# Local LLM Benchmark: Administrative Tasks

DGX Spark Evaluation

Dmytro Klepachevskyi* Žygimantas Girdauskas* Robert Mackevič*
Tadas Šubonis*

June 24, 2026

**Abstract**

We benchmark 13 locally-run language models on a 21-task administrative work suite (calendaring, document parsing, financial and time-tracking calculations, email triage, and professional writing) on a single DGX Spark with 128 GB unified memory. All models run via the Ollama backend. Grading uses a mix of deterministic Python checks for structured tasks and a local `llama3.1:70b` LLM judge for open-ended writing.

**Gemma4:31b (0.807) and Gemma4:12b (0.802) are statistically tied for first place.** The result is from clean isolated re-runs (runs 0055 and 0044) after an earlier GPU-contention artifact had placed Gemma4:31b near the bottom of the field. The 12B variant is the pragmatic default for production: nearly identical quality at roughly a third of the memory footprint.

**Other strong models specialise.** `Qwen3.6:latest` (0.748) is the only model in the suite to pass `task_pdf_to_calendar` and leads non-Gemma models on automated structured tasks, making it the best choice for document parsing and file generation. `GLM-4.7-flash` (0.737) and `gemma4:31b` lead on open-ended writing (judge-8 mean above 0.88).

**Speed and quality are largely decoupled.** The fastest model tested, `gemma4:12b` at roughly 10 output tok/s, is also a quality co-leader. The slowest, `llama3.1:70b` at under 0.2 tok/s, scores near the bottom, but that is a resource artifact from a parallel run, not a capability signal. Model size is a weak predictor of quality at this scale: the 8–14B band spans nearly the full score range, and the largest model tested (`llama3.1:70b`) underperforms several 8–12B models once its parallel-run contention is accounted for.

**Failure modes fall into two patterns.** Three structured tasks (PDF-to-calendar, financial-ratio, todo-cleanup) resist nearly all models regardless of size, pointing to tooling or task-design issues rather than raw capability gaps. `phi4:14b` is non-functional beyond the sanity check, and `deepseek-r1:32b` writes well (judge-8 0.754) but never completes structured or tool-calling tasks (auto-9 mean 0.111), even with zero timeouts.

*AAI Labs, Vilnius, Lithuania

# Contents

**1 Overview** **3**

**2 Related work** **3**

**3 Task suite** **4**

**4 LLM-as-judge** **6**

**5 Models evaluated** **6**

**6 Scoring methodology** **7**

**7 Results and analysis** **9**
7.1 Overall ranking . . . 9
7.2 Score distribution . . . 9
7.3 Sub-score comparison by grading type . . . 9
7.4 Quality vs. model size . . . 9
7.5 Quality vs. speed . . . 10
7.6 Tokens per second by model . . . 10
7.7 Per-task scores . . . 11
7.8 Top performers . . . 11
7.9 Mid-range models . . . 13
7.10 Structural task failures . . . 14
7.11 Failed or resource-limited runs . . . 15
7.12 Model family observations . . . 16

**8 Conclusions** **16**

**9 Future work** **17**

# 1 Overview

This report evaluates 13 locally-run language models on a 21-task administrative work suite covering calendaring, document parsing, financial calculations, email triage, and professional writing. The goal is to identify which models are best suited for office and knowledge-worker tasks that must operate fully offline on a DGX Spark with 128 GB unified memory.

All models run through the Ollama inference backend and are tested via PinchBench v2.0.0-rc1, which drives each model through a standardised agent harness with a 120-second per-task timeout. Grading combines deterministic Python checks for structured tasks (file creation, numerical accuracy, field validation) with a local `llama3.1:70b` LLM judge for open-ended writing and analysis tasks. The entire evaluation – model inference, judging, and scoring – runs on-device with no cloud dependencies.

**Contributions.** This report provides a systematic comparison of 13 locally-run LLMs on a 21-task administrative work suite at this scale. The benchmark suite is open and reproducible, and the results provide actionable model picks for teams deploying LLM-based agents on edge hardware.

**Setup summary:**

- **Hardware:** DGX Spark, 128 GB unified memory, Ollama backend
- **Framework:** PinchBench v2.0.0-rc1 (`-suite admin`, 21 tasks)
- **Judge:** `ollama/llama3.1:70b` running locally (fully offline)
- **Task timeout:** 120 s per task (agent wall-clock limit)
- **Models evaluated:** 13 (covering Gemma, Qwen, GLM, GPT-OSS, Nemotron, LLaMA, DeepSeek, Phi families)

**Naming convention.** Model tags (e.g. `gemma4:31b`) refer to the Ollama identifier used in commands. Chart labels and paragraph headers use a shortened display name (e.g. Gemma4 31B) for readability. Both forms refer to the same model.

# 2 Related work

**On-device LLM evaluation.** Laskaridis et al. [1] introduce MELT, a benchmarking infrastructure for LLM inference across mobile devices, measuring throughput, power, and Q&A accuracy for models up to 7B. MobileAIBench [2] extends this to on-device LLM and LMM use cases including summarisation and instruction following. Both focus on consumer mobile hardware (Snapdragon, Exynos) and models in the 1–7B range, while our work targets a more capable edge platform (DGX Spark, 128 GB) with models up to 70B. The Edge-LLM competition [3] evaluates sub-3B models on Jetson Orin, again at a smaller scale than reported here.

**LLM agent and tool-use benchmarks.** AgentBench [4] evaluates LLMs as autonomous agents across coding, web browsing, and database tasks, but runs on cloud-hosted models. The Berkeley Function-Calling Leaderboard (BFCL) [5] ranks models on function-calling accuracy with single-call and multi-turn scenarios. SWE-bench [6] tests software engineering via GitHub issue resolution. These benchmarks target frontier cloud models and do not address the constraints of local deployment. Our admin suite occupies a different niche: office productivity tasks (calendaring, email, financial calculations) on a fully local stack.

**LLM-as-judge.** Zheng et al. [7] show that GPT-4 as a judge aligns with human preferences over 80% of the time on MT-Bench and Chatbot Arena. Subsequent work [8] surveys biases in

LLM judges (positional, verbosity, self-enhancement) and proposes mitigation strategies. Thakur et al. report that even strong judges like Llama-3.1 70B diverge by up to 5 points on absolute rating scales. Our use of a local `llama3.1:70b` judge accepts these limitations in exchange for full offline operation; a full cross-judge comparison is left for future work (Section 9).

**Ollama and local inference.** Ollama wraps llama.cpp with an OpenAI-compatible API, making local model deployment practical for non-research users. Performance comparisons between Ollama and vLLM [9] show vLLM has lower time-to-first-token under load, but Ollama's simplicity and single-model serving model suit the sequential benchmark design used here. Prior Ollama benchmarks (e.g. [10]) measure raw throughput but not task-level quality, which is the gap our work addresses.

**DGX Spark benchmarks.** Several sources evaluate LLM inference specifically on the DGX Spark. Signal65 [11] provides a first-look hardware review, highlighting the 128 GB unified memory as the key differentiator versus consumer GPUs. StorageReview [12] reports throughput benchmarks on Spark, including Llama 3.1 8B at FP4 reaching ∼924 tok/s at 128-way concurrency and Qwen3 Coder 30B at FP8 hitting ∼483 tok/s at batch 64, focusing on batch inference rather than agent task quality. Exxact [13] benchmarks agent behaviour (tool calling, argument discipline, multi-step chaining) on Spark using an OpenClaw-based harness with Qwen, Gemma, and Nemotron models, finding that larger agent-oriented models tend to be more reliable over multi-step chains. 2389 Research [14] documents early Spark experiments running Llama 4 Scout (109B) via Ollama and training a LoRA adapter for Llama 3.1 70B, reporting practical experience rather than structured benchmarks. Our work differs from these by evaluating 13 models on a 21-task administrative suite with both deterministic and LLM-judge grading, providing a broader quality comparison than throughput-only or single-model studies.

## 3 Task suite

The `admin` suite contains 21 tasks across three grading types:

- **Automated** (9 tasks): deterministic Python graders check file contents, field values, and structure. No judge required.
- **Hybrid** (4 tasks): half the score comes from automated checks; the other half from the LLM judge evaluating quality, completeness, and style.
- **LLM judge** (8 tasks): fully open-ended writing or analysis tasks graded entirely by the judge on a rubric.

Table 1: Full admin task suite.

| Task ID | Name | What it tests | Grading |
|---|---|---|---|
| *Automated tasks* | | | |
| `task_sanity` | Sanity Check | Smoke test: agent responds correctly through the full benchmark stack. | auto |

| Task ID | Name | What it tests | Grading |
|---|---|---|---|
| `task_calendar` | Calendar Event | Creates an ICS event file from a natural-language request with correct fields. | auto |
| `task_pdf_to_calendar` | PDF to Calendar | Extracts event dates from a school calendar PDF and produces a valid ICS file. | auto |
| `task_todo_list_cleanup` | Todo Cleanup | Reorganizes a noisy todo list, deduplicates items, and flags overdue entries. | auto |
| `task_financial_ratio_calc.` | Financial Ratio | Computes inventory turnover ratio from FY2024 financials and explains the calculation. | auto |
| `task_budget_variance_report` | Budget Variance | Reads a JSON budget, computes per-department variance, flags overruns >10%. | auto |
| `task_meeting_agenda_parser` | Agenda Parser | Parses a plain-text agenda into structured JSON with metadata, items, durations, owners. | auto |
| `task_time_tracking_summary` | Time Tracking | Aggregates CSV time entries per employee and project; finds top contributor. | auto |
| `task_invoice_parser` | Invoice Parser | Extracts vendor, line items, and totals from a plain-text invoice into JSON. | auto |
| *Hybrid tasks (auto + LLM judge)* | | | |
| `task_email_triage` | Email Triage | Sorts inbox by priority and category; produces structured triage with next actions. | hybrid |
| `task_spreadsheet_summary` | Spreadsheet Summary | Analyzes tabular data and writes a findings summary. | hybrid |
| `task_email_search` | Email Search | Searches an email corpus for relevant messages and synthesizes results. | hybrid |
| `task_travel_itinerary_planner` | Travel Itinerary | Plans a 3-day conference trip (Chicago, Sept 14–16) with schedule and budget. | hybrid |
| *LLM judge tasks* | | | |
| `task_daily_summary` | Daily Summary | Concise daily summary from research notes or materials. | llm_judge |
| `task_email` | Email Drafting | Professional email draft from an office communication scenario. | llm_judge |
| `task_email_reply_drafting` | Email Reply | Contextual reply to an incoming professional email. | llm_judge |

| Task ID | Name | What it tests | Grading |
|---|---|---|---|
| task_summary | Document Summary | Compact, accurate administrative summary of a longer document. | llm_judge |
| task_contract_analysis | Contract Analysis | Reviews contract language and extracts legal or operational conclusions. | llm_judge |
| task_eli5_pdf_summary | ELI5 PDF Summary | Explains a PDF document in plain language for a non-expert audience. | llm_judge |
| task_performance_review_summary | Performance Review | Transforms raw manager notes into a polished mid-year performance review. | llm_judge |
| task_weekly_status_email | Weekly Status Email | Writes a professional status update email from bullet-point notes. | llm_judge |

## 4 LLM-as-judge

Hybrid and `llm_judge` tasks are graded by a locally-running LLM judge via the `-judge` flag. The judge receives the task rubric and the model's output and returns a structured score with per-criterion breakdowns and notes. All runs in this report used:

```
--judge ollama/llama3.1:70b
```

The judge model (`llama3.1:70b`) runs on the same DGX Spark via Ollama, through the OpenAI-compatible endpoint at `http://localhost:11434/v1/chat/completions`. It is called sequentially after each task; there is no concurrent GPU contention between the model under test and the judge because Ollama pages models in and out between requests.

The benchmark framework also supports `groq/` and `anthropic/` judge prefixes for cloud-based grading, but all results here are fully local.

**Judge limitations.** `llama3.1:70b` is a capable but not state-of-the-art judge. Scores on writing tasks (especially nuanced rubrics like email reply quality or ELI5 explanations) may shift if re-graded with a stronger model such as `llama3.3:70b` or Claude. Rankings among the top models should be treated as approximate within $\pm 0.03$ on judge-heavy tasks.

## 5 Models evaluated

Table 2 lists all 13 models evaluated in this report.

**Parallel-run models (marked †):** `gpt-oss:latest`, `qwen3:8b`, and `llama3.1:70b` were launched simultaneously via tmux in run 0053, causing GPU contention in Ollama. The parallel run understates their scores relative to what a clean sequential run would produce; treat them as lower bounds.

**Isolated re-runs:** Two models (`gemma4:31b` and `deepseek-r1:32b`) were re-run in isolation (runs 0055 and 0056) after their initial runs were resource-limited by timeouts. The clean results are reported throughout.

Table 2: Models evaluated (Ollama backend, `llama3.1:70b` judge).

| Model tag | Family | Run | Notes | Total |
|---|---|---|---|---|
| `gemma4:31b` | Google Gemma-4 | 0055 | Best overall after clean re-run; top judge mean (0.963). | 0.807 |
| `gemma4:12b` | Google Gemma-4 | 0044 | Co-leader; best hybrid mean; slightly more tokens than 31B. | 0.802 |
| `qwen3.6:latest` | Qwen 3.6 | 0041 | Best on structured output; only model to pass PDF-to-cal. | 0.748 |
| `glm-4.7-flash` | Zhipu GLM-4.7 | 0038 | Second-highest judge mean; fast generation. | 0.737 |
| `gpt-oss:latest` | GPT-OSS | 0053† | Strong overall; parallel run understated auto scores. | 0.687 |
| `nemotron3:33b` | NVIDIA Nemotron | 0034 | 33B NVIDIA assistant; solid writing, high timeout rate. | 0.649 |
| `qwen3.5:27b` | Qwen 3.5 | 0047 | Strong judge scores; 13/21 timeouts capped total. | 0.567 |
| `qwen3:8b` | Qwen 3 | 0053† | Best hybrid mean; parallel run affected other categories. | 0.510 |
| `llama3.1:70b` | Meta LLaMA 3.1 | 0053† | 70B; parallel run caused 14 TOs; score severely understated. | 0.507 |
| `qwen3.5:9b` | Qwen 3.5 | 0036 | Mid-range; consistent but no standout tasks. | 0.462 |
| `llama3.1:8b` | Meta LLaMA 3.1 | 0040 | 8B baseline; best on financial ratio among small models. | 0.456 |
| `deepseek-r1:32b` | DeepSeek R1 | 0056 | Clean re-run; strong writing but fails all structured tasks. | 0.426 |
| `phi4:14b` | Microsoft Phi-4 | 0033 | Non-functional after sanity; zero tool use recorded. | 0.188 |

## 6 Scoring methodology

The total score for each model is the arithmetic mean across all 21 task scores. Each task score is itself a mean over its grading criteria (0.0–1.0 per criterion). For hybrid tasks the criteria are split equally between automated checks and judge rubric items.

Three sub-scores are reported for analysis:

- **Auto-9:** mean of the 9 fully automated tasks. Measures structured output, file creation, and numerical accuracy. No judge involved.
- **Hybrid:** mean of the 4 hybrid tasks. Measures document processing, search, and planning where both correctness and quality matter.
- **Judge-8:** mean of the 8 llm_judge tasks. Measures writing quality, reasoning depth, and professional tone on open-ended tasks.

A task that times out (agent exceeds 120 s) or errors is recorded at its partial score (which may be 0.0 if no output was produced) and flagged in the per-task table.

# 7 Results and analysis

This section presents the benchmark results and interprets them. We first present the overall ranking and score distributions, then examine quality-versus-speed trade-offs, per-task scores, and finally discuss individual model performance, structural task failures, and model family patterns.

## 7.1 Overall ranking

Table 3: Overall ranking across all 13 models on the 21-task admin suite. † = parallel run (GPU contention); scores for these models are understated.

| Rank | Model | Run | Total/21 | Auto-9 | Hybrid | Judge-8 | Tokens | TOs* / Err |
|---|---|---|---|---|---|---|---|---|
| 1 | gemma4:31b | 0055 | **0.807** | **0.759** | 0.604 | **0.963** | 2.86 M | 0 / 0 |
| 2 | gemma4:12b | 0044 | 0.802 | 0.705 | **0.748** | 0.939 | 3.30 M | 2 / 0 |
| 3 | qwen3.6:latest | 0041 | 0.748 | 0.745 | 0.654 | 0.797 | 1.01 M | 7 / 0 |
| 4 | glm-4.7-flash | 0038 | 0.737 | 0.648 | 0.642 | 0.884 | 1.33 M | 6 / 1 |
| 5 | gpt-oss:latest | 0053† | 0.687 | 0.556 | 0.643 | 0.858 | 0.98 M | 7 / 0 |
| 6 | nemotron3:33b | 0034 | 0.649 | 0.589 | 0.537 | 0.773 | 1.26 M | 10 / 1 |
| 7 | qwen3.5:27b | 0047 | 0.567 | 0.395 | 0.525 | 0.782 | 0.41 M | 13 / 1 |
| 8 | qwen3:8b | 0053† | 0.510 | 0.377 | **0.711** | 0.561 | 1.07 M | 8 / 1 |
| 9 | llama3.1:70b | 0053† | 0.507 | 0.207 | 0.508 | 0.843 | 0.15 M | 14 / 5 |
| 10 | qwen3.5:9b | 0036 | 0.462 | 0.247 | 0.525 | 0.674 | 1.00 M | 13 / 1 |
| 11 | llama3.1:8b | 0040 | 0.456 | 0.279 | 0.538 | 0.613 | 0.64 M | 5 / 1 |
| 12 | deepseek-r1:32b | 0056 | 0.426 | 0.111 | 0.478 | 0.754 | 0.19 M | 0 / 0 |
| 13 | phi4:14b | 0033 | 0.188 | 0.111 | 0.233 | 0.253 | ∼0 | 0 / 0 |

*TO = timeout (agent exceeded 120 s wall-clock limit).

## 7.2 Score distribution

Figure 1 shows the mean total score across all 21 tasks. Models marked † (GPT-OSS, LLaMA3.1 70B, Qwen3 8B) were run in parallel and their scores are understated.

## 7.3 Sub-score comparison by grading type

Figure 2 breaks the total score into the three grading categories: Auto-9 (structured tasks), Hybrid (mixed auto + judge), and Judge-8 (open-ended writing). The overall score smooths over individual strengths and weaknesses; the breakdown below makes them visible. DeepSeek-R1 is the most lopsided profile: judge-8 of 0.754 with auto-9 of just 0.111. LLaMA3.1 70B's low auto-9 (0.207) reflects parallel-run GPU contention, not model capability.

## 7.4 Quality vs. model size

Figure 3 plots total score against declared parameter count on a log scale. The relationship is weak: the 8–14B cluster spans nearly the full score range (0.188 to 0.802), meaning a small model can outperform a much larger one depending on training quality and architecture. The two Gemma4 variants outperform every larger model tested, including the 70B LLaMA3.1. In practice, this

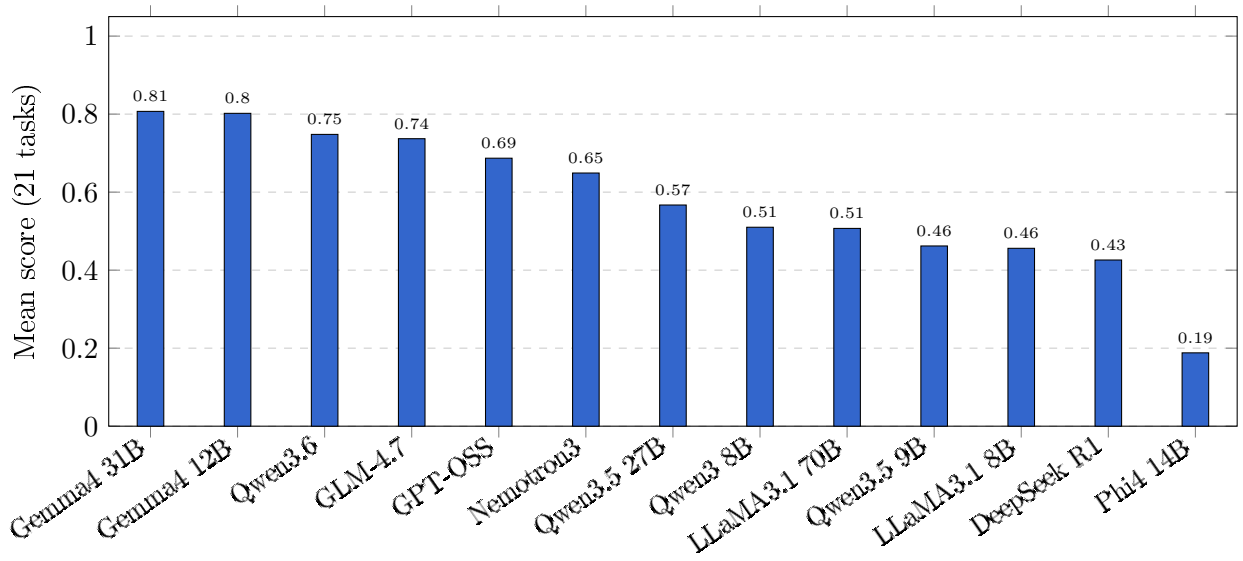


Figure 1: Mean total score across all 21 tasks. † = parallel run (GPU contention); these three models' scores are lower bounds.

means you cannot pick a model by parameter count alone; training quality and architecture matter more at this scale. The largest model tested (`llama3.1:70b`, 70B) sits mid-pack, though its score is understated by parallel-run contention.

Sizes for `gemma4`, `nemotron3`, `qwen3`/`qwen3.5`, `llama3.1`, `deepseek-r1`, and `phi4` come directly from their model tags. `qwen3.6:latest`, `glm-4.7-flash:latest`, and `gpt-oss:latest` carry no size in the tag; they are plotted at 20B, 9B, and 20B respectively as best-effort estimates (`gpt-oss:latest` defaults to the 20B checkpoint in Ollama's library) and should be treated as approximate.

### 7.5 Quality vs. speed

Size is one axis; speed is another. Speed is measured as effective output throughput: total output tokens divided by total wall-clock execution time across all 21 tasks (`efficiency.total_output_tokens` / `efficiency.total_execution_time_seconds` in each run's JSON). This includes tool-call round trips and agent overhead, not just raw decode time, so it reflects what a user would actually experience. Numbers depressed by GPU contention (marked †) understate the model's true generation speed, not its quality. `phi4:14b` produced 0 output tokens and cannot be placed on a log scale; it is omitted from Figure 4 and noted in the caption instead.

### 7.6 Tokens per second by model

The scatter plot in Figure 4 pairs speed with quality. Figure 5 ranks speed alone, without the quality axis. GPT-OSS, DeepSeek-R1, LLaMA3.1 70B, and Qwen3 8B ran under parallel-run GPU contention; their true single-model throughput is likely higher than shown. `phi4:14b` produced 0 output tokens across the entire suite and is included with a zero value.

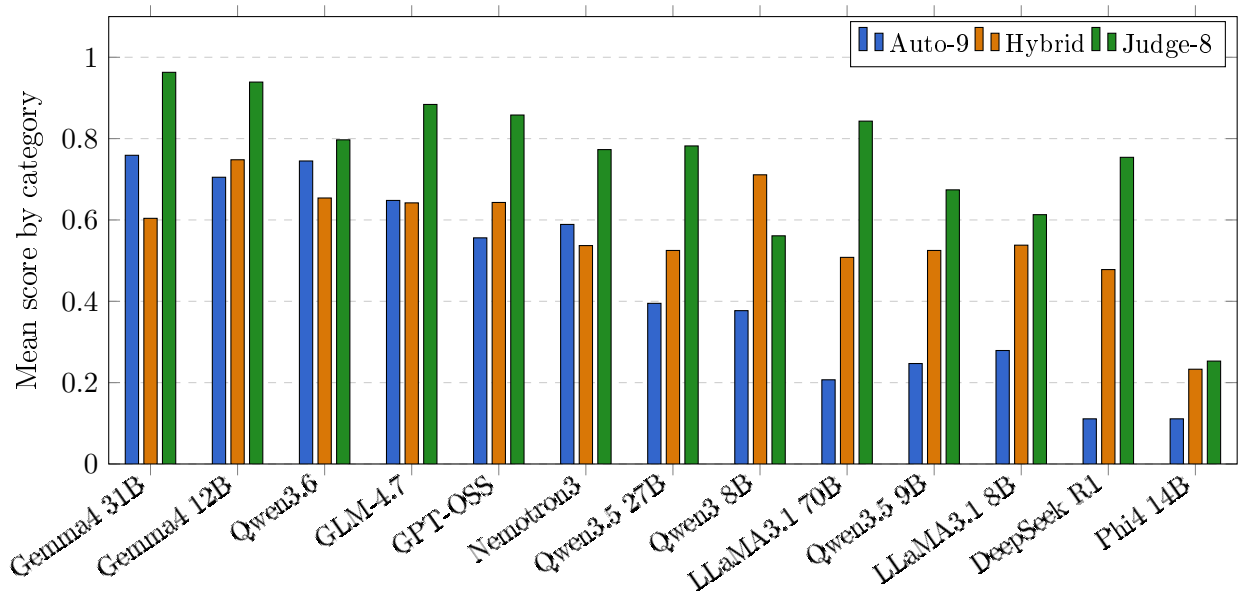


Figure 2: Per-category sub-scores. Automated tasks reveal structured-output capability; judge tasks reveal writing and reasoning quality. DeepSeek-R1's profile is the most lopsided: judge-8 of 0.754 with auto-9 of just 0.111. LLaMA3.1 70B's low auto-9 (0.207) reflects parallel-run GPU contention, not model capability.

### 7.7 Per-task scores

Table 4 shows individual task scores for all 13 models, in overall-rank order. Column abbreviations: G31=gemma4:31b, G12=gemma4:12b, Q36=qwen3.6, GLM=glm-4.7-flash, GPT=gpt-oss, Nem=nemotron3:33b, Q527=qwen3.5:27b, Q38=qwen3:8b, L70=llama3.1:70b, Q59=qwen3.5:9b, L8=llama3.1:8b, DS=deepseek-r1:32b, Phi=phi4:14b. $^{T}$=timed out. $^{E}$=error.

The following subsections discuss individual model performance, structural task failures, and model family patterns.

### 7.8 Top performers

**Gemma4 31B: overall winner after clean re-run (0.807).** Gemma4:31b takes the top spot once re-run in isolation (run 0055), edging out gemma4:12b by 0.005. Its 0.963 judge-8 mean is the highest in the suite, and it is the only model to score above 0.95 across daily summary (1.000), email drafting (1.000), contract analysis (1.000), performance review (1.000), and weekly status (1.000). It also clears the structured-task bar that trips most models: todo-list-cleanup (1.000, the only perfect score on that task), budget variance (1.000), meeting agenda (1.000), time tracking (1.000), and invoice parser (1.000). Its gaps mirror the rest of the field: PDF-to-calendar (0.000), financial ratio (0.000), and spreadsheet summary (0.000). The clean run posted zero timeouts and zero errors on 2.86M tokens, confirming that its original rank-11 placement was a resource artifact, not a capability ceiling.

**Gemma4 12B: co-leader (0.802).** Gemma4:12b is statistically tied with the 31B variant (0.802 vs. 0.807, within judge noise), and actually holds the *best hybrid mean* (0.748) of any model. It achieves perfect or near-perfect scores on the writing set: email triage (1.000), spreadsheet summary

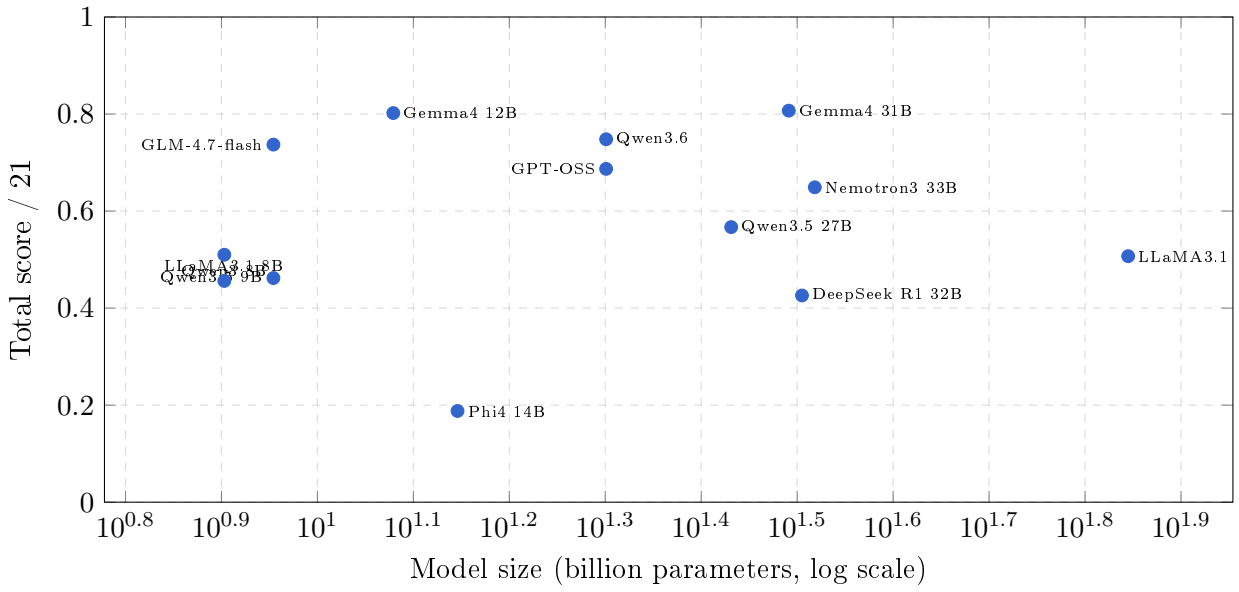


Figure 3: Total score vs. model size. Size is a weak predictor at this scale: the 8–14B cluster spans nearly the full score range (0.188–0.802), and the largest model (LLaMA3.1 70B, 70B) sits mid-pack, though its score is understated by parallel-run contention. The two Gemma4 variants outperform every larger model tested.

(1.000), email search (0.960), contract analysis (1.000), daily summary (1.000), ELI5 PDF (0.820), and text summary (0.950). On structured tasks it scores invoice parser (1.000), budget variance (0.857), meeting agenda (0.833), and time tracking (0.778). Like the 31B it misses PDF-to-calendar and financial ratio. At 12B parameters it is lighter to host than the 31B for nearly identical quality, making it the better default when memory or latency is constrained. **Recommended for production when writing quality and analytical depth matter most.**

**Qwen3.6: best on structured tasks (0.748).** Qwen3.6:latest holds a near-top auto-9 mean (0.745, second only to gemma4:31b) and is the *only model in the evaluation to pass* `task_pdf_to_calendar` (1.000). It also scores perfectly on all four new automated tasks (budget variance, meeting agenda, time tracking, invoice parser) and is one of two models to score on todo-list-cleanup (0.875). Its judge-8 mean of 0.797 trails gemma4:12b noticeably: email reply (0.610), contract analysis (0.512), and text summary (0.740) pull it back. **Best choice when the primary workload is document parsing, structured data extraction, and file generation.**

**GLM-4.7-flash: writing specialist (0.737).** Zhipu GLM-4.7-flash achieves the second-highest judge mean (0.884), reaching perfect scores on email drafting, contract analysis, meeting agenda, invoice parser, travel itinerary, performance review, and weekly status. Its weakness is time-constrained aggregation: todo-list-cleanup times out (0.000), PDF-to-calendar fails (0.000), and time-tracking-summary scores only 0.278. The flash variant generates quickly, contributing to a comparatively low timeout rate (6/21). **Strong candidate when the workload is primarily writing, drafting, and analysis with less emphasis on structured data tasks.**

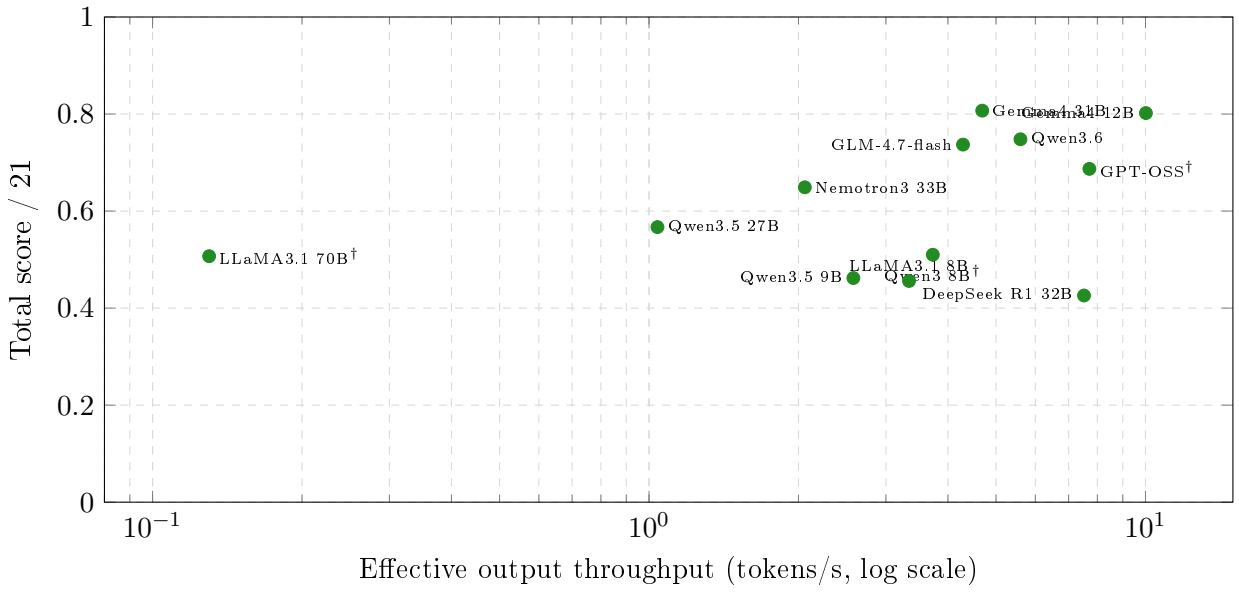


Figure 4: Total score vs. effective output throughput. † = parallel run (GPU contention depresses both speed and score for these three). Quality and speed are largely decoupled: Gemma4 12B is both the fastest model tested and a quality co-leader, while LLaMA3.1 70B's near-zero throughput reflects it barely loading under contention rather than a slow-but-capable model. `phi4:14b` (0 output tokens, 0.188 score) is omitted: it cannot be placed on a log-scale throughput axis.

**GPT-OSS (0.687, parallel run).** GPT-OSS shows a judge mean of 0.858, the third-highest, and achieved perfect scores on email drafting, time tracking, meeting agenda, and invoice parser. However, its auto-9 mean (0.511) is artificially suppressed by the parallel run (run 0053): it was launched simultaneously with llama3.1:70b and qwen3:8b, causing GPU contention that produced 7 timeouts on automated tasks. A clean sequential re-run would likely push its total score into the 0.71–0.75 range, potentially placing it second or third overall.

**Nemotron3 33B (0.649).** NVIDIA's Nemotron3 33B places sixth with a balanced but not outstanding profile (auto-9 0.589, judge-8 0.773). The main drag is a high timeout rate (10/21): several judge tasks that the model likely could have handled with more time simply ran out. Token consumption is reasonable at 1.26M, suggesting the model was generating actively rather than looping.

### 7.9 Mid-range models

**Qwen3.5 27B (0.567).** A strong judge mean (0.782) is almost entirely offset by 13/21 timeouts. When the model completes a task it often does so well (invoice parser: 1.000, travel itinerary: 0.500, perf review: 1.000, weekly status: 1.000), but the 120 s timeout is too tight for a 27B parameter model at Ollama's token generation speed. Re-running with `-timeout-multiplier 2` would give a much clearer picture of its actual capability. Its current score significantly understates its quality.

**Qwen3 8B (0.510, parallel run) and LLaMA3.1 70B (0.507, parallel run).** Both were run in parallel with GPT-OSS in run 0053. Qwen3:8b achieves the highest hybrid mean (0.711) of any model, suggesting it handles multi-faceted document tasks well when it gets GPU time. LLaMA3.1 70B is the most damaged by the parallel run: 14 timeouts and 5 errors on only 0.15M

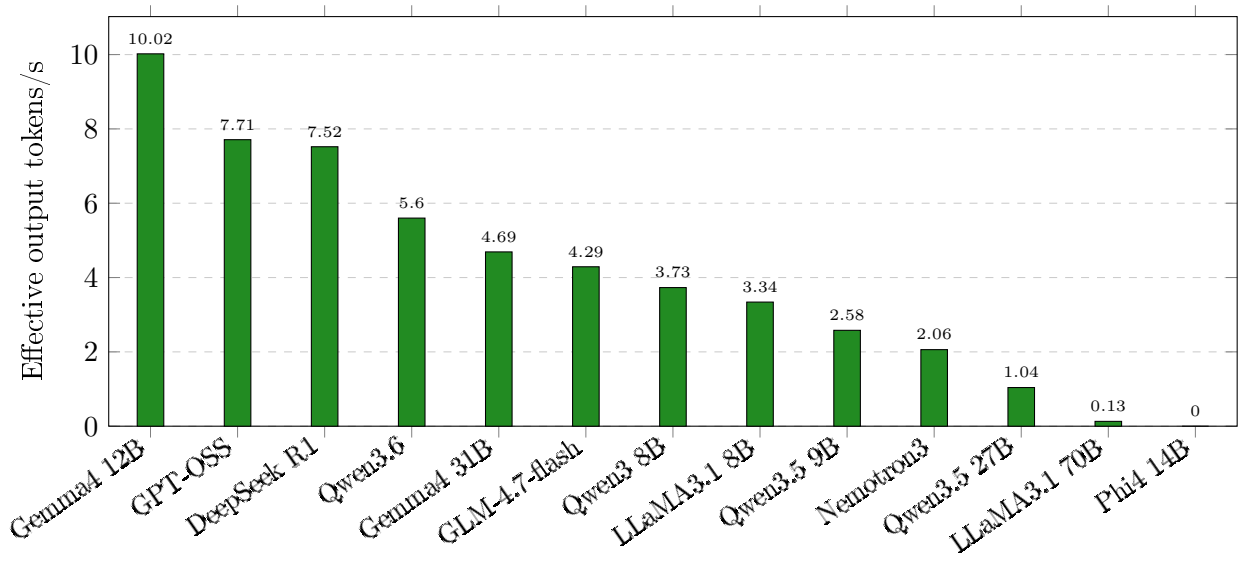


Figure 5: Effective output throughput (output tokens / total execution time) by model, all 21 tasks. GPT-OSS, DeepSeek-R1, LLaMA3.1 70B, and Qwen3 8B ran under parallel-run GPU contention (Section 7); their true single-model throughput is likely higher than shown. Phi4:14b produced 0 output tokens across the suite.

tokens indicates it barely loaded for most tasks. Its judge mean of 0.843 (where it did generate output) suggests substantially better actual capability than its total score implies.

**Qwen3.5 9B and LLaMA3.1 8B.** These sit close together (0.462 vs. 0.456). LLaMA3.1 8B is the only small model to score on financial ratio calculation (0.700), the same partial-credit result as the much larger GLM-4.7-flash and LLaMA3.1 70B. Both models complete tasks reliably but do not excel in any category.

**DeepSeek-R1 32B: the most lopsided profile (0.426, clean run).** The clean isolated re-run (run 0056) reveals a model that is strong at writing but cannot do structured tool-use. Its judge-8 mean is 0.754 (higher than nemotron3:33b, qwen3.5:9b, and llama3.1:8b) with a perfect weekly status (1.000), strong email drafting (0.950), and the joint-best ELI5 PDF (0.860). Yet its auto-9 mean is just 0.111: it passes only the sanity check and scores 0.000 on *all eight* other automated tasks (calendar, todo, financial ratio, budget variance, meeting agenda, time tracking, invoice parser, PDF-to-calendar). This is not a timeout artifact: the run had zero timeouts and zero errors on 0.19M tokens. The reasoning model writes coherent prose but consistently fails to emit the structured files and tool calls the automated graders require. DeepSeek-R1 is a poor fit for agentic, file-producing admin work regardless of timeout budget.

### 7.10 Structural task failures

**PDF-to-calendar: one model passes out of 13.** `task_pdf_to_calendar` was passed by qwen3.6:latest (1.000) and scored 0.000 by every other model, with 5 of the other 12 timing out and 2 erroring. The task requires reading a PDF school calendar, extracting event dates, and generating a valid ICS file. The consistent failure across diverse model families and sizes points to a tooling constraint rather than a model reasoning gap: the agent's PDF-reading tool likely returns

Table 4: Per-task scores for all 13 models. Bold = highest score in row. A=automated, H=hybrid, J=llm_judge.

| Task | T | G31 | G12 | Q36 | GLM | GPT | Nem | Q527 | Q38 | L70 | Q59 | L8 | DS | Phi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Automated tasks* | | | | | | | | | | | | | | |
| Sanity Check | A | **1.000** | **1.000** | **1.000**$^{T}$ | **1.000** | **1.000**$^{T}$ | **1.000** | **1.000**$^{T}$ | **1.000** | **1.000**$^{T}$ | **1.000**$^{T}$ | **1.000**$^{T}$ | **1.000** | **1.000** |
| Calendar Create | A | 0.833 | **1.000** | 0.833 | **1.000** | **1.000** | 0.667 | **1.000**$^{T}$ | 0.000 | 0.000$^{T}$ | 0.833$^{T}$ | 0.667 | 0.000 | 0.000 |
| PDF→Calendar | A | 0.000 | 0.000 | **1.000** | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{E}$ | 0.000$^{T}$ | 0.000 | 0.000$^{T}$ | 0.000$^{E}$ | 0.000 | 0.000 |
| Todo Cleanup | A | **1.000** | 0.875 | 0.875$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000 | 0.000 |
| Financial Ratio | A | 0.000 | 0.000 | 0.000$^{T}$ | **0.700**$^{T}$ | 0.000$^{T}$ | 0.000 | 0.000$^{T}$ | 0.000 | **0.700**$^{T}$ | 0.000$^{T}$ | **0.700** | 0.000 | 0.000 |
| Budget Variance | A | **1.000** | 0.857 | **1.000** | 0.857 | 0.000$^{T}$ | 0.857 | 0.000$^{T}$ | 0.857 | 0.000$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000 | 0.000 |
| Meeting Agenda | A | **1.000** | 0.833 | 0.000$^{T}$ | **1.000** | **1.000** | **1.000**$^{T}$ | 0.000 | **1.000**$^{T}$ | 0.167$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.000 | 0.000 |
| Time Tracking | A | **1.000** | 0.778 | **1.000**$^{T}$ | 0.278 | **1.000** | 0.778$^{T}$ | 0.556$^{T}$ | 0.389 | 0.000$^{T}$ | 0.389 | 0.000 | 0.000 | 0.000 |
| Invoice Parser | A | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.143$^{T}$ | 0.000$^{T}$ | 0.000$^{T}$ | 0.143 | 0.000 | 0.000 |
| *Hybrid tasks* | | | | | | | | | | | | | | |
| Email Triage | H | 0.943 | **1.000** | 0.600 | 0.943$^{T}$ | 0.600 | 0.600$^{T}$ | 0.600$^{T}$ | 0.970 | 0.531$^{E}$ | 0.600$^{T}$ | 0.474 | 0.600 | 0.486 |
| Spreadsheet Summ. | H | 0.000 | **1.000** | 0.025$^{T}$ | 0.025$^{T}$ | **1.000**$^{T}$ | 0.075$^{E}$ | 0.400$^{T}$ | 0.400$^{T}$ | 0.400$^{E}$ | 0.400$^{T}$ | 0.383 | 0.400 | 0.025 |
| Email Search | H | 0.600 | 0.960 | **0.990**$^{T}$ | 0.600 | 0.000 | 0.600 | 0.600$^{T}$ | 0.600 | 0.600$^{T}$ | 0.600 | 0.420 | 0.600 | 0.390 |
| Travel Itinerary | H | 0.875 | 0.031 | **1.000** | **1.000** | 0.972 | 0.875 | 0.500$^{T}$ | 0.875 | 0.500$^{T}$ | 0.500$^{T}$ | 0.875$^{T}$ | 0.312 | 0.031 |
| *LLM judge tasks* | | | | | | | | | | | | | | |
| Daily Summary | J | **1.000** | **1.000** | **1.000** | 0.700$^{E}$ | 0.950 | 0.550$^{T}$ | 0.840 | 0.700 | 0.700 | **1.000** | 0.700 | 0.700 | 0.350 |
| Email Draft | J | **1.000** | 0.950 | 0.950 | **1.000** | **1.000** | 0.950$^{T}$ | 0.150$^{T}$ | 0.950 | **1.000**$^{T}$ | 0.000$^{E}$ | **1.000** | 0.950 | 0.100 |
| Email Reply | J | 0.890 | **0.920** | 0.610 | 0.800$^{T}$ | 0.100 | 0.800 | 0.890$^{T}$ | 0.100 | 0.800$^{E}$ | 0.740$^{T}$ | 0.250 | 0.740 | 0.100 |
| Text Summary | J | **0.950** | **0.950** | 0.740 | **0.950** | **0.950** | **0.950** | 0.740 | 0.610$^{T}$ | 0.610$^{T}$ | **0.950** | 0.650 | 0.410 | 0.100 |
| Contract Analysis | J | **1.000** | **1.000** | 0.512 | **1.000** | **1.000**$^{T}$ | 0.312$^{T}$ | **1.000**$^{T}$ | 0.250 | **1.000**$^{E}$ | 0.512 | 0.500 | 0.500 | 0.250 |
| ELI5 PDF | J | **0.860** | 0.820 | 0.562 | 0.620 | **0.860** | 0.750 | 0.637 | 0.438$^{E}$ | 0.637$^{T}$ | 0.188 | 0.560 | **0.860** | 0.062 |
| Perf Review | J | **1.000** | 0.875$^{T}$ | **1.000** | **1.000** | **1.000** | 0.875$^{T}$ | **1.000** | **1.000**$^{T}$ | **1.000**$^{E}$ | **1.000**$^{T}$ | 0.810 | 0.875 | 0.062 |
| Weekly Status | J | **1.000** | **1.000**$^{T}$ | **1.000** | **1.000** | **1.000** | **1.000**$^{T}$ | **1.000** | 0.438$^{T}$ | **1.000**$^{T}$ | **1.000** | 0.438 | **1.000** | **1.000** |
| **Total/21** | | **0.807** | 0.802 | 0.748 | 0.737 | 0.687 | 0.649 | 0.567 | 0.510 | 0.507 | 0.462 | 0.456 | 0.426 | 0.188 |

insufficient structured text for the model to extract dates reliably. Qwen3.6's success may reflect its stronger pattern-matching on semi-structured OCR output. This task should be treated as an outlier until the PDF tooling is improved.

**Financial ratio: universal partial credit.** Only three models score on `task_financial_ratio_calculation`, and all three score identically at 0.700: GLM-4.7-flash, LLaMA3.1 70B (parallel-run), and LLaMA3.1 8B. All three timed out but produced enough output for partial grading. The consistent 0.700 (rather than 1.000) suggests a shared error: the models compute the inventory turnover ratio using beginning inventory rather than average inventory. Models scoring 0.000 failed to produce any graded output, usually timing out without generating a file.

**Todo-list-cleanup narrows quickly.** Only three models score above zero on `task_todo_list_cleanup`: gemma4:31b (1.000, the only perfect score), gemma4:12b (0.875), and qwen3.6 (0.875). Every other model times out or produces 0.000. The task requires deduplicating, reorganizing, and annotating a 15-item list, which is not a heavy workload; the timeout pattern suggests the models are entering a planning loop rather than directly writing the output file.

**Invoice parser: broad success, then cliff.** Seven of 13 models achieve 1.000 on `task_invoice_parser` (gemma4:31b, gemma4:12b, qwen3.6, glm-4.7-flash, gpt-oss, nemotron3:33b, qwen3.5:27b). Then the scores drop to 0.143 (qwen3:8b, llama3.1:8b) or 0.000 for the remaining four. The two models scoring 0.143 created the file but missed most fields. This suggests a capability threshold around the 14–27B parameter range for reliable structured extraction.

## 7.11 Failed or resource-limited runs

**Gemma4 31B and DeepSeek-R1 32B: resolved by isolated re-runs.** The initial runs of both models (0045, 0046) were resource-limited: 14/21 timeouts with near-zero token consumption, caused by model loading and generation not fitting inside the 120 s per-task budget while the

GPU was contended. Re-running each in isolation (runs 0055 and 0056) eliminated the problem entirely: both posted zero timeouts and zero errors. The clean results overturn the earlier picture: gemma4:31b is in fact the top model overall (0.807, Section above), and DeepSeek-R1 is a capable writer that simply cannot do structured tool-use (0.426). Their results in all tables here are the clean re-runs.

**Phi4 14B: non-functional after sanity.** Phi4:14b passes the sanity check (1.000) but produces no meaningful tool use on any subsequent task. Total recorded tokens are effectively zero (<1K across 20 tasks). The model appears to receive the task prompts but either fails to invoke tools or produces responses that the agent framework cannot parse as tool calls. This failure pattern is almost certainly an initialization or auth failure rather than a capability issue with phi-4 as a model. A re-run with `-verbose` is needed to diagnose the exact failure point.

**Three models in a single parallel run (0053).** LLaMA3.1 70B, GPT-OSS, and Qwen3 8B were started simultaneously via tmux with all three sharing the same Ollama instance. The 70B model bears the heaviest cost (14 TOs, 5 errors, 0.15M tokens) because it requires far more memory bandwidth than the other two and likely could not load into memory while the smaller models occupied the VRAM. All three scores from run 0053 should be treated as lower bounds, not final values.

### 7.12 Model family observations

**Gemma 4 family:** The two variants are the top two models overall and are statistically tied (31B 0.807, 12B 0.802). The 31B has a slight edge on judge-graded writing (0.963 vs. 0.939) and structured tasks (auto-9 0.759 vs. 0.705), while the 12B leads on hybrid tasks (0.748 vs. 0.604) and is cheaper to host. For most deployments the 12B is the pragmatic pick; the 31B is worth the extra memory only when peak writing quality is the priority.

**Qwen family:** Shows a coherent progression on structured tasks. Qwen3.6 leads, then Qwen3.5 27B (timeout-limited), Qwen3.5 9B, and Qwen3 8B. Writing task capability follows a similar order but the gap between 27B and 9B is larger there.

**LLaMA 3.1 family:** The 8B model (0.456) outperforms the 70B model (0.507, parallel run contaminated) only because of run-quality issues. The 70B's judge mean when it did generate (0.843) suggests it would rank 3rd–4th in a clean run.

**Chinese-family models (GLM-4.7-flash):** Highly competitive on open-ended writing and close to gemma4:12b overall. The flash variant's speed advantage over full-size models reduces timeouts without obvious quality cost.

**Reasoning models (DeepSeek-R1 32B):** The clean re-run shows the bottleneck is not the timeout but the model's output behavior. DeepSeek-R1 writes well (judge-8 0.754) but almost never emits the structured files and tool calls the automated graders need (auto-9 0.111), even with zero timeouts. Reasoning-distilled models that front-load a long chain-of-thought and then return prose rather than committing to tool calls, are a poor match for agentic, file-producing admin tasks. This is a capability/behavior mismatch, not a resource limit.

## 8 Conclusions

This evaluation tested 13 locally-run LLMs on a 21-task administrative work suite, all running offline on a single DGX Spark. The results are consistent and actionable.

Gemma4:31b and Gemma4:12b are the strongest all-around models, statistically tied at 0.807 and 0.802 respectively. The 12B variant is the pragmatic production choice: nearly identical quality at a third of the memory footprint, and the fastest model tested at roughly 10 output tok/s. Model size is a weak predictor of quality at this scale. The 8–14B band spans nearly the full score range, and the largest model tested (LLaMA3.1 70B) underperforms several smaller models once parallel-run contention is accounted for.

Specialisation matters. Qwen3.6 is the only model to pass PDF-to-calendar and leads on structured tasks. GLM-4.7-flash and Gemma4:31b produce the best writing. DeepSeek-R1 writes well but fails all structured tool-use tasks, and Phi4:14b is non-functional beyond a basic sanity check.

Three structured tasks (PDF-to-calendar, financial-ratio, todo-cleanup) resist nearly all models, suggesting tooling or task-design issues rather than capability gaps. These should be revisited before drawing firm conclusions about any model's structured-output ceiling.

The picks below apply to the benchmark setup used throughout this report: the OpenClaw agent harness driving each model through the 21-task suite, with a local `llama3.1:70b` model as the LLM judge for open-ended tasks.

**Best all-around model (writing + analysis + structured tasks):**
`ollama/gemma4:31b` (0.807) and `ollama/gemma4:12b` (0.802) are tied at the top. Prefer the 12B as the default (near-identical quality at lower hosting cost), and the 31B when peak writing quality justifies the extra memory.

**Best for document parsing and structured data extraction:**
`ollama/qwen3.6:latest`: uniquely passes PDF-to-calendar and leads on automated structured tasks.

**Best for pure writing and open-ended drafting:**
`ollama/gemma4:31b`, `ollama/glm-4.7-flash`, or `ollama/gemma4:12b`; all achieve judge means above 0.880.

**Best small model (≤9B):**
`ollama/llama3.1:8b` and `ollama/qwen3.5:9b` are close; llama3.1:8b edges out on financial reasoning, qwen3.5:9b on writing quality.

**Avoid for agentic admin work:**
`ollama/deepseek-r1:32b` (writes well but fails structured tool-use) and `ollama/phi4:14b` (non-functional after sanity).

**Not yet representative:**
`ollama/llama3.1:70b`, `ollama/gpt-oss:latest`, `ollama/qwen3:8b`, still need clean sequential re-runs (parallel run 0053).

## 9 Future work

Several directions follow naturally from this evaluation.

**Harness and judge sensitivity.** All results in this report use a single combination: the OpenClaw agent harness with a local `llama3.1:70b` judge. A grid experiment varying both dimensions (OpenClaw vs. Hermes harness, local `llama3.1:70b` vs. cloud `gpt-4.1-mini` judge) was started

but remains incomplete. The Hermes runs also use a different scoring taxonomy (PRODUCTIVITY/WRITING/ANALYSIS vs. our Auto-9/Hybrid/Judge-8), making direct score comparison unreliable. Completing the grid with a unified scoring layer would isolate whether ranking shifts come from the harness, the judge, or task categorisation. Prior work on LLM-as-judge reliability [7, 8] shows that judge choice can shift absolute scores by up to 5 points, so this is not a theoretical concern.

**Expanding model and task coverage.** The 13 models tested here represent a snapshot of what was available in Ollama as of June 2026. New models are released monthly. The benchmark suite is designed to be extensible: adding a model requires only an Ollama tag and a single command. Priority candidates for inclusion are Llama 4, Mistral Large, and the next Gemma and Qwen generations. On the task side, the admin suite could be extended to cover code generation, data analysis in spreadsheets, and multi-turn conversation management, which would test agent coherence over longer horizons than the current single-turn design.

**On-device fine-tuning.** The DGX Spark's 128 GB unified memory enables full-parameter fine-tuning of 8B models and LoRA adaptation of 70B models locally [14]. A natural next step is to fine-tune a mid-range model (e.g. `gemma4:12b`) on the task rubrics and agent tool-call formats used in this benchmark, then re-evaluate. If fine-tuning closes the gap between the 12B and 31B variants on structured tasks, the case for the 12B as a production default strengthens further.

**Reproducibility on alternative hardware.** The benchmark suite runs on any Ollama-compatible host. Running it on alternative edge platforms, such as Apple Silicon Mac Studio (128–192 GB unified memory) or multi-GPU workstations, would test whether the model rankings generalise or are specific to the Grace Blackwell architecture. Signal65 [11] and StorageReview [12] report throughput numbers on Spark for common models; extending those comparisons to task-level quality would help teams choose hardware with confidence.

**Agent reliability over long chains.** This benchmark measures single-task performance with a 120-second timeout. Real agent deployments involve multi-step workflows with tool chaining, context accumulation, and error recovery. Exxact's work [13] on multi-hop agent benchmarks on Spark suggests that model reliability degrades non-linearly with chain depth. Extending the admin suite to include multi-task workflows (e.g. "read this email, extract action items, create calendar events, draft replies") would test whether the models that score well on individual tasks also hold up over longer agent loops.

## References

[1] A. Laskaridis et al. MELTing point: Mobile evaluation of language transformers. *arXiv preprint arXiv:2403.12844*, 2024.

[2] S. Murthy et al. MobileAIBench: Benchmarking LLMs and LMMs for on-device use cases. *arXiv preprint arXiv:2406.10290*, 2024.

[3] S. Liu, K. Han et al. Edge-LLM: Edge-device large language model competition. *NeurIPS 2024 Competition Track*, 2024.

[4] X. Liu et al. AgentBench: Evaluating LLMs as agents. *ICLR*, 2024.

[5] F. Patel et al. Berkeley function-calling leaderboard. *gorilla.cs.berkeley.edu/bfcl*, 2024.

[6] J. Jimenez et al. SWE-bench: Can language models resolve real-world GitHub issues? *ICLR*, 2024.

[7] L. Zheng et al. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. *NeurIPS*, 2023.

[8] D. Verga et al. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

[9] Red Hat Developer. Ollama vs. vLLM: A deep dive into performance benchmarking. *developers.redhat.com*, August 2025.

[10] aidatatools. Ollama benchmark: LLM throughput via Ollama. *github.com/aidatatools/ollama-benchmark*, 2025.

[11] R. Shrout. NVIDIA DGX Spark first look: A personal AI supercomputer on your desk. *Signal65 Research*, 2025.

[12] StorageReview. NVIDIA DGX Spark review: The AI appliance bringing datacenter capabilities to desktops. *storagereview.com*, 2025.

[13] Exxact Corporation. Benchmarking local AI agents on NVIDIA DGX Spark (Ollama + OpenClaw). *exxactcorp.com/blog*, May 2026.

[14] 2389 Research. Week 0: NVIDIA DGX Spark experiments. *2389.ai/posts*, 2025.

## Acknowledgements

We thank Skaylink and especially Ervin Rizgelis for loaning us the DGX Spark. Having the device for several weeks gave us enough time to run and re-run the full benchmark suite under controlled conditions. This work would not have been possible without their support.