KH Labs technical report · KHL-TR-2026-03 · November 2026

The LLM bill: one support workload, priced across models, and what actually cuts cost

We re-priced one support-bot workload, token for token, under different models and six cost levers. Most of the savings came from sending fewer tokens and fewer reasoning tokens, not from a cheaper price list.

KH Labs Data: runs of 11-12 Nov 2026 Prices checked: 2026-11-11 / 2026-11-12

Executive summary

- On a synthetic support workload (81 requests, scaled to 1.2M a month; each call at the list price of the model that served it: Claude Sonnet 5.5 or Gemini 3.1 Flash-Lite), six engineering levers took the modelled bill from **$23,117** to **$531** a month (97.7% lower), while the eval moved from 78/81 to 76/81. These are the live Cost Cutter demo's numbers.
- Trimming prompt and RAG context was the biggest single lever (−$13,410), then capping output and reasoning (−$5,000) and routing easy calls to a small model (−$3,488). Caching, dedupe and batch add −$639, −$26 and −$23.
- Cost per *resolved* support question fell from $0.0158 to $0.00086: price per correct answer, not per call, is the number to manage.
- Reasoning tokens drive the spread. Measured on the same 20 support prompts, Gemini 3.8 Flash (high reasoning) wrote 519 output tokens per answer vs 63 for Gemini 3.1 Flash-Lite.
- Prompt caching only pays at high hit rates: at 0% hits the 1.25× cache-write premium makes it cost more (−$1,844/month on the baseline), at 95% it saves $6,564.

1. Method

Workload. A fictional SaaS support assistant ("Corvid Cloud support AI", the same workload as the Cost Cutter demo): 37 support questions answered with RAG over a 32-chunk knowledge base (8 are repeats or rephrasings), 24 ticket classifications with gold labels, 12 thread summaries and 8 nightly account digests: 81 requests in total. Monthly figures assume the mix repeats at 1,200,000 requests a month; that volume is an assumption, the per-request token counts are measured.

Runs. Each lever is a real pipeline run through one OpenAI-compatible API gateway (big model `claude-sonnet-5-5-high`, small model `gemini-3.1-flash-lite`), temperature 0, on 11 Nov 2026. Token counts come from the API `usage` field of every call, so hidden reasoning tokens are included the way they are billed. Caching, dedupe and batch are applied to measured tokens: caching with an assumed 95% hit rate on stable prefixes of at least 1,024 tokens, dedupe with an exact plus normalised-text cache replay (3 exact and 5 normalised hits in 81), batch at 50% for the nightly digest only.

Prices. Official list prices read from the providers' pricing pages (Anthropic, OpenAI, Google) on 2026-11-11. As in the Cost Cutter demo, the headline prices each call at the list price of the model that served it: Claude Sonnet 5.5 or Gemini 3.1 Flash-Lite. Table 2 re-prices the same tokens on each provider's "big" and "small" tier; that tier mapping is our assumption. The 16-model market table uses prices re-checked on 2026-11-12 (sources per row in the CSV) and the third-party Artificial Analysis Intelligence Index as a coarse quality signal.

Quality guard. Exact label match for classification; an LLM judge (`gemini-4-argon-high`, written rubric: correct, complete, policy-safe, full KB in context) for everything else. A lever only counts if the final pipeline still passes the eval; every miss is listed in the source data.

Model sweep. Separately, we ran 20 support prompts (and four other task types, 200 prompts per model) on 7 Gemini models on 11 Nov 2026, judged by `gemini-4-carbon-high`; a second judge re-scored 20 responses with 98.4% criterion agreement.

2. Results

2.1 Six levers, one bill

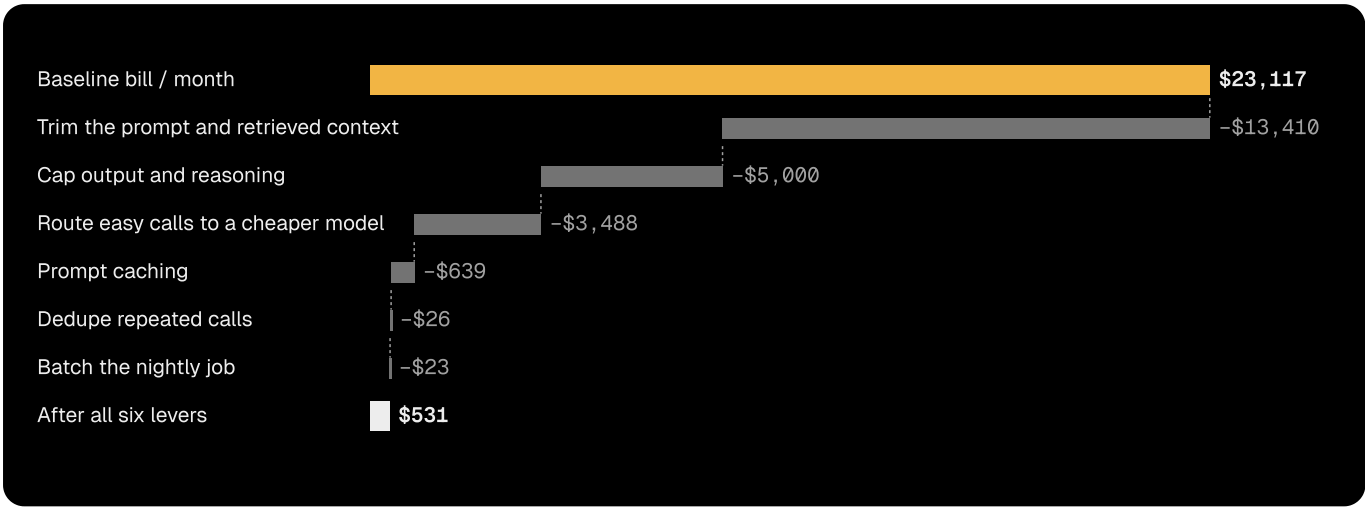


Figure 1. Sequential savings on the modelled monthly bill, each call at the list price of the model that served it: Claude Sonnet 5.5 or Gemini 3.1 Flash-Lite. Each bar is the saving of that lever applied after the ones above it; the six bars sum exactly to the difference, $23,117 − $531. Dedupe (−$26) and batch (−$23) read small because earlier levers already removed most of the cost.

Lever	Bill before	Bill after	Saving
Trim the prompt and retrieved context	$23,117	$9,707	**−$13,410**
Cap output and reasoning	$9,707	$4,707	**−$5,000**
Route easy calls to a cheaper model	$4,707	$1,220	**−$3,488**
Prompt caching	$1,220	$580	**−$639**
Dedupe repeated calls	$580	$554	**−$26**
Batch the nightly job	$554	$531	**−$23**

Table 1. The waterfall in numbers, as run; the same figures as the live Cost Cutter demo.

Price list	Big / small tier	Baseline	After routing	After all six
As run (headline)	Claude Sonnet 5.5 / gemini-3.1-flash-lite	$23,117	$1,220	$531
Anthropic list	Claude Sonnet 5.5 / Claude Haiku 5.5	$23,117	$1,018	$433
OpenAI list	gpt-6.1-sol / gpt-6-luna	$23,117	$1,018	$425
Google list	gemini-4-carbon / gemini-3.1-flash-lite	$23,117	$1,220	$524

Table 2. Re-pricing: the same measured tokens priced on each provider's list, for comparison only; the headline is the as-run row. The big-tier models share a $2/$10 list price per million input/output tokens, so the baseline and the first two levers are identical on every list; the lists differ from routing on, where the small tiers' prices differ.

2.2 Quality held: 78/81 → 76/81

Task	Requests	Baseline pass	Final pass	Cost/month before	Cost/month after
Support questions	37	36/37	35/37	$8,420	$449
Ticket classification	24	23/24	23/24	$4,582	$26
Thread summaries	12	12/12	10/12	$3,107	$34
Nightly digests	8	7/8	8/8	$7,007	$23

Table 3. Eval by task. Overall 78/81 → 76/81. Two of the 5 final misses are also baseline misses; the other three are completeness or fidelity misses the baseline passed. Cost per resolved support question: $0.0158 → $0.00086.

2.3 The same support prompt on seven models

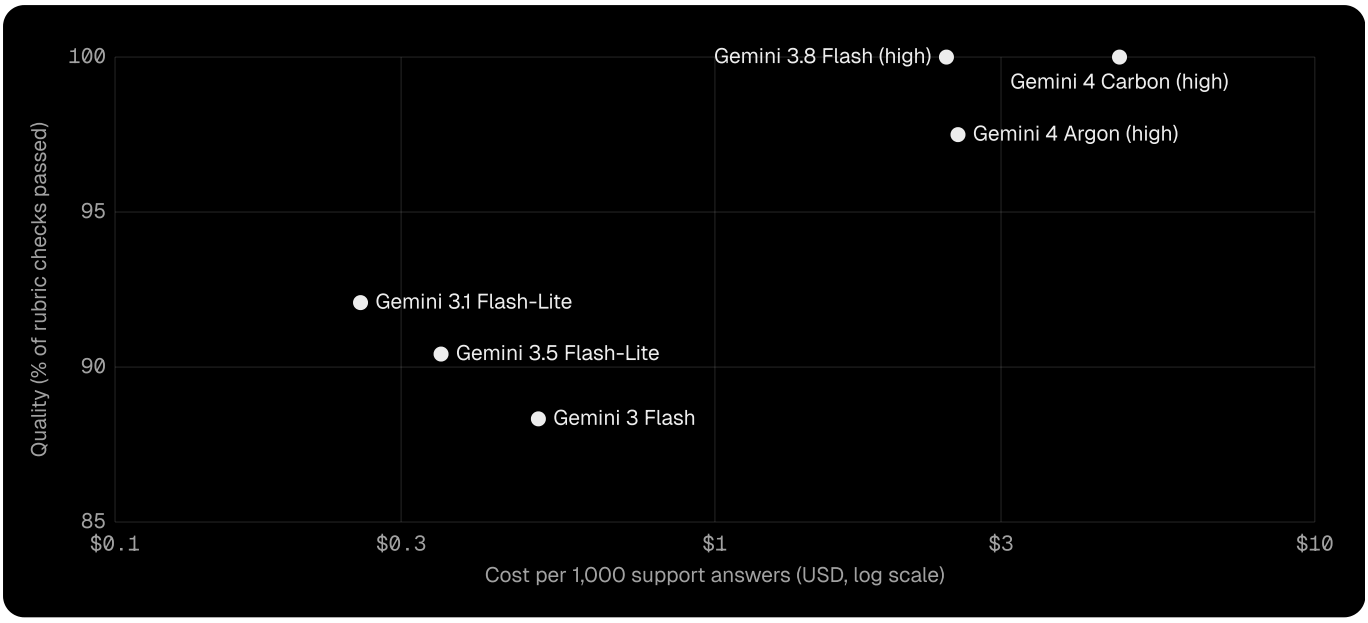


Figure 2. Measured cost per 1,000 support answers vs rubric quality for six of the seven models, 20 prompts per model, list prices. The seventh, Gemini 3.8 Flash (tiered), is a variant of 3.8 Flash and is in Table 4 only. Higher-reasoning models are not on a different price list so much as they write more tokens.

Model	Quality	Output tokens / answer	Cost / 1,000 answers	Cost / 1,000 resolved
Gemini 3.1 Flash-Lite	92.1%	63	$0.257	$0.279
Gemini 3.5 Flash-Lite	90.4%	62	$0.350	$0.387
Gemini 3 Flash	88.3%	61	$0.508	$0.575
Gemini 3.8 Flash (high)	100.0%	519	$2.432	$2.432
Gemini 3.8 Flash (tiered)	97.5%	531	$2.479	$2.543
Gemini 4 Argon (high)	97.5%	118	$2.542	$2.607
Gemini 4 Carbon (high)	100.0%	336	$4.727	$4.727

Table 4. Support task, 7 measured models. "Resolved" divides cost by the pass rate. With 20 items per model, differences of a few points are within noise.

2.4 Sensitivity: caching and routing

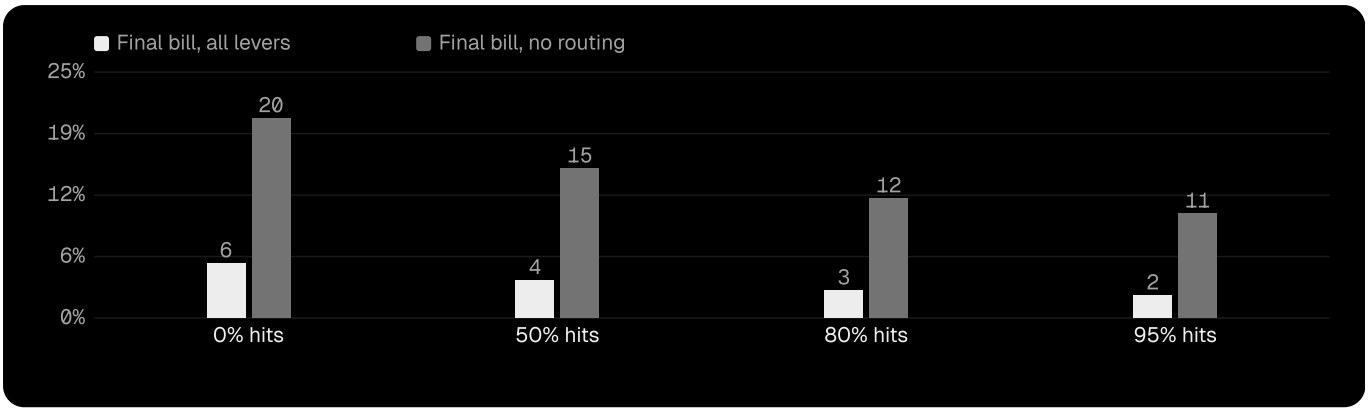


Figure 3. Final monthly bill as a share of the $23,117 baseline, by prompt-cache hit rate, with all six levers (white) and without model routing (grey), at as-run prices. Routing matters more than the cache hit rate once prompts are trimmed.

Cache hit rate	Caching alone on baseline	Final bill (all levers)	Final bill without routing
0%	-$1,844 (costs more)	$1,289	$4,696
50%	$2,581	$890	$3,518
80%	$5,236	$651	$2,810
95%	$6,564	$531	$2,457

Table 5. “Caching alone” is the monthly saving from caching on the unoptimised baseline. At a 0% hit rate it costs money because of the 1.25× cache-write price of the big model; OpenAI and Google pages list no write premium, so there it is ~$0.

2.5 The market, priced on one support chat

Model	Provider	AA index	100k chats / month
GPT-6 Luna	OpenAI	21	$10
Claude Haiku 5.5	Anthropic	30	$10
DeepSeek V4.1 Flash 1017	DeepSeek	29	$27
Gemini 3.5 Flash-Lite	Google	12	$35
Qwen3.8 27B (hosted)	Qwen (Alibaba Cloud)	20	$51
Gemini 3.8 Flash	Google	31	$72
GPT-6.1 Terra	OpenAI	42	$96
DeepSeek V4 Pro 0813	DeepSeek	23	$111
Qwen3.8 Max	Qwen (Alibaba Cloud)	34	$168
GPT-6.1 Sol	OpenAI	50	$193
Claude Sonnet 5.5	Anthropic	42	$193
Gemini 4 Carbon	Google	60	$193
Gemini 4 Argon	Google	50	$193
Claude Opus 5.5	Anthropic	52	$386
Claude Fable 5.2	Anthropic	62	$772
GPT-6 Astra	OpenAI	55	$965

Table 6. 100,000 support chats a month at 650 input and 63 output tokens each (our measured support-prompt size on a non-reasoning model), list prices checked 2026-11-12. The AA index is a third-party general score, not our eval; the cheapest row is not a like-for-like swap. Reasoning models will write more output tokens than 63, so their real cost is higher than this row shows (see Table 4).

3. Scope

One fictional workload, one run per stage at temperature 0, one LLM judge per study; monthly volume is assumed, per-request tokens are measured. Caching assumes a 95% hit rate and ignores cache TTL and storage fees. Prices are list prices on the stated dates and change (Google lists a Gemini 4 Carbon price rise on 2026-12-01). The baseline is a deliberately unoptimised but common shape (big reasoning model on every call, long prompt, 10 retrieved chunks, uncapped output), so a lean stack will see smaller percentages; the order of the levers is the transferable result.

4. Recommendations

- Measure cost per *resolved* request, from your own `usage` logs, before changing models.
- Cut tokens first: trim the system prompt (keep every rule), rerank to 3 chunks, cap output and set a low reasoning budget where the eval allows.
- Route classification, summaries and routine questions to a small model, with an escalation rule back to the big one; gate every change on a fixed eval set.
- Turn on prompt caching only for long stable prefixes with a high real hit rate; check the provider's cache-write price.
- Batch anything not user-facing (nightly digests, back-office summaries).

Appendix

Data. r3_waterfall.csv (as-run prices and 3 provider price lists × 6 levers), r3_measured_models.csv (7 models × 10 tasks), r3_market_prices.csv (16 models, source URL per row).

Code. The Cost Cutter engine (30 tests), the LLM Bill Benchmark runner and the sensitivity re-pricing script are not published; we share them on request. Live demos: Cost Cutter, LLM Bill Benchmark.

Cite this report

KH Labs (2026). The LLM bill: one support workload, priced across models, and what actually cuts cost. KH Labs Technical Report KHL-TR-2026-03. https://khlabs-research.pages.dev/llm-bill/