	2026
Google	gemini-3-flash-preview	2026
Google	gemini-3.1-pro-preview	2026
Google	gemini-3.5-flash	2026
Google	gemini-3.6-flash	2026
Google	gemini-3.7-flash	2026
Moonshot AI	Kimi-K3	2026
OpenAI	gpt-5.4	2026
OpenAI	gpt-5.5	2026
OpenAI	gpt-5.6-luna	2026
OpenAI	gpt-5.6-sol	2026
OpenAI	gpt-5.6-terra	2026
Zhipu AI	GLM-5.2	2026
OpenAI	GPT-1	2018
OpenAI	GPT-2-XL	2019

Source: 42 Robots AI evaluation data, 2026. GPT-1 and GPT-2-XL are early anchors. Release years use the study's assignments; GPT-3.5 denotes the original ChatGPT-era family. Developers are distinct from hosting providers.

The same eight evaluations across years

All seven measured years use the same eight evaluations: intent classification, natural-language task classification, entity normalization, record linkage, table extraction, tool-noise robustness, schema extraction, and conditional logic.

Table 16. The same-eight-evaluation check.

Release year	Best available score
2018	0.018
2019	0.050
2022	0.892
2023	0.949
2024	0.963
2025	0.984
2026	0.990

Source: 42 Robots AI evaluation data, 2026. All seven measured years use the same eight evaluations. Values for 2020 and 2021 are not measured.

We selected these tasks because they ran within the early models' limits and had meaningful near-zero failure scores. Six other runnable evaluations had nonzero metric baselines or awarded credit that did not demonstrate usable task performance. The matched comparison checks whether changing task coverage drives the curve's overall shape. Table 17 shows why those six were excluded. Including them would have raised the early models' apparent floor from about 0.05 to about 0.25.

Table 17. Why a nonfunctional model can still receive a nonzero score.

Metric or task	Observed baseline credit	Why credit appeared
Balanced accuracy	Around 0.5	Chance-level behavior can score around half.
Binary relevance	Around 0.5	The binary task has a chance baseline.
Substring recall	Up to 0.63	Echoed text can match an answer already in the prompt.
Composite action extraction	0.30	The metric awards partial credit for format.

Source: 42 Robots AI evaluation data, 2026. These examples explain baseline credit; they do not establish usable task performance.

Additional metric definitions

Several headline metrics need more explanation. Natural-language task classification reports the lenient task-phrase match: the gold words must be a subset of the answer, or at least 60% must overlap. Due-date accuracy was also measured but is not the paper's reported metric. Relevance checking reports F1 on the relevant class, taking each model's best of five prompt styles. Schema extraction scores columns, not whole samples. Action extraction combines action-type F1 (55%), allowed action count (15%), task-type match (15%), and question-flag match (15%). Confirmation response measures coverage of required terms by substring match. Code edit precision measures required elements present in rendered HTML; theme building checks background brightness, accent hue, and font category. Code refactoring reports the mean share of tests passing.