BV-SALA Savings Layer: Verifiable Measurement of Cost Savings in Large Language Model Applications

Joan-Marc Fisa, Founder @Bivelio
jmfisa@bivelio.com
BiVelio Inc - www.bivelio.com/research
savings.bivelio.com*

August 2026

Abstract

Cost claims for large language model (LLM) optimizations require explicit workloads, uncertainty, and a documented cost basis. This paper presents BV-SALA Savings Layer, an in-process SDK and local gateway that combine provider-reported usage with versioned list prices and separate four savings ledgers. Live SDK double-runs used a percentile cluster bootstrap with 10,000 resamples. On GSM8K-derived traffic, gpt-4o-mini showed a 31.5% cost reduction (95% CI: 25.7–36.7; 100 families, 200 requests per arm) and claude-sonnet-5 showed 75.1% (95% CI: 69.3–79.6; 31 families, 62 requests per arm). The Anthropic arm included 41 percentage points from a prefix-cache effect enabled by cache_control in the instrumented condition, not a pre-existing provider discount. A separate counterfactual replay of 19,779 gateway requests estimated 2.49% (95% CI: 1.50–3.57); it was not a live double-run or invoice reconciliation. We distinguish measured, reference-valued, and estimated amounts, state quality and latency observations without claiming non-inferiority, and present only provisional economic extrapolations.

Keywords—LLM; inference cost; provider-reported usage; prefix caching; cluster bootstrap; prompt optimization.

1 Introduction

The mass adoption of large language models (LLMs) has added a volatile item to engineering budgets: inference spending. Cost depends on request design, conversational context, provider caching, and repeated work. Prior research addresses prompt compression [1–3], attention and prefix reuse [4–9], response caching [10–12], model routing [13, 14], and efficient serving [6, 15, 16]. For an organization evaluating these levers, a further problem remains: connecting a reported saving to a defined workload, an uncertainty statement, and an auditable cost basis. Provider-reported usage and versioned list prices provide documentary evidence for that calculation; they are not, by themselves, a reconciled invoice or metrological traceability.

An isolated percentage can combine calls that were never executed, removed or compressed tokens, and cache effects whose attribution differs across experimental arms. These mechanisms have different workload dependencies and should not be summed into an unconditional claim. Interpretable reporting therefore requires the model, traffic construction, clustering unit, interval procedure, and distinction between live measurement and counterfactual estimation.

This paper presents BV-SALA Savings Layer, developed by BiVelio, and evaluates a limited set of public-bench experiments. Its bookkeeping principle is to expose the evidence behind each amount rather than to claim that every amount comes from an identical instrument. The contributions are:

- **Non-intrusive architecture (Section III):** an in-process Node.js SDK and a local gateway for coding agents, with provider traffic kept on the configured application path and prompt content excluded from telemetry by design.
- **Separate-ledger accounting (Section IV):** avoided calls, eliminated tokens, compressed tokens, and provider-cached tokens are reported separately, with measured, reference-valued, and estimated cost bases.
- **Cluster-aware uncertainty and explicit evidence types (Sections IV–V):** live SDK double-runs are analyzed by percentile cluster bootstrap, while the gateway result is labeled as a counterfactual replay with a partially reconstructed baseline.
- **Reproducible reporting and bounded economics (Sections V–VI):** workloads, seeds, observed quality and latency, limitations, and a break-even expression using a total monthly fee are reported without presenting extrapolation as validated generalization.

All numerical claims below are restricted to the reported experiments and artifacts available through savings.bivelio.com. No invoice reconciliation was performed for these experiments. BiVelio Inc - www.bivelio.com/research develops the product evaluated here; Sections VII and the Conflict of Interest statement address the resulting risks.

1.1 Research Questions

RQ1. What cost reductions are observed in the specified live SDK double-runs when provider-reported usage is valued at versioned list prices?

RQ2. How much uncertainty remains when dependence is handled by family- or session-level cluster bootstrap?

RQ3. Which components can be attributed to avoided calls versus cache behavior enabled by the instrumented arm?

RQ4. What quality and latency observations are available, and which claims remain unsupported?

RQ5. How should a counterfactual gateway replay and a provisional break-even calculation be interpreted relative to live evidence?

2 Related Work

Token pricing is a commercial decision made by providers, not a property of the Transformer architecture. Against that pricing background, one optimization front reuses computation. Prompt Cache [4], CachedAttention [5], SGLang [6], and CacheBlend [7] study attention or KV-cache reuse under different serving assumptions. Anthropic and OpenAI also document provider-side prompt caching [8, 9]. These mechanisms can lower usage charges, but an experiment must distinguish caching already present in both arms from caching enabled by the intervention.

A second front reduces prompt tokens: LLMLingua [1], LLMLingua-2 [2], and gist tokens [3] use distinct compression strategies. A third reuses responses: GPTCache [10], SCALM [11], MeanCache [12], and vCache [24] address freshness, privacy, and per-prompt error bounds; P-CHR AUC and Operational Retention Rate evaluate deployment [27]. Routing systems such as FrugalGPT [13], RouteLLM [14], and RouteNLP [25] select among models and calibrate escalation. vLLM [15], SGLang [6], and speculative decoding [16] improve serving efficiency. OpenTelemetry's GenAI conventions [17] provide an observability vocabulary, not cost attribution.

BV-SALA combines established mechanisms with request-identity guards, separate ledgers, calibrated abstention, and economic gating. Its differentiation is the integration of accounting and safety rather than each cache, serialization, or observability mechanism in isolation. vCache is the closest comparator for semantic caching, but the present work does not establish equivalent guarantees, model coverage, or hit rates. Current model routing uses economic and structural checks rather than a learned closed loop, and its semantic-quality evidence remains absent. Structured-data rewriting also requires task- and model-specific quality evaluation: Notation Matters shows that token reductions can reduce tool-calling accuracy and amplify multi-turn parsing failures [26]. The uncertainty discussion uses established terminology [18,19] without asserting a calibration chain or strict metrological traceability. The evidence therefore supports a measurement-and-guardrails contribution, not a claim of general optimization leadership.

3 System Architecture

BV-SALA has three components (Figure 1). The SDK and gateway record provider-reported usage and derive costs using versioned list prices. Telemetry includes token counts and costs plus permitted operational metadata, such as model, timestamps, opaque identifiers, and software version; prompt content is excluded by design. Upstream endpoints and telemetry settings are configurable, so the architecture is not a universal privacy guarantee for every deployment.

3.1 SDK for Node.js Applications

The SDK (`@bivelio/savings-layer`) targets JavaScript and TypeScript applications that call an LLM API with their own key. It is an ESM package with zero runtime dependencies; the exact tokenizer is an optional peer dependency. Wrapping the provider with `createSavingsLayer` returns a `SavingsReport` alongside `generate()` or `generateStream()` responses. In streaming mode, final provider-reported usage is read when the stream closes; if it is absent, the request is unmeasured and cannot support a measured amount. Native adapters cover Anthropic and Gemini, and OpenAI-compatible endpoints can be used where their response contracts expose the required usage fields.

3.2 Gateway for Coding Agents

The gateway targets coding agents, including Claude Code [20], Codex CLI, Aider, and Goose. It records requests with provider-reported usage and applies versioned list prices. On supported Anthropic configurations, the cache-policy lever changes cache writes from one-hour to five-minute TTL only when at least 30% of prompt bytes are new, leaving other turns untouched. This threshold concerns bytes, not a qualitative judgment that most of a prompt is new. The policy changes cache behavior rather than prompt content, but its net effect must be evaluated because misses can offset savings. Other SDKs can use the gateway as a base URL when compatible with the configured upstream.

3.3 Savings Skill for Subscriptions

For teams that consume LLMs through subscription plans with usage windows, there is a third, free piece: `savings-mode`, a skill of pure instructions (no proxy, no interceptor) that the agent installs in `~/.claude/skills` and that reduces the two masses that deplete the quota fastest: the model's output and tool results. Its effect on subscription limits is pending a published A/B measurement and, consequently, is not associated here with any figure.

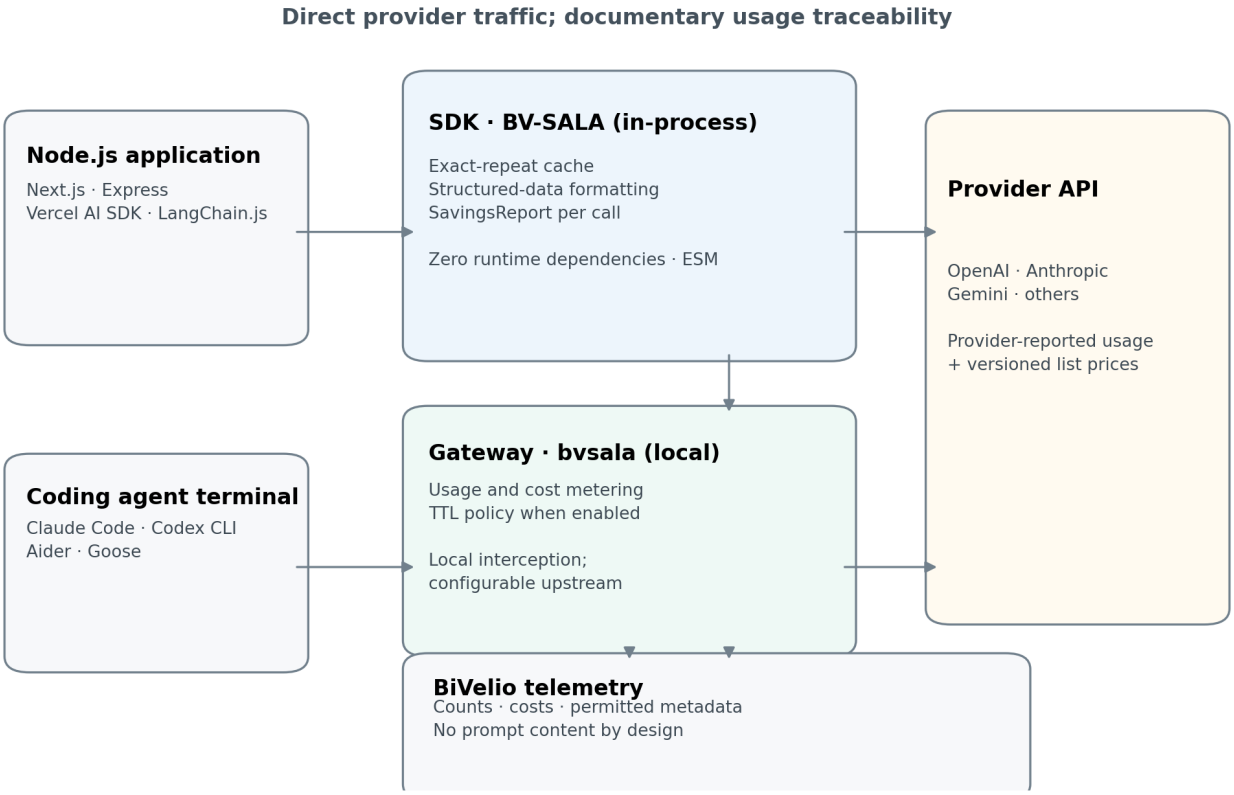


Figure 1. Deployment architecture. The SDK operates in process and the gateway locally on the configured provider path. Costs are computed from provider-reported usage and versioned list prices. Telemetry comprises counts, costs, and permitted metadata rather than prompt content by design; deployment configuration still requires review.

3.4 Privacy Model

The default design prevents prompt content from being sent to BiVelio telemetry: processing occurs in process or locally, while telemetry carries counts, costs, and permitted metadata such as model names, timestamps, opaque request or session identifiers, and software versions. Metadata can still be operationally sensitive. Because upstream providers and deployment settings are configurable, the paper does not claim a universal guarantee that content can never leave the client environment; deployments must verify endpoints, retention, access controls, and telemetry configuration.

4 Measurement Methodology

4.1 Live Double-Runs and Counterfactual Replay

The live SDK experiments compare a base condition A without the layer and an instrumented condition B on the same model and constructed content. For scenario S, such as GSM8K-derived traffic [21], C_{base} and C_{opt} are costs computed from provider-reported usage and the versioned list prices in force for the analysis. The primary metric is relative cost reduction in equation (1). This live protocol is distinct from the gateway analysis in Section V.B, which replays logged requests counterfactually and partially reconstructs its baseline; not every result in the paper is a live double-run.

$$R = \frac{C_{\text{base}} - C_{\text{opt}}}{C_{\text{base}}}. \tag{1}$$

Equation (1) is exposed per request as `report.cost.netSavingsRatio` where the required fields are available. Provider-reported usage documents the quantities entering the calculation, and versioned list prices document their valuation. This is documentary traceability to usage and price records, not strict metrological traceability and not proof of the amount on an invoice. Differences between list-price calculations and invoices—such as negotiated prices, credits, taxes, rounding, or later provider adjustments—remain outside the present evidence.

4.2 Cluster Bootstrap Uncertainty

The live SDK intervals use a non-parametric percentile cluster bootstrap with 10,000 resamples [22, 23]. Requests within a family are retained together; the family, not the individual request, is the resampling unit. The OpenAI and Anthropic flow-construction seed is 20260928 and the bootstrap seed is `0xa11`. The gateway replay instead resamples whole sessions with seed 20261003. For each resample b, equation (1) is recomputed, and the 95% interval is given by the 2.5th and 97.5th empirical percentiles in equation (2), not by a normal approximation.

Requests are therefore not treated as independent observations.

$$\mathrm{CI}_{95} = \left[Q_{0.025}\left(\{R^{(b)}\}\right), Q_{0.975}\left(\{R^{(b)}\}\right) \right]. \tag{2}$$

The interval summarizes resampling variability under the observed cluster design. It does not capture all temporal drift, model changes, price changes, or counterfactual reconstruction error. Small numbers of families or sessions can also make percentile intervals unstable; the sample sizes accompany every result.

4.3 The Four Ledgers

`SavingsReport` separates four ledgers (Table 1). Their valuation can be measured, reference-valued, or estimated. In particular, an avoided call has no new provider usage reading. Under stated equivalence conditions it may be valued by reference to a prior identical executed request; that is a reference value, not an identical counter reading for the avoided call. Other counterfactuals are estimates. The categories remain visible per request and in aggregate rather than being collapsed into an unconditional measured total.

The separation serves two purposes. First, cache-related usage reductions are not automatically attributed to the layer: attribution depends on the experimental arms. Second, each ledger answers a different engineering question and depends differently on traffic, prompt structure, and provider behavior.

4.4 Measured, Reference-Valued, and Estimated Amounts

An amount is measured when directly supported by provider-reported usage for that execution and a documented price version. It is reference-valued when a counterfactual amount uses an earlier equivalent or identical request under explicit matching conditions. It is estimated when reconstructed from logs, list-price calculations, or other assumptions without such a direct reference. A stream without final usage is unmeasured. These labels concern evidentiary basis, not mutually exclusive kinds of optimization, and do not convert reference values or estimates into invoice-verified savings.

4.5 Public Bench of Trials

Public-bench reports identify model, SDK version, traffic construction, condition costs, ledgers, cluster unit, seeds, and interval. The September 28, 2026 SDK flow used exact-repeat caching; on Anthropic, the instrumented arm also introduced `cache_control`. The published package contains no general prompt compressor, so the compression ledger applies only when structured data are supplied in the supported form. The gateway replay is published as a different evidence type and is not pooled with the live SDK experiments.

5 Public Bench Results

5.1 Live SDK Double-Runs

Table 2 and Figure 2 separate the two live SDK double-runs from the gateway counterfactual replay. All amounts use provider-reported usage where available and versioned list prices; none is presented as an invoice-reconciled result.

In the OpenAI experiment, each arm evaluated 200 requests from 100 GSM8K families. The constructed stream contained 30% exact repeats and 20% paraphrases. The gpt-4o-mini cost reduction was 31.5% (95% CI: 25.7–36.7), using SDK 0.4.31. The ratio $r = 0.315/0.30 = 1.05$ describes this single operating point. It is not an empirical slope, a regression estimate, or evidence that savings scale linearly at other repeat rates.

The Anthropic experiment used a reduced flow of 31 families and 62 requests, not the same stream as OpenAI. The claude-sonnet-5 reduction was 75.1% (95% CI: 69.3–79.6). Baseline cache-read usage was zero, whereas instrumented-arm cache-read usage was 72,447 tokens because the layer introduced `cache_control`. Thus approximately 41 percentage points represent a prefix-cache effect enabled in the instrumented arm, not a provider discount pre-existing in both conditions. The remaining reduction is associated principally with avoided calls, but this two-arm design cannot cleanly separate all interactions between deduplication and caching.

5.2 Counterfactual Gateway Replay

The gateway result is a counterfactual replay of 19,779 requests from 51 Claude Code sessions over 13 days in one organization. At versioned list prices, changing eligible cache writes from one-hour to five-minute TTL yielded an estimated net reduction of 2.49% (95% CI: 1.50–3.57), with whole-session resampling and seed 20261003. The baseline was partially reconstructed from logs rather than produced by a contemporaneous live double-run. Consequently, this figure is not invoice-verified and should not be interpreted as equivalent to the SDK evidence.

5.3 Observed Quality, Latency, and Non-Breakage Checks

For the Sonnet flow, observed accuracy was 100% on 56 of 56 scored answers in each arm. This does not establish quality equivalence or non-inferiority. Median provider-call latency was 1,969 ms in A and 2,225 ms in B; these descriptive medians do not characterize the full latency distribution or isolate every source of overhead.

Gemini non-breakage checks on October 1–2, 2026 produced no publishable savings interval. They are retained as operational checks, not as evidence of savings or as a substitute for a powered comparative experiment. Semantic-cache evidence is also bounded: on SemBenchmarkLmArena, $\delta = 15\%$ certified while $\delta \leq 12\%$ did not; calibration required 13,121 labels, and the live GSM8K calibrated arm produced zero semantic hits. These outcomes demonstrate abstention and calibration behavior, not broad semantic-cache savings [28].

Table 1. The four ledgers. They remain separate to reduce double counting; the cost basis of each amount must additionally be identified as measured, reference-valued, or estimated.

Ledger	Content and valuation criterion
`avoided_calls`	Calls not executed; reference-valued from an equivalent executed request only under explicit matching conditions, otherwise estimated.
`eliminated_tokens`	Prompt information removed; measured only for directly observed usage and documented prices, otherwise classified by its actual evidence basis.
`compressed_tokens`	Information retained with fewer tokens; valued from observed usage where available, with counterfactual components labeled.
`provider_cached_tokens`	Provider cache usage reported separately; attribution depends on whether caching was pre-existing or enabled by the instrumented arm.

Table 2. Cost reductions and 95% percentile cluster-bootstrap intervals. The first two rows are live SDK double-runs (family-level resampling); the gateway row is a counterfactual replay (session-level resampling) with a partially reconstructed baseline.

Evidence, system, and model	Reduction	95% CI	Workload and basis
Live SDK A/B · gpt-4o-mini	31.5%	[25.7; 36.7]	100 families / 200 requests; 30% exact repeats + 20% paraphrases; SDK 0.4.31
Live SDK A/B · claude-sonnet-5	75.1%	[69.3; 79.6]	Reduced flow: 31 families / 62 requests; `cache_control` enabled in B
Counterfactual gateway replay · Anthropic TTL	2.49%	[1.50; 3.57]	19,779 requests / 51 sessions / 13 days; partially reconstructed baseline; list prices

6 Economic Analysis and Scenarios

6.1 Total-Cost Model

Commercial prices change and are therefore not reproduced as a product table. Let F denote the total monthly fee for the deployment, across all applicable seats or identities. Break-even compares F with provisional savings under explicitly stated assumptions. A complete adoption decision must also include any usage-based commission, integration and operating costs, taxes, and other total-cost-of-ownership (TCO) components; these are not estimated here.

6.2 Break-Even Threshold

Let B be the monthly provider spend traversing the layer at the same list-price basis, ρ the exact-repeat fraction, and r the ratio observed at the single OpenAI operating point. For a provisional avoided-calls extrapolation $S(\rho) = B \cdot r \cdot \rho$, break-even requires $F < S(\rho)$, giving the threshold in equation (3). This expression is conditional arithmetic, not a validated demand or response curve.

$$\rho^* = \frac{F}{B \cdot r}. \tag{3}$$

Here $r = 0.315/0.30 = 1.05$ is a ratio at one observed point, not a fitted slope. No confidence interval for a slope is used. For example, if $B \cdot r/F = 20$, the provisional threshold is $\rho^* = 5\%$; if $B \cdot r/F = 5$, it is 20%. These values merely illustrate equation (3). Figure 3 presents the same dimensionless arithmetic without prices or an empirical uncertainty band.

6.3 Scope of Extrapolation

Equation (3) is useful only as a provisional screen before measurement on an organization's own traffic. Exact repeats may be rare or highly clustered, and paraphrases are not assumed to produce the same benefit. Cache effects, eliminated tokens, compressed tokens, and commissions are excluded from this simple expression. Generalization to other providers, models, repeat rates, or production workloads is not established; shadow-mode observation and a preregistered repeat-rate sweep are more defensible next steps.

7 Discussion and Threats to Validity

7.1 Internal Validity

Contemporaneous SDK double-runs reduce, but do not eliminate, differences in traffic and provider state. Family-level resampling respects dependence within constructed families; it does not model every provider-side change. The Anthropic comparison is additionally vulnerable to attribution ambiguity because `cache_control` was introduced only in the instrumented arm. A factorial design is required to separate deduplication, cache enablement, and their interaction. The gateway replay carries further reconstruction risk because its baseline was not a contemporaneous live arm.

7.2 External Validity

The SDK samples are small: 100 OpenAI families and 31 Anthropic families, with constructed repetition. The gateway covers 51 sessions, 13 days, and one organization. These samples do not establish temporal stability across days, provider

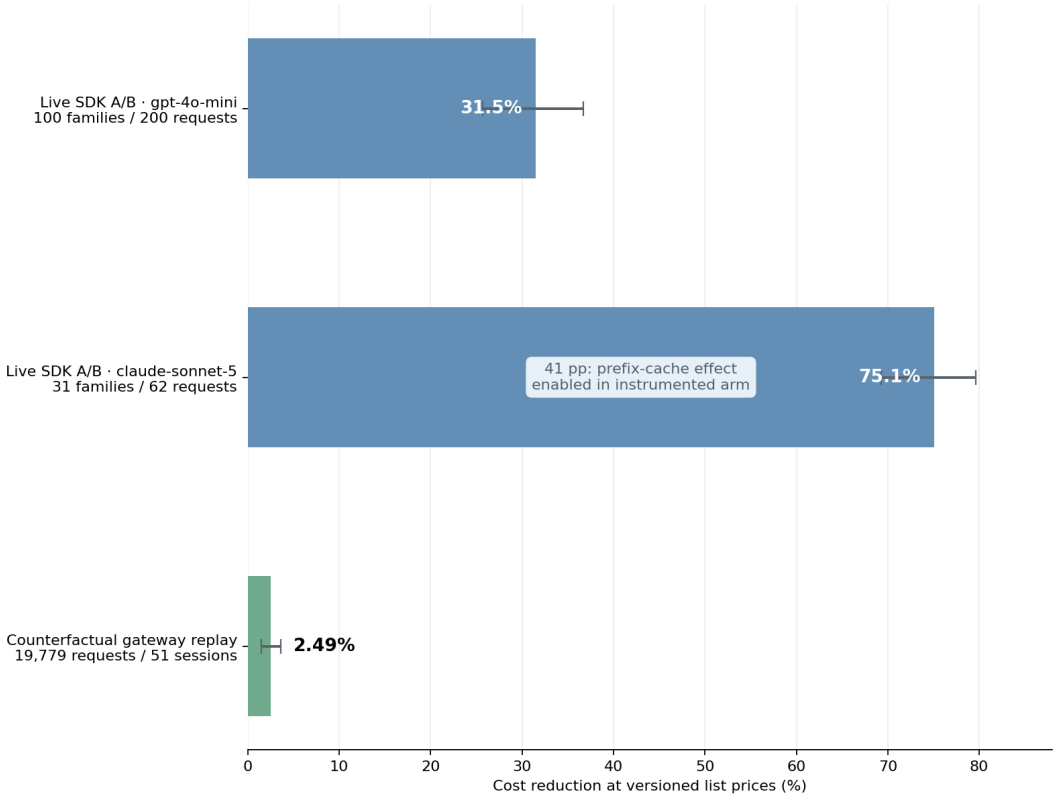


Figure 2. Cost reductions at versioned list prices with 95% cluster-bootstrap intervals. OpenAI and Anthropic are live SDK double-runs; the gateway result is a counterfactual replay, not a live double-run. In the Anthropic experiment, approximately 41 percentage points are a prefix-cache effect enabled by `cache_control` in the instrumented arm. Error bars show the reported intervals.

updates, price revisions, or production workload drift. Exact caching also changes answer freshness and the response distribution when an application would otherwise receive a new sample. The results therefore cannot be generalized to creative, time-sensitive, multi-tenant, or semantically similar traffic without new experiments.

7.3 Privacy and Operational Dependence

Local processing reduces prompt exposure to the measurement vendor, but counts, costs, model names, timestamps, opaque identifiers, and versions remain telemetry and require governance. Configurable upstreams prevent a universal content-flow guarantee. Operational dependence, agent coverage, provider response changes, and software supply-chain risk also remain. Measure mode limits response modification, but activated caching and TTL levers can change system behavior and must be evaluated in the deployment context.

7.4 Limitations

The study does not demonstrate quality non-inferiority. Sonnet accuracy was 100% on 56 of 56 scored answers in each arm, but this small, apparently easy subset does not establish equivalence; median provider-call latency was 1,969 ms in A and 2,225 ms in B, an observed difference rather than a complete latency analysis. Exact-repeat caching can reduce freshness and distributional diversity. Other limitations are the small samples, limited days and temporal drift, one gateway organization, lack of independent replication, private raw gateway logs, and the authors' developer role. The reference semantic store is in-memory and linear in records per lookup; multi-replica consistency and production-scale latency are unproven. Current routing lacks evidence of semantic correctness, and TOON is not evaluated as a shipped optimization. Reproduction depends on the specified software branch and immutable analysis artifacts; later code or mutable reports are not substitutes. List-price estimates are not invoice reconciliation, and ledger-specific amounts should not be summed without a defined estimand.

8 Conclusions and Future Work

This paper reports a bounded evaluation of BV-SALA using provider-reported usage, versioned list prices, separate ledgers, and cluster-aware intervals. Live SDK double-runs yielded 31.5% for gpt-4o-mini and 75.1% for claude-sonnet-5 under their respective constructed workloads. The latter includes a cache effect enabled in the instrumented arm. A distinct gateway counterfactual replay yielded 2.49% with a partially reconstructed baseline. These results support workload-specific cost calculations, not universal savings claims, strict metrolog-

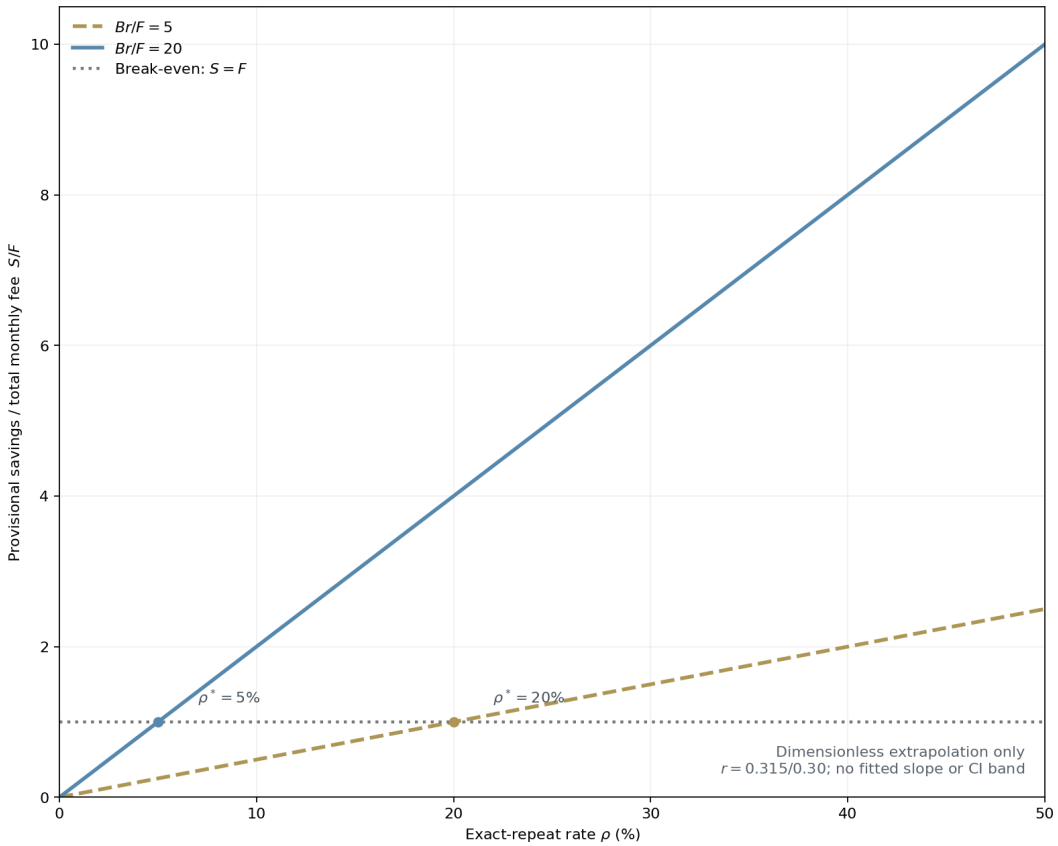


Figure 3. Provisional, dimensionless extrapolation only. $S/F = (B \cdot r/F)\rho$, with $r = 0.315/0.30$ from one OpenAI operating point. Lines for $B \cdot r/F = 5$ and 20 intersect $S = F$ at $\rho^* = 20\%$ and 5%, respectively. This is not a regression, a measured dose-response relationship, or a forecast for other workloads.

ical traceability, or invoice-level verification. The economic threshold is provisional arithmetic using r at one point.

Future work should use a factorial deduplication-by-`cache_control` design for attribution; sweep ρ to estimate rather than assume a response curve; run multi-day hierarchical experiments across families, sessions, and days; and preregister adequately powered non-inferiority tests for answer quality, freshness, and latency. External replication on independent organizations and providers, a current full-stack B2 comparison, invoice reconciliation including negotiated prices and credits, and release of reproducible resampling artifacts are also required. Semantic-cache evaluation should report precision across cache utilization and Operational Retention Rate, while persistent stores require transactional multi-replica coordination and latency budgets. Additional work should assess routing correctness, semantic-cache errors, subscription-skill effects, and interactions among ledgers.

9 Data and Code Availability

Public-bench summaries and the software distribution are available through `savings.bivelio.com`. The SDK results reported here use version 0.4.31 and the stated flow and bootstrap seeds. Replication must use the specified branch and immutable analysis artifacts. Raw gateway logs contain private operational data and are not public; this restricts independent verification of the replay and baseline reconstruction. No claim is made that all raw provider records or invoices are publicly available.

10 Conflict of Interest

Joan-Marc Fisa is the founder of BiVelio Inc - `www.bivelio.com/research`, which develops the product evaluated in this paper. No independent replication or third-party audit is reported. This relationship should be considered when interpreting the results.

11 Author Contributions

Author: Joan-Marc Fisa, Founder @Bivelio - jmfisa@bivelio.com. The research and development reported here were conducted under BiVelio Inc - `www.bivelio.com/research`.

A Conclusiones (resumen en español)

Este trabajo presenta una evaluación delimitada de BV-SALA basada en usage reportado por el proveedor y tarifas de lista versionadas, con trazabilidad documental y sin afirmar trazabilidad metrológica ni conciliación de facturas. Las ejecuciones live del SDK usaron bootstrap percentile por clúster con 10.000 remuestreos: 31,5% [25,7; 36,7] en OpenAI (100 familias, 200 solicitudes) y 75,1% [69,3; 79,6] en Anthropic (31 familias, 62 solicitudes). En Anthropic, unos 41 puntos porcentuales corresponden al efecto de caché de prefijo habilitado mediante `cache_control` en el brazo instrumentado; el baseline tenía cero cache-read y el brazo B, 72.447 tokens. El 2,49% [1,50; 3,57] del gateway procede de una reproducción contrafactual separada de 19.779 solicitudes, 51 sesiones y 13 días, con baseline parcialmente reconstruida, no de una doble ejecución live. Se distinguen importes measured, reference-valued y estimated; la telemetría incluye recuentos, costes y metadatos permitidos. La precisión Sonnet fue 56/56 en cada brazo y la latencia mediana de llamadas al proveedor fue 1.969 ms (A) y 2.225 ms (B), sin demostrar no inferioridad. El umbral económico usa F como fee mensual total y $r = 0{,}315/0{,}30$ como razón en un único punto, no como pendiente validada. Las muestras pequeñas, la deriva temporal, la frescura de la caché exacta, la organización única, la condición de desarrollador de los autores y los logs gateway privados limitan las conclusiones. Quedan para trabajo futuro el diseño factorial, el barrido de ρ, experimentos jerárquicos de varios días, pruebas de no inferioridad, replicación externa y conciliación con facturas.

References

[1] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of EMNLP*, 2023.

[2] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, et al. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of ACL 2024*, pp. 963–981, 2024. `https://aclanthology.org/2024.findings-acl.57/`

[3] Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems 36*, 2023.

[4] In Gim, Guojun Chen, Seung-seob Lee, Nils Sathiamoorthy, Shriram Patil, Michael Kaminsky, and Petros Maniatis. Prompt Cache: Modular attention reuse for low-latency inference. arXiv:2311.04934, 2023.

[5] Bin Gao, Ziming He, Purushotham Sharma, Qingyang Kang, Djordje Jia, et al. Cost-efficient large language model serving for multi-turn conversations with CachedAttention. In *Proceedings of USENIX NSDI*, 2024.

[6] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, et al. SGLang: Efficient execution of structured language model programs. arXiv:2312.07104, 2023.

[7] Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of EuroSys 2025*, pp. 94–109. `https://doi.org/10.1145/3689031.3696098`

[8] Anthropic. Prompt Caching—Documentation. `https://docs.anthropic.com/en/docs/build-with-claude/prompt-caching`

[9] OpenAI. Prompt Caching—API Guide. `https://platform.openai.com/docs/guides/prompt-caching`

[10] Fu Bang. GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings. In *Proceedings of the 3rd NLP-OSS Workshop*, 2023.

[11] Jiaxing Li, Chi Xu, Feng Wang, Isaac M. von Riedemann, Cong Zhang, and Jiangchuan Liu. SCALM: Towards semantic caching for automated chat services with large language models. arXiv:2406.00025, 2024.

[12] Waris Gill, Mohamed Elidrisi, Pallavi Kalapatapu, Ammar Ahmed, Ali Anwar, and Muhammad Ali Gulzar. MeanCache: User-centric semantic caching for LLM web services. arXiv:2403.02694, 2024.

[13] Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. arXiv:2305.05176, 2023.

[14] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs with preference data. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.18665.

[15] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, et al. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of ACM SOSP*, 2023.

[16] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of ICML*, 2023.

[17] OpenTelemetry Project. Generative AI semantic conventions. `https://opentelemetry.io/docs/specs/semconv/gen-ai/`

[18] Joint Committee for Guides in Metrology. *Evaluation of measurement data—Guide to the expression of uncertainty in measurement*. JCGM 100:2008, 2008.

[19] Joint Committee for Guides in Metrology. *International vocabulary of metrology—Basic and general concepts and associated terms (VIM)*, 3rd ed. JCGM 200:2012, 2012.

[20] Anthropic. Claude Code—Agentic coding tool documentation. `https://www.anthropic.com/claude-code`

[21] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.

[22] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.

[23] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, 1997.

[24] Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron, et al. vCache: Verified semantic prompt caching. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2502.03771.

[25] Dongxin Guo, Jikun Wu, and Siu Ming Yiu. RouteNLP: Closed-loop LLM routing with conformal cascading and distillation co-optimization. arXiv:2604.23577, 2026.

[26] Bernhard C. Geiger. Notation matters: A benchmark study of token-optimized formats in agentic AI systems. arXiv:2605.29676, 2026.

[27] Aditeya Baral. Closing the operational gap in semantic caching. arXiv:2606.19719, 2026.

[28] BiVelio. Calibrated semantic-cache evaluation and abstention report. Public benchmark documentation, September 2026. `https://savings.bivelio.com/benchmarks`