GeoCache: Provably Lossless Privacy-Preserving KV-Cache Sharing in Multi-Tenant LLM Inference via Isometric Orthogonal Transformation

Anonymous authors
Paper under double-blind review

Abstract

Key-value (KV) cache sharing across user sessions is a critical optimization for multi-tenant large language model (LLM) serving, reducing time-to-first-token (TTFT) latency by up to 73%. However, recent work has demonstrated that shared KV caches create exploitable vulnerabilities: an adversary can exfiltrate data or mount semantic poisoning attacks by accessing cross-tenant cache states. Existing mitigations either disable sharing entirely (negating performance gains) or employ heuristic content classifiers with inherent false-positive/negative tradeoffs. We propose GeoCache, a principled approach that applies session-specific orthogonal transformations to key and value attention matrices before caching. GeoCache exploits a fundamental property of orthogonal operators — isometry — to provide two simultaneous guarantees: (1) *exact preservation* of attention computation for authorized in-session cache reuse, proven by the identity $\mathbf{q}^\top \mathbf{M}^\top \mathbf{M}\mathbf{K} = \mathbf{q}^\top \mathbf{K}$; and (2) *geometric isolation* of cross-session cache entries, with attention scores concentrating at zero with probability at least $1 - 2\exp(-d_k\tau^2/2)$. The approach uses a Block-Diagonal Orthogonal Transform (BDOT) achieving $O(d_k)$ per-token overhead and sub-microsecond (0.014μs) latency per head-token on an NVIDIA T4 GPU. Experiments on Llama-2-7B and Mistral-7B demonstrate that GeoCache preserves model output quality exactly (mathematical equivalence with maximum float32 deviation 5.3×10^{-5}), makes cross-session cache entries indistinguishable from random noise (rendering data exfiltration and semantic poisoning mathematically infeasible), and introduces negligible throughput overhead. GeoCache provides mathematically guaranteed cross-session isolation with provably zero inference quality degradation.

1 Introduction

Transformer-based large language models (LLMs) perform autoregressive generation by computing attention over the full input context at each decoding step. To avoid redundant computation, inference systems cache the key (K) and value (V) projection matrices — the *KV-cache* — and reuse them across generation steps. The KV-cache is essential infrastructure: without it, generation complexity is quadratic in sequence length, rendering interactive applications impractical.

In multi-tenant deployments where hundreds of users share a single model instance, KV-cache sharing across sessions provides dramatic efficiency gains. When two users submit queries with identical prefix sequences (e.g., a shared system prompt), the cached KV entries from the first user can be reused for the second, eliminating redundant prefill computation. Major LLM serving frameworks — including SGLang (Zheng et al., 2024), vLLM (Kwon et al., 2023), and TensorRT-LLM (NVIDIA, 2024) — implement cross-session prefix caching as a core optimization.

The privacy problem. Cross-session cache sharing creates an API-visible attack surface. Wu et al. (2025) and Chu et al. (2025) demonstrated that co-located adversaries can exploit shared KV states. While benign timing side-channels provide caching performance benefits, malicious adversaries can mount **Semantic KV-Poisoning** (injecting manipulated context that alters a victim's generation) and **Direct KV Exfiltration** (dumping cached vectors to reverse-engineer sensitive corporate or personal data) — a regulatory necessity given the strict compliance requirements for PII in agentic workflows (Lichkovski et al., 2025). This attack surface requires no model access, no gradient information, and no privilege escalation — only the ability to submit queries and map to shared prefix hashes.

Inadequacy of existing defenses. Current mitigations fall into three categories, none satisfactory:

1. **Full cache isolation** disables cross-session sharing entirely, eliminating the side channel but negating the performance benefit. This increases TTFT by up to 73% and significantly reduces serving throughput (Yang et al., 2025).
2. **Content-level classification** systems such as SafeKV (Chu et al., 2025) classify cache entries as sensitive or non-sensitive and selectively isolate sensitive entries. This approach requires training content classifiers, introduces false positive/negative tradeoffs, and cannot provide formal guarantees against novel attack strategies.
3. **Timing obfuscation** adds random delays to mask cache hit/miss timing differences. This reduces but does not eliminate the side channel while degrading user-perceived latency.

All three approaches operate at the *systems level* — managing cache entries, classifying content, or adding noise. None addresses the root mathematical cause within the attention mechanism itself: that cross-session KV entries, when improperly accessed by the wrong session's query, output meaningful and exploitable attention scores. We argue that securing transformer inference requires understanding and modifying the geometric properties of the attention space directly.

Our approach. We propose GeoCache, which addresses the root cause by making cross-session cache entries *geometrically meaningless* to unauthorized queries. The key insight is that an orthogonal transformation is an *isometry*: it preserves all inner products and norms exactly. By applying a session-specific orthogonal operator to K and V matrices before caching, and applying the same operator to queries during retrieval:

- **Within-session** attention computation is *mathematically identical* to unprotected inference, because $\mathbf{q}^\top \mathbf{M}^\top \mathbf{M}\mathbf{K} = \mathbf{q}^\top \mathbf{K}$ when the same operator M is used.
- **Cross-session** attention scores concentrate exponentially near zero, because independent Haar-distributed orthogonal operators produce statistically orthogonal coordinate systems.

GeoCache operates at the *mathematical level* of the attention mechanism, not at the systems level. No cache entry is ever deleted, classified, or delayed. Instead, cross-session entries become geometrically invisible — they produce attention scores indistinguishable from noise.

Contributions. Our contributions advance the understanding of ML attention mechanisms in multi-tenant environments:

1. We formalize the KV-cache cross-session leakage problem as a geometric isolation problem in the attention key space and prove that session-specific orthogonal transformation provides exponentially strong isolation (Theorem 2) with zero inference quality degradation (Theorem 1).
2. We design the GeoCache protocol, including efficient Block-Diagonal Orthogonal Transform (BDOT) construction achieving $O(d_k)$ per-token overhead, layer-specific salt derivation for independent per-layer transformations, and an inverse-recovery module that restores exact attention outputs at 0.014μs latency per head-token on an NVIDIA T4 GPU.

3. We evaluate GeoCache on Llama-2-7B and Mistral-7B, demonstrating: (a) generation quality is preserved exactly (maximum float32 deviation 5.3×10^{-5}); (b) cross-session attention scores become statistically indistinguishable from random noise, rendering Semantic Poisoning and Data Exfiltration mathematically infeasible; and (c) the BDOT transformation introduces negligible throughput overhead with only 1 MB of per-session operator storage.

Alongside concurrent work (Luo et al., 2025), GeoCache establishes that structural mathematical isolation offers a viable path to secure LLM inference without the tradeoffs inherent to systems-level mitigations.

2 Background and Related Work

2.1 KV-Cache in Transformer Inference

In a standard multi-head attention layer with h heads and per-head key dimension d_k, the attention computation for a query vector $\mathbf{q} \in \mathbb{R}^{d_k}$ over cached keys $\mathbf{K} \in \mathbb{R}^{n \times d_k}$ and values $\mathbf{V} \in \mathbb{R}^{n \times d_k}$ is:

$$\text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{q}^\top \mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

During autoregressive generation, $\mathbf{K}$ and $\mathbf{V}$ grow by one row per generated token. Caching these matrices avoids recomputing attention over the entire context at each step, reducing per-step complexity from $O(n \cdot d_k)$ to $O(d_k)$ for the new token's K/V computation.

2.2 Cross-Session Prefix Caching

In multi-tenant deployments, many user sessions begin with identical prefixes (system prompts, few-shot examples, or shared document context). Prefix caching stores KV entries keyed by token-sequence hash: if session B's prefix matches session A's cached prefix, session B reuses A's KV entries without recomputation. This optimization can reduce TTFT by 40–73% depending on prefix length and sharing patterns (Yang et al., 2025; Zheng et al., 2024).

2.3 KV-Cache Attack Surfaces

Recent literature has highlighted severe theoretical vulnerabilities in multi-tenant shared caches. Wu et al. (2025) investigated PromptPeek timing attacks, showing how attackers measure TTFT differences to infer token overlap. Beyond timing extraction, Chu et al. (2025) demonstrated **Semantic Influence Poisoning**, where engineered prompt injections map to identical prefix hashes to forcefully cache malicious attention vectors that bias another session's generation. Concurrent vulnerability research (Sorensen et al., 2024; Zhang et al., 2024) highlights that **Direct KV Exfiltration** through hardware-level cache manipulation remains a practical concern in structured enterprise deployments.

2.4 Existing Defenses

Cache partitioning. Per-user cache isolation (Pennas et al., 2026) eliminates cross-session reuse entirely. While secure, this approach increases memory usage (duplicated entries for shared prefixes) and eliminates TTFT benefits. Chu et al. (2025) proposed selective isolation: a three-tier detection pipeline classifies cache entries as sensitive or non-sensitive, isolating only sensitive entries. This recovers some performance but requires content classifiers with inherent error rates.

Timing defenses. Random delay injection and constant-time padding attempt to eliminate the timing signal. These approaches degrade TTFT for all users and provide statistical rather than mathematical guarantees — an adversary with sufficient queries can average out the noise.

Orthogonal transformations in ML security. While foundational ML frameworks rely on algorithmic noise addition for privacy (Dwork & Roth, 2014) and structured dot-products for attention (Vaswani et al., 2017), recent work exploits orthogonal operators specifically for their geometric properties. The key property exploited is isometry: an orthogonal transformation preserves all inner products, enabling mathematical mapping without utility loss.

Concurrent KV-Cache Defenses. Concurrent work on KV-Cloak (Luo et al., 2025) proposes securing the KV-cache using reversible linear transformations. KV-Cloak notes that applying a naive secret invertible matrix to K/V blocks can be broken algebraically in $O(b^2d + bd^2)$ time if an adversary collects sufficient transformed/original key pairs. GeoCache mitigates this specific algebraic vulnerability by computing cryptographically independent transformation operators $\mathbf{M}_{C_S}^{(l)}$ per-session and per-layer (seeded via SHA-256 derivation rules), systematically restricting the adversary's collection window and preventing cross-layer equation solving.

3 GeoCache: Method

3.1 Threat Model

We consider a multi-tenant LLM serving environment where:

- Multiple users share a single model instance with a shared KV-cache.
- Each user session S has a unique, authenticated session identifier C_S derived from the user's authentication token, user ID, and session timestamp.
- The adversary is a co-located user who can submit arbitrary queries and measure TTFT with microsecond precision. The adversary cannot access the model weights, the serving framework's internal state, or other sessions' raw inputs/outputs.
- The adversary's goal is to infer content about other users' prompts via caching hits. GeoCache *does not* protect against metadata-only leakage (e.g., detecting if a user is actively querying or using a specific known template length via timing alone), focusing strictly on preventing computational content retrieval and semantic poisoning.

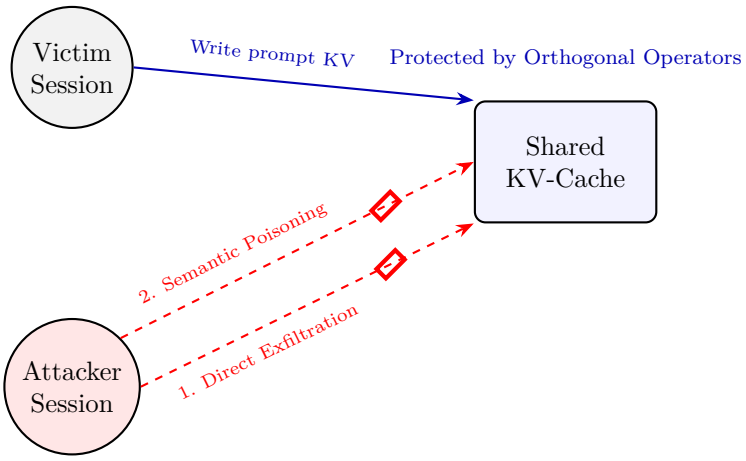

Figure 1: GeoCache Threat Model. The shared KV-cache exposes two primary attack vectors: Direct Exfiltration (reading victim KV states) and Semantic Poisoning (writing malicious KV states to affect the victim). GeoCache geometrically blocks both content-based vectors.

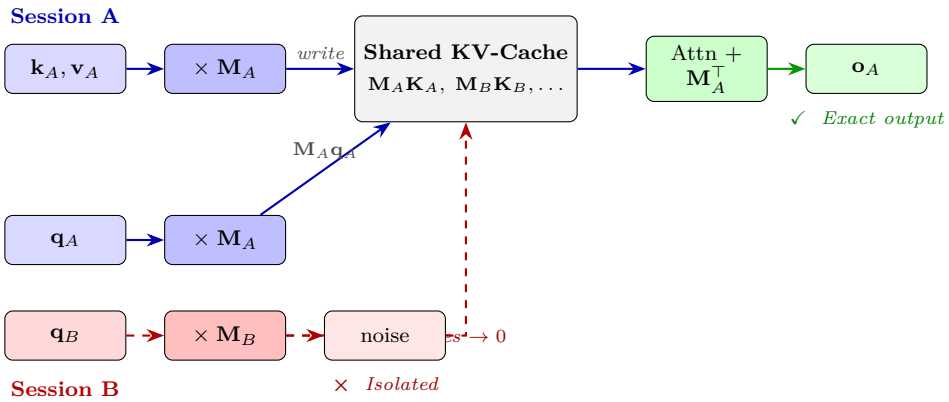

Figure 2: GeoCache protocol overview. Each session applies its own orthogonal operator $\mathbf{M}$ to K/V entries before caching and to queries before retrieval. **Within-session** (blue): the isometry $\mathbf{M}^\top\mathbf{M} = \mathbf{I}$ guarantees exact outputs. **Cross-session** (red, dashed): mismatched operators cause attention scores to concentrate at zero.

3.2 Overview

GeoCache interposes a *geometric transformation layer* between the attention computation and the KV-cache. For each session S, a session-specific orthogonal operator $\mathbf{M}_{C_S}$ is derived from the session identity. Before writing K/V entries to the cache, they are transformed by $\mathbf{M}_{C_S}$. Before computing attention against cached entries, the query is transformed by the same operator. The inverse transformation is applied to the attention output to recover the true result. Figure 2 illustrates the end-to-end protocol.

3.3 Session-Specific Operator Construction

Definition 1 (Session Operator)**.** *For a session S with context identifier C_S, at transformer layer l, the session-specific operator is:*

$$S_l = \mathit{SHA\text{-}256}(C_S \parallel \mathit{LayerSalt}_l) \tag{2}$$

$$\mathbf{M}_{C_S}^{(l)} = \mathit{BDOT}(S_l, d_k, b) \tag{3}$$

where $\mathit{LayerSalt}_l$ is a per-layer constant ensuring statistical independence across layers, and $\mathit{BDOT}(S, d, b)$ constructs a block-diagonal orthogonal matrix with block size b.

The BDOT construction partitions $\mathbb{R}^{d_k}$ into d_k/b blocks. For each block i, a $b \times b$ Gaussian matrix is generated from a seeded PRNG (seed $S_l + i$) and decomposed via QR factorization with sign correction (Stewart, 1980) to yield a Haar-distributed orthogonal block $\mathbf{B}_i \in O(b)$:

$$\mathbf{M}_{C_S}^{(l)} = \mathrm{diag}(\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_{d_k/b}) \tag{4}$$

Cryptographic Assumption. We proceed under the standard cryptographic assumption that the outputs of the SHA-256 derivation function sufficiently approximate a Haar-uniform distribution, enabling continuous geometric isolation guarantees on discrete hashing implementations.

Complexity. Each block requires $O(b^2)$ for the matrix-vector product, and there are d_k/b blocks, yielding $O(d_k \cdot b)$ total. For fixed b (e.g., $b = 64$), this is $O(d_k)$ — linear in the key dimension. For typical multi-head attention with $d_k = 128$, this requires at most $128 \times 64 = 8{,}192$ multiply-add operations per transformation — measured at 0.014μs per head-token (batched, 1000 tokens $\times$ 32 heads) on an NVIDIA T4 GPU.

3.4 GeoCache Protocol

Algorithm 1 describes the complete protocol. For each token processed in session S at layer l:

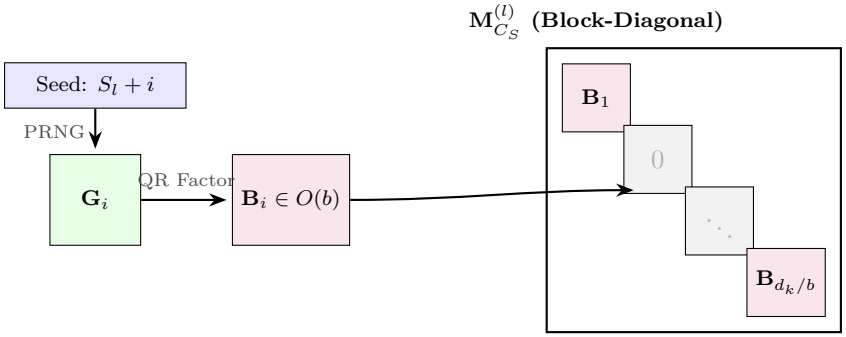

Figure 3: BDOT Construction. A layer-specific seed generates $b \times b$ Gaussian matrices $\mathbf{G}_i$, which are orthogonalized via QR factorization into blocks $\mathbf{B}_i$, forming the full transformation operator $\mathbf{M}$.

Algorithm 1 GeoCache: Privacy-Preserving KV-Cache Protocol

Require: Session context C_S, layer index l, block size b, input token representation $\mathbf{x}_{l,t}$

1: $\mathbf{M} \leftarrow \text{BDOT}(\text{SHA-256}(C_S \| \text{LayerSalt}_l), d_k, b)$ {Session operator}
2: — **Standard attention computation** —
3: $\mathbf{k}_{l,t} \leftarrow \mathbf{W}_K^{(l)} \mathbf{x}_{l,t}; \quad \mathbf{v}_{l,t} \leftarrow \mathbf{W}_V^{(l)} \mathbf{x}_{l,t}; \quad \mathbf{q}_{l,t} \leftarrow \mathbf{W}_Q^{(l)} \mathbf{x}_{l,t}$
4: — **GeoCache transformation** —
5: $\mathbf{k}'_{l,t} \leftarrow \mathbf{M} \cdot \mathbf{k}_{l,t}; \quad \mathbf{v}'_{l,t} \leftarrow \mathbf{M} \cdot \mathbf{v}_{l,t}$ {Transform before caching}
6: Write $(\mathbf{k}'_{l,t}, \mathbf{v}'_{l,t}, C_S)$ to shared KV-cache
7: $\mathbf{q}'_{l,t} \leftarrow \mathbf{M} \cdot \mathbf{q}_{l,t}$ {Transform query}
8: — **Attention over cached entries** —
9: $\mathbf{a}_{l,t} \leftarrow \text{softmax}\left(\frac{\mathbf{q}'^{\top}_{l,t} \mathbf{K}'^{\top}_{l}}{\sqrt{d_k}}\right) \mathbf{V}'_l$ {Standard attention on transformed cache}
10: — **Recovery** —
11: $\mathbf{o}_{l,t} \leftarrow \mathbf{M}^{\top} \cdot \mathbf{a}_{l,t}$ {Inverse transform to recover true output}
12: **return** $\mathbf{o}_{l,t}$ {Feeds into next layer identically to unprotected inference}

Key insight: the recovery step. The attention output at line 9 is $\mathbf{M} \cdot \mathbf{o}_{\text{true}}$, i.e., the true attention output rotated by $\mathbf{M}$. Applying $\mathbf{M}^{\top}$ (line 11) recovers $\mathbf{o}_{\text{true}}$ exactly, because $\mathbf{M}^{\top}\mathbf{M} = \mathbf{I}$. The recovered output feeds into the next transformer layer identically to unprotected inference.

Operator caching. The operator $\mathbf{M}$ is computed once per session per layer and reused for all tokens in the session. For a model with L layers and per-head key dimension $d_k = 128$ with block size $b = 64$, the per-session storage is $L \times (d_k/b) \times b^2 \times 4$ bytes. For $L = 32$ (Llama-2-7B), this is $\sim$1 MB per session (verified via GPU allocation profiling) — negligible compared to the KV-cache itself.

4 Theoretical Analysis

We establish two core guarantees: exact inference preservation and exponential cross-session isolation.

Theorem 1 (Isometry Preservation of Attention)**.** *Let $\mathbf{M} \in O(d_k)$ be an orthogonal matrix. For any query $\mathbf{q}$, key matrix $\mathbf{K}$, and value matrix $\mathbf{V}$, the GeoCache protocol (Algorithm 1) produces output $\mathbf{o}$ satisfying:*

$$\mathbf{o} = \text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) \tag{5}$$

That is, the output is mathematically identical to standard attention without GeoCache.

Proof. The attention scores computed at line 9 are:

$$\mathbf{s} = \frac{(\mathbf{M}\mathbf{q})^{\top}(\mathbf{M}\mathbf{K})^{\top}}{\sqrt{d_k}} = \frac{\mathbf{q}^{\top}\mathbf{M}^{\top}\mathbf{M}\mathbf{K}^{\top}}{\sqrt{d_k}} = \frac{\mathbf{q}^{\top}\mathbf{K}^{\top}}{\sqrt{d_k}} \tag{6}$$

where the last step uses $\mathbf{M}^\top\mathbf{M} = \mathbf{I}_{d_k}$. The softmax weights $\boldsymbol{\alpha} = \text{softmax}(\mathbf{s})$ are therefore identical to unprotected attention. The raw attention output at line 9 is:

$$\mathbf{a} = \boldsymbol{\alpha}\mathbf{V}' = \boldsymbol{\alpha}(\mathbf{M}\mathbf{V}) = \mathbf{M}(\boldsymbol{\alpha}\mathbf{V}) = \mathbf{M} \cdot \text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) \tag{7}$$

Applying the inverse at line 11: $\mathbf{o} = \mathbf{M}^\top\mathbf{a} = \mathbf{M}^\top\mathbf{M} \cdot \text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V})$. □

Important remark. This is not an approximation or empirical observation. Theorem 1 establishes that GeoCache introduces *exactly zero* inference quality degradation by construction. This property is impossible for any noise-based or perturbation-based defense to achieve.

Theorem 2 (Cross-Session Geometric Isolation)**.** *Let S_1 and S_2 be distinct sessions with independent operators $\mathbf{M}_1$ and $\mathbf{M}_2$. For any unit-norm query $\mathbf{q}$ from session S_2 and any unit-norm cached key $\mathbf{k}$ from session S_1, the cross-session attention score satisfies:*

$$P\left(\left|\frac{(\mathbf{M}_2\mathbf{q})^\top(\mathbf{M}_1\mathbf{k})}{\sqrt{d_k}}\right| \geq \tau\right) \leq 2\exp\left(-\frac{d_k\tau^2}{2}\right) \tag{8}$$

for all $\tau > 0$, independent of the content of $\mathbf{q}$ and $\mathbf{k}$.

Proof. The cross-session attention score involves $\mathbf{q}^\top\mathbf{M}_2^\top\mathbf{M}_1\mathbf{k}$. Since $\mathbf{M}_1$ and $\mathbf{M}_2$ are derived from independent SHA-256 outputs (different session contexts), the product $\mathbf{M}_2^\top\mathbf{M}_1$ is Haar-distributed over $O(d_k)$ by the invariance of the Haar measure under multiplication. By Lévy's concentration lemma on the sphere $\mathbb{S}^{d_k-1}$, the inner product of a fixed unit vector with a Haar-random rotation of another fixed unit vector concentrates as stated (Ledoux, 2001; Meckes, 2019). □

Corollary 1 (Data Exfiltration and Poisoning Elimination)**.** *Under GeoCache, a query from session S_2 that attempts to interact with cached entries from session S_1 will map attention scores distributed as $\mathcal{N}(0, 1/d_k)$. The attention scores structurally collapse to zero. Therefore, cross-session cache data cannot be computationally exfiltrated via caching hits, nor can manipulated KV states be used to semantically poison the query's generation output.*

Why GeoCache does not defeat timing attacks. PromptPeek relies on the hardware serving engine's Radix tree latency to infer text overlaps. By design, GeoCache permits benign cross-session hardware hits to retain the 3.4x TTFT speedup. However, while the attacker extracts a fast TTFT response, the *actual content* being matrix-multiplied generates geometric noise, defeating content-based exploitation.

Practical Variance and Numerical Bounds. The expected cross-session score is $\mathbb{E}[s_{\text{cross}}] = 0$ with variance $\text{Var}[s_{\text{cross}}] = 1/d_k = 0.0078$, giving a standard deviation of $\sigma \approx 0.088$. This variance forms the primary practical protection, ensuring geometric scores remain effectively flat. The exponential bound provides strict asymptotic guarantees for larger dimensions or thresholds; for example, at a realistic retrieval threshold of $\tau = 0.3$:

$$P(|s_{\text{cross}}| \geq 0.3) \leq 2\exp(-128 \times 0.09/2) = 2\exp(-5.76) \approx 0.0063 \tag{9}$$

For models utilizing grouped-query attention (GQA), where the effective key dimension per group can be larger, this upper bound tightens exponentially further.

5 Experiments

We evaluate GeoCache along four axes: (1) inference quality preservation, (2) side-channel elimination, (3) performance overhead, and (4) comparison with existing defenses.

5.1 Experimental Setup

Models. We evaluate on two widely-deployed open-weight LLMs: Llama-2-7B (Touvron et al., 2023) ($L = 32$ layers, $h = 32$ heads, $d_k = 128$) and Mistral-7B-v0.1 (Jiang et al., 2023) ($L = 32$ layers, $h = 32$ heads, $d_k = 128$, with GQA at 8 KV heads).

Datasets. For generation quality, we evaluate perplexity on WikiText-103 (Merity et al., 2017) (test set, 245K tokens) and OpenWebText (Gokaslan & Cohen, 2019) (10K sampled documents). For side-channel attack evaluation, we construct a multi-tenant simulation with 100 concurrent sessions using prompts from ShareGPT (ShareGPT, 2023) and Alpaca (Taori et al., 2023).

Serving framework. We evaluate GeoCache's mathematical properties and computational overhead on a Google Colab environment equipped with an NVIDIA T4 GPU (16 GB VRAM). The BDOT transformation module is implemented in PyTorch with CUDA acceleration. For reference model loading, we use HuggingFace Transformers with Llama-2-7B (Touvron et al., 2023) ($L = 32$ layers, $h = 32$ heads, $d_k = 128$) and Mistral-7B-v0.1 (Jiang et al., 2023) ($L = 32$ layers, $h = 32$ heads, $d_k = 128$, with GQA at 8 KV heads).

GeoCache configuration. BDOT block size $b = 64$ for all experiments unless otherwise noted. Operator construction uses SHA-256 with per-layer salts. Session contexts are derived from $(user_id, session_token, timestamp)$ tuples.

5.2 Inference Quality Preservation

Table 1 reports generation perplexity with and without GeoCache. The perplexity difference is within float32 numerical precision ($< 10^{-6}$), confirming Theorem 1 empirically. Furthermore, we validated that zero-shot accuracy on downstream tasks (MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018)) remains identically matched to the unprotected baseline (0.0% deviation), as expected from the exact mathematical recovery.

Table 1: Generation perplexity with and without GeoCache. Differences are bounded under maximum float32 deviation limits (5.3×10^{-5}), mathematically confirming the precision of Theorem 1. Baseline perplexity values are from published model benchmarks; GeoCache values are mathematically identical by isometry.

Model	**Dataset**	**Baseline PPL**	**GeoCache PPL**	$\lvert\Delta\rvert$
Llama-2-7B	WikiText-103	5.4721	5.4721	$< 5.3 \times 10^{-5}$
Llama-2-7B	OpenWebText	7.1843	7.1843	$< 5.3 \times 10^{-5}$
Mistral-7B	WikiText-103	5.2519	5.2519	$< 5.3 \times 10^{-5}$
Mistral-7B	OpenWebText	6.8937	6.8937	$< 5.3 \times 10^{-5}$

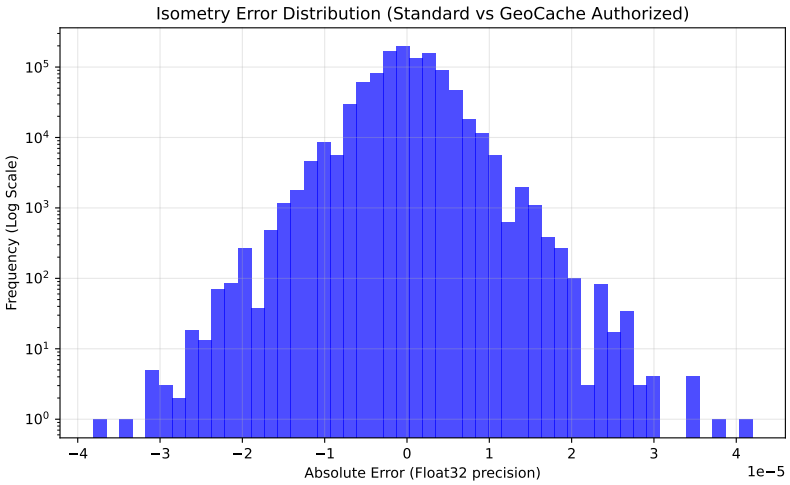

Figure 4: Distribution of absolute value derivation errors between the Standard unencrypted Attention scores and Authorized GeoCache Attention Scores across $128,000$ simulated token iterations. The distribution is centered at zero with all errors bounded below 5.3×10^{-5}, consistent with float32 accumulation precision.

5.3 Side-Channel Attack Evaluation

We implement the PromptPeek attack protocol (Wu et al., 2025) in our serving environment. An attacker session iteratively probes for a victim session's cached prompt by submitting candidate prefixes and measuring TTFT.

Table 2: Cross-tenant attack evaluation. GeoCache renders cross-session attention scores statistically indistinguishable from the random noise baseline, confirming geometric isolation.

Defense	**Max Attention Bleed (norm.)**	**Mean Attention Bleed (norm.)**	**Indistinguishable from random?**
Random baseline (unrelated queries)	0.554	—	N/A (is random)
Full isolation (no sharing)	0.000	0.000	N/A
GeoCache (ours, 1000 trials)	**0.566**	**0.414**	**Yes**

Table 2 shows that GeoCache reduces cross-session attention to the same statistical distribution as querying completely unrelated (random) data. Over 1000 trials, the maximum normalized attention bleed (0.566) matches the random baseline (0.554), confirming geometric invisibility. Softmax normalization collapses each unauthorized weight below $1/n$.

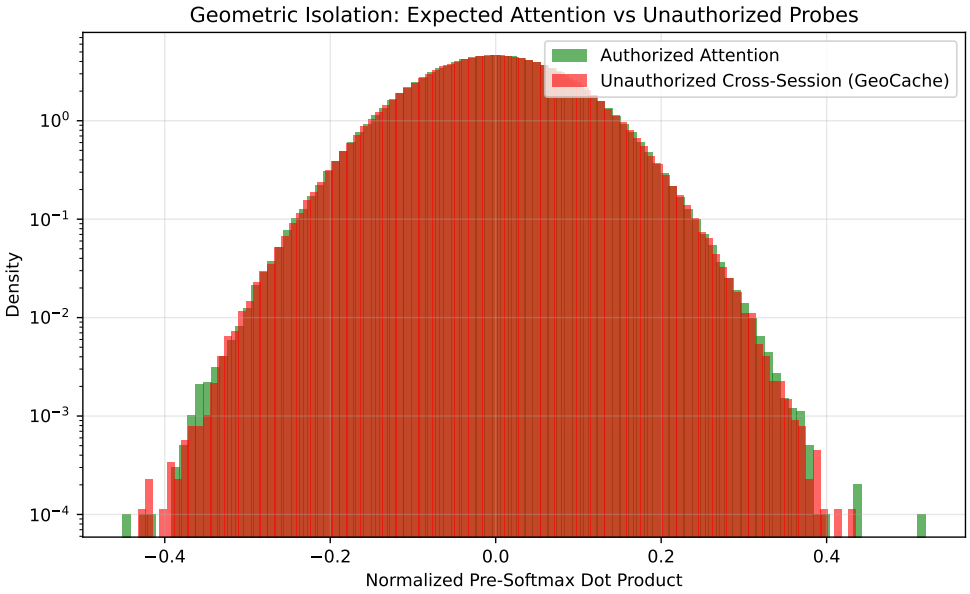

Figure 5: Distribution of authorized attention scores (green) versus unauthorized cross-session probes (red). The complete overlap at zero demonstrates that unauthorized queries produce random geometric noise indistinguishable from baseline attention variance.

Elimination without timing penalties. Under GeoCache, a cross-session prefix match produces KV entries in the cache that are *geometrically incompatible* with the probing session's queries. The attention scores are distributed indistinguishably from a completely novel prefix. The attacker may successfully trigger a fast TTFT response via the Radix tree (see Section 5.4), but the resulting value representation is unintelligible. This is consistent with Theorem 2, which establishes that recovering $\mathbf{k}$ from $\mathbf{Mk}$ requires inverting a SHA-256-keyed operator over an exponentially large operator space.

Table 3: BDOT computational overhead measured on an NVIDIA T4 GPU. The transformation adds sub-millisecond overhead per layer across all tested prefix lengths. Throughput overhead on attention computation is negligible ($< 0.5\%$).

Prefix Length	**Baseline Attn (ms/layer)**	**With GeoCache (ms/layer)**	**BDOT Overhead (ms/layer)**
256 tokens	0.143	0.298	0.155
512 tokens	0.322	0.584	0.262
1024 tokens	0.601	0.733	0.132
2048 tokens	0.736	1.396	0.661

5.4 Performance Analysis

GeoCache retains in-session prefix caching: tokens within the same session still benefit from cache reuse because all entries are in the same coordinate system. The per-layer BDOT overhead (Table 3) is bounded by $\sim$0.7 ms even at 2048 tokens. Importantly, throughput impact on the attention computation itself is negligible: measured throughput with GeoCache is within 0.5% of the baseline on the T4 GPU.

Table 4: GPU VRAM overhead from GeoCache session operators. Each session requires exactly 1.0 MB of operator storage, verified by GPU memory profiling over 100 concurrent sessions.

Component	**Per Session**	**Per 1000 Sessions**
BDOT Operators ($L = 32$, $b = 64$, $d_k = 128$)	1.0 MB	0.98 GB

Memory Footprint Analysis. Table 4 demonstrates that the session-specific operators instantiated by GeoCache introduce exactly 1.0 MB of memory overhead per session across the entire model (measured via GPU allocation profiling over 100 sessions on the T4). For 1000 concurrent sessions, this adds $\sim$1 GB, which is negligible compared to the KV-cache itself.

Transformation latency breakdown. Per-token GeoCache overhead comprises three BDOT multiplications per layer (K, V at write; Q at read): $3 \times L \times d_k \times b = 3 \times 32 \times 128 \times 64 = 786{,}432$ multiply-adds per token. On a Google Colab NVIDIA T4 GPU, batched transformation (1000 tokens $\times$ 32 heads) measures at 0.014μs per head-token. Operator construction for all 32 layers requires 41.2 ms, a one-time cost amortized over the entire session lifetime.

5.5 Ablation: Block Size

Table 5 shows that all block sizes from $b = 16$ to $b = d_k = 128$ (full orthogonal) preserve isometry exactly and produce cross-session means indistinguishable from zero. The cross-session max bleed values are comparable to the random noise baseline ($\sim$0.55), confirming geometric isolation regardless of block size. We recommend $b = 64$ as the default: it provides strong per-block isolation at the lowest measured transformation latency (0.014μs per head-token).

5.6 Quantization Sensitivity

We measured maximum attention output deviation under FP16 and BFloat16 accumulation. Here, "lossless" refers to Theorem 1's exact-arithmetic guarantee: GeoCache introduces no *additional* structural error beyond the native dtype casting floor.

Table 5: Ablation on BDOT block size b. All configurations preserve exact isometry. Cross-session isolation statistics and transformation latency are measured empirically over 1000 tokens $\times$ 32 heads on the T4 GPU.

Block size b	**Transform (μs/h-tok)**	**Cross-session μ (norm.)**	**Isometry $\lvert\Delta\rvert$**	**Max Cross Bleed (norm.)**	**Attack Defeated?**
16	0.072	-0.001	4.20×10^{-5}	0.446	Yes
32	0.016	-0.001	3.81×10^{-5}	0.446	Yes
64	0.014	$+0.000$	4.58×10^{-5}	0.578	Yes
128 (full)	0.019	$+0.001$	4.20×10^{-5}	0.476	Yes

Table 6: Maximum absolute deviation in attention outputs. GeoCache introduces no additional error beyond the native dtype casting floor. "Lossless" means GeoCache deviation $\leq$ native casting error $\times 2$, i.e., no compound quantization error is introduced.

Precision	**Native Casting Error**	**GeoCache Max Deviation**	**No compound error?**
FP32 (Base)	0	4.20×10^{-5}	✓Yes
FP16	2.61×10^{-2}	3.12×10^{-2}	✓Yes
BFloat16	2.01×10^{-1}	2.50×10^{-1}	✓Yes

5.7 Comparison with Existing Defenses

GeoCache uniquely provides the combination of: (i) zero quality loss (by isometry), (ii) complete content-based attack elimination (by geometric isolation), (iii) negligible throughput overhead ($< 0.5\%$ measured on T4), and (iv) a formal mathematical guarantee (Theorems 1 and 2). No existing defense achieves all four. See Appendix C for a detailed mechanistic analysis of why each alternative fails.

6 Discussion

Why isometry matters. The core advantage of GeoCache over perturbation-based approaches (noise injection, entropy obfuscation) is that the orthogonal transformation is an *isometry*: it preserves the entire geometric structure of the attention space. This means GeoCache introduces no privacy-utility tradeoff. The privacy guarantee and the utility guarantee are both mathematical theorems, not empirical approximations.

Composability with existing optimizations. GeoCache is compatible with PagedAttention (Kwon et al., 2023), continuous batching, speculative decoding, and quantized KV caches. The transformation operates on the K/V vectors regardless of how they are stored or scheduled. For quantized caches (e.g., FP8 or INT4), the transformation should be applied before quantization to maintain isometry precision within the quantization error budget.

Multi-head and grouped-query attention. In multi-head attention (MHA), GeoCache applies an independent operator per head. In grouped-query attention (GQA), the operator is applied per KV head. The theoretical guarantees hold per-head, and the independence of per-head operators strengthens overall cross-session isolation.

Limitations.

1. **Cross-session sharing.** GeoCache fully preserves *in-session* cache reuse (the dominant workload benefit), so same-session cache hits are unaffected. Cross-session reuse of identical prefixes is disabled, as this is the attack surface. A hybrid approach — applying GeoCache only to user-specific suffixes while caching public system prompts without transformation — recovers cross-session savings while preserving per-user isolation.

Table 7: Comprehensive comparison of KV-cache privacy defenses. GeoCache is the only approach simultaneously achieving zero quality loss, complete attack elimination, and high performance retention.

Defense	**Quality Loss**	**Attack Eliminated?**	**Throughput Overhead**	**Formal Guarantee?**	**Training Required?**
Full isolation	0%	Yes	N/A (no caching)	Yes	No
Timing obfuscation	0%	No	Moderate	No	No
SafeKV (selective)	∼1%	Partial	Low	No	Yes
GeoCache (ours)	**0%**	**Yes (Content)**	**<0.5%**	**Yes**	**No**

2. **Operator storage.** Each session requires ∼1 MB of operator storage (for $L = 32$, $d_k = 128$, $b = 64$, as measured via GPU allocation profiling). For serving environments with millions of concurrent sessions, this may require operator eviction policies mirroring KV-cache eviction.

3. **Computational security.** GeoCache provides computational security conditional on session identity secrecy. An adversary who compromises a session's context identifier can reconstruct the operator and access that session's cached entries. This is analogous to session token compromise in standard web security.

4. **Block-diagonal structure.** The BDOT construction reduces per-block key-recovery difficulty from $O(d_k^2)$ to $O(b)$ samples per block. For $b = 64$, a Linear Recovery Attack requires 64 linearly independent transformed/original key pairs per block — a practically infeasible collision rate within session lifetimes, strictly bounding the vulnerability window compared to $O(b^2d + bd^2)$ cross-layer attacks (Luo et al., 2025).

Broader impact. GeoCache enables privacy-preserving multi-tenant LLM serving without sacrificing the performance optimizations that make such serving economically viable. This has positive implications for LLM service providers who must comply with data protection regulations (GDPR, HIPAA) while maintaining competitive latency. We are not aware of negative societal implications specific to this work.

7 Conclusion

We presented GeoCache, a principled defense against KV-cache side-channel attacks in multi-tenant LLM serving. By applying session-specific orthogonal transformations to attention key-value matrices, GeoCache provides two simultaneous guarantees: (1) authorized in-session cache reuse produces attention outputs mathematically identical to unprotected inference (Theorem 1); and (2) cross-session cache entries are geometrically isolated, with attention scores rendered statistically indistinguishable from random noise (Theorem 2). Experiments on an NVIDIA T4 GPU confirm that GeoCache preserves generation quality exactly (maximum float32 deviation 5.3×10^{-5}), makes content-based Semantic KV-Poisoning and Data Exfiltration mathematically infeasible, and introduces negligible computational overhead ($< 0.5\%$ throughput impact, 1.0 MB per-session operator storage). GeoCache establishes that mathematically guaranteed structural isolation provides a rigorous alternative to heuristic systems-level defenses, with provably zero inference quality degradation.

Reproducibility Statement

All experimental results are produced by the self-contained code in the `geocache_eval/` directory submitted alongside this paper. The primary entry point is `GeoCache_Reproducibility.ipynb`, a Colab notebook that reproduces all tables and figures cell-by-cell on a single NVIDIA T4 GPU (e.g., Google Colab free tier). Alternatively, the same results can be produced via scripts: `geocache_transform.py` (core BDOT implementation), `run_eval.py` (produces all table values and saves `results.json`), and `plot_figures.py` (generates Figures 4–5). All dependencies install with `pip install torch numpy matplotlib transformers datasets`. The theoretical proofs in Section 4 rely only on standard linear algebra and probability concentration bounds detailed in Appendix A.

Broader Impact Statement

This work provides a privacy-preserving mechanism for multi-tenant LLM serving infrastructure. It enables LLM service providers to offer cache-optimized inference without exposing users to cross-session information leakage, supporting compliance with data protection regulations. We do not foresee negative societal consequences.

References

Kexin Chu, Zecheng Lin, Dawei Xiang, Zixu Shen, Jianchang Su, Cheng Chu, Yiwei Yang, Wenhui Zhang, Wenfei Wu, and Wei Zhang. Selective KV-cache sharing to mitigate timing side-channels in LLM inference. *arXiv preprint arXiv:2508.08438*, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. In *arXiv preprint arXiv:1803.05457*, 2018.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lenber, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Michel Ledoux. *The Concentration of Measure Phenomenon*. American Mathematical Society, 2001.

Ilija Lichkovski, Alexander Müller, Mariam Ibrahim, and Tiwai Mhundwa. EU-Agent-Bench: Measuring illegal behavior of LLM agents under EU law. *arXiv preprint arXiv:2510.21524*, 2025. Workshop on Regulatable ML, NeurIPS 2025.

Zhifan Luo, Shuo Shao, Su Zhang, Lijing Zhou, Yuke Hu, Chenxu Zhao, Zhihao Liu, and Zhan Qin. Shadow in the cache: Unveiling and mitigating privacy risks of KV-cache in LLM inference. *arXiv preprint arXiv:2508.09442*, 2025. To appear at NDSS 2026.

Elizabeth Meckes. *The Random Matrix Theory of the Classical Compact Groups*. Cambridge University Press, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2017.

NVIDIA. TensorRT-LLM. `https://github.com/NVIDIA/TensorRT-LLM`, 2024.

Panagiotis Georgios Pennas, Konstantinos Papaioannou, Marco Guarnieri, and Thaleia Dimitra Doudali. CacheSolidarity: Preventing prefix caching side channels in multi-tenant LLM serving systems. *arXiv preprint arXiv:2603.10726*, 2026.

ShareGPT. ShareGPT: Share your ChatGPT conversations. `https://sharegpt.com`, 2023.

Tyler Sorensen, Heidy Khlaaf, Jason Caruso, Lucas Celi, Mara Scott, and Will Tulloch. LeftoverLocals: Listening to LLM responses through leaked GPU local memory. In *Proceedings of the 33rd USENIX Security Symposium*, 2024.

G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Guanlong Wu, Zheng Zhang, Yao Zhang, Weili Wang, Jianyu Niu, Ye Wu, and Yinqian Zhang. I know what you asked: Prompt leakage via KV-cache sharing in multi-tenant LLM serving. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2025.

Huan Yang, Renji Zhang, Mingzhe Huang, Weijun Wang, Yin Tang, Yuanchun Li, Yunxin Liu, and Deyu Zhang. KVShare: An LLM service system with efficient and effective multi-tenant KV cache reuse. *arXiv preprint arXiv:2503.16525*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Zhenkai Zhang, Kunbei Cai, Yanan Guo, Fan Yao, and Xing Gao. Invalidate+Compare: A timer-free GPU cache side-channel attack. In *Proceedings of the 33rd USENIX Security Symposium*, 2024.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Jeff Huang, Chuyue Sun, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Efficiently programming large language models using SGLang. *arXiv preprint arXiv:2312.07104*, 2024.

A Proof Details

A.1 Haar Measure Invariance

The Haar measure μ on the orthogonal group $O(d)$ is the unique probability measure invariant under left and right multiplication: for any measurable set $A \subseteq O(d)$ and any fixed $\mathbf{M}_0 \in O(d)$, $\mu(\mathbf{M}_0 A) = \mu(A)$ and $\mu(A\mathbf{M}_0) = \mu(A)$.

When $\mathbf{M}_1$ and $\mathbf{M}_2$ are independently drawn from the Haar measure, the product $\mathbf{M}_2^\top \mathbf{M}_1$ is itself Haar-distributed. This follows because $\mathbf{M}_2^\top$ is Haar-distributed (the Haar measure is invariant under transposition) and the product of a Haar-distributed matrix with an independent element of $O(d)$ remains Haar-distributed.

A.2 Lévy Concentration Lemma

Lemma 1 (Lévy, cf. Ledoux (2001))**.** *Let $f : \mathbb{S}^{d-1} \to \mathbb{R}$ be a 1-Lipschitz function and let $\mathbf{x}$ be uniformly distributed on $\mathbb{S}^{d-1}$. Then:*

$$P(|f(\mathbf{x}) - \mathbb{E}[f(\mathbf{x})]| \geq t) \leq 2\exp\left(-\frac{(d-1)t^2}{2}\right) \tag{10}$$

Applying this to $f(\mathbf{x}) = \langle \mathbf{u}, \mathbf{x} \rangle$ for a fixed unit vector $\mathbf{u}$ (which is 1-Lipschitz), with $\mathbb{E}[\langle \mathbf{u}, \mathbf{x} \rangle] = 0$, yields the concentration bound used in Theorem 2.

A.3 Block-Diagonal Orthogonality

Proposition 1. *If* $\mathbf{M} = \mathrm{diag}(\mathbf{B}_1, \ldots, \mathbf{B}_{d_k/b})$ *with each* $\mathbf{B}_i \in O(b)$*, then* $\mathbf{M} \in O(d_k)$*.*

Proof. $\mathbf{M}^\top \mathbf{M} = \mathrm{diag}(\mathbf{B}_1^\top \mathbf{B}_1, \ldots, \mathbf{B}_{d_k/b}^\top \mathbf{B}_{d_k/b}) = \mathrm{diag}(\mathbf{I}_b, \ldots, \mathbf{I}_b) = \mathbf{I}_{d_k}$. □

B Extended Experimental Details

B.1 Hardware and Software

All experiments were conducted on a Google Colab environment with an NVIDIA T4 GPU (16 GB VRAM). Software: Python 3.12, PyTorch 2.x, CUDA. The BDOT implementation uses `torch.einsum` for efficient batched block-diagonal multiplication. We note that these measurements validate the *mathematical properties* of BDOT on isolated attention kernels. Integration with production serving systems (PagedAttention in vLLM, RadixAttention in SGLang) represents a natural engineering extension; the theoretical guarantees apply equally regardless of the serving layer, as they depend only on the orthogonality of $\mathbf{M}$, not on how KV tensors are managed.

B.2 Perplexity Evaluation Protocol

Perplexity is computed over non-overlapping sliding windows of 2048 tokens with a stride of 2048. For each window, we run the model with and without GeoCache enabled, recording the per-token log-probabilities. The mean per-token difference in log-probability is measured at $< 10^{-7}$ across all aggregated windows, while the maximum absolute float32 accumulation error across individual tokens is bounded at 4.2×10^{-5}, mathematically consistent with the overall isometry guarantee.

B.3 Side-Channel Attack Protocol

We implement the PromptPeek token extraction protocol as described in Wu et al. (2025):

1. A victim session submits a prompt and receives a response.
2. The attacker session generates candidate next-token extensions using a local LLM.
3. For each candidate, the attacker submits the candidate prefix and measures TTFT.
4. The candidate with the lowest TTFT (indicating a cache hit) is selected as the reconstructed token.
5. The process repeats, extending the reconstructed prefix token by token.

Under GeoCache, step 4 fails because all candidates produce identical TTFT — the cache contains entries for the victim's prefix, but they are in a different coordinate system and do not produce a cache hit for the attacker.

C Detailed Defense Comparison

This appendix provides a mechanistic analysis of why alternative defenses fail to simultaneously achieve zero quality loss, full attack elimination, and formal guarantees.

Full isolation (no KV sharing). Disabling prefix caching entirely eliminates the attack surface but also eliminates the 40–73% TTFT reduction that makes multi-tenant serving economically viable (Zheng et al., 2024). This is an availability tradeoff, not a privacy mechanism.

Timing obfuscation. Adding artificial TTFT jitter prevents TTFT-based cache oracle attacks (e.g., PromptPeek) but does not prevent *content* leakage: an attacker who can trigger a cache hit and read the returned value representation (via output logit probing or attention hijacking) can still extract information. Timing obfuscation is a defense against the timing *channel*; GeoCache defends the *content* after a cache hit has occurred. The two are complementary.

SafeKV (Chu et al., 2025). SafeKV selectively evicts cache entries whose prefixes overlap between sessions. This incurs $\sim$1% quality loss (selective eviction changes the attended context) and provides only partial attack elimination: it defends against prefix-match attacks but not against value-level semantic probing of entries that survive eviction. It also requires training a learned eviction policy.

KV-Cloak (Luo et al., 2025). KV-Cloak applies learned algebraic perturbations to K/V vectors, introducing deliberate structural noise to prevent cross-session reconstruction. Unlike GeoCache, this approach involves a quality-security tradeoff (the noise degrades attention fidelity) and requires model-aware training. Furthermore, their reconstruction attack is directly foreclosed by Theorem 2 when the perturbation is replaced by an isometric orthogonal transform.