# QUANTIZATION ENABLES PRIVATE DENSE RETRIEVAL AGAINST MALICIOUS SERVICE PROVIDERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Dense retrieval, the key component of Retrieval Augmented Generation (RAG), retrieves the most relevant documents by comparing dense vector representations of queries and passages from a large corpus. In privacy-sensitive applications, the server observes the query and controls which evidence is returned, creating both confidentiality and integrity risks. We formulate private dense retrieval as providing query privacy and retrieval integrity against a malicious server, and develop a two-round cryptographic protocol that provides both guarantees. Our protocol reduces private and verifiable retrieval to multiplication of a committed matrix by an encrypted vector and uses low-bit quantization to make this computation practical. We evaluate the resulting trade-off between cryptographic cost, retrieval quality, and downstream RAG accuracy across six embedding models, four language models, and corpora of up to 2.68 million passages. Our results show that, with a clipped quantizer, three-bit quantization largely preserves retrieval quality and downstream accuracy, while a private query over a corpus the size of a clinical reference requires one to three minutes of server time. These results suggest that private dense retrieval is already practical for moderately sized, privacy-sensitive corpora when minute-scale latency is acceptable.

## 1 INTRODUCTION

Dense retrieval is a central paradigm for large-scale information access. Bi-encoder models (Karpukhin et al., 2020; Reimers & Gurevych, 2019; Ni et al., 2022) represent documents and queries as vectors in a shared embedding space and answer queries by nearest-neighbor search. This is the mechanism underlying semantic search, open-domain question answering and the retrieval stage of retrieval-augmented generation (RAG) (Lewis et al., 2020). For exact retrieval by inner-product scoring, the interaction can be expressed as a matrix-vector product: the server holds an embedding matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, where each row represents a document, and the user submits a query embedding $\mathbf{x} \in \mathbb{R}^n$. In such a setting, retrieval reduces to computing the score vector $\mathbf{Ax}$ and fetching the highest-scoring documents.

However, this interaction exposes query embeddings to the server in the clear, potentially revealing sensitive information about the user's intent. In particular, these embeddings can be linked across queries and inverted to recover information about the underlying text (Morris et al., 2023; Song & Raghunathan, 2020). Consider clinical decision support, where AI-assisted synthesis of the medical literature is increasingly common[1] and physicians submit queries to tools such as *UpToDate* and *OpenEvidence*, which report reaching over three million clinicians worldwide and a majority of US physicians, respectively (Wolters Kluwer, 2026a; NBC News, 2026). Every such query is visible to the provider, to a compromised insider or to anyone who breaches the service or its logs.[2] By cross-referencing with a clinician's appointment schedule, an adversary could link individual patients to the topics their physician searched and reconstruct a working diagnosis. Such linkage can disclose protected health information subject to the HIPAA Privacy Rule (Office for Civil Rights, 2002).

[1] 39% of US physicians report incorporating summaries of medical research and standards of care into practice, up from 13% two years earlier (American Medical Association, 2026).

[2] Medical IT providers are attractive targets for cyberattacks in practice, as in the 2026 Cegedim Santé incident that exposed 15 million medical files (Agence France-Presse, 2026).



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

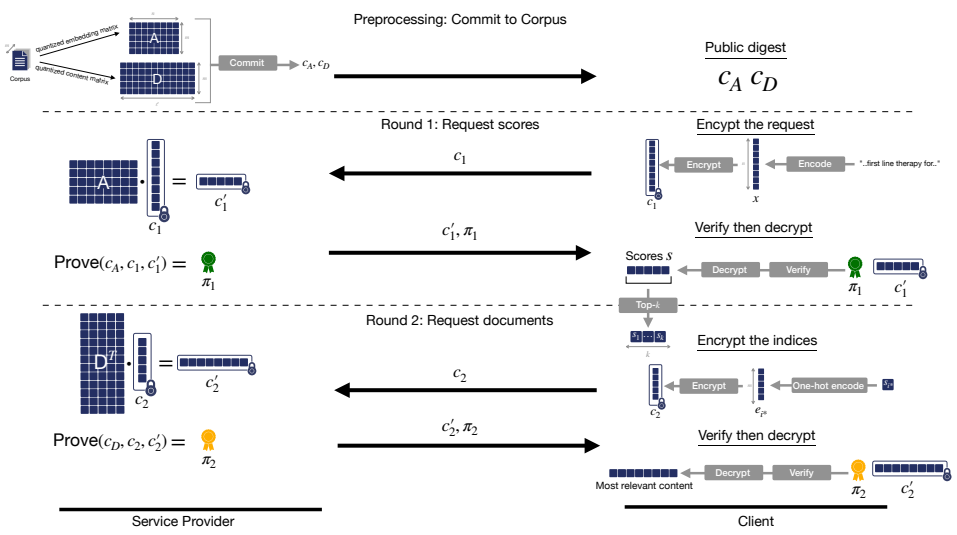



Figure 1: Private dense retrieval. Round 1 returns encrypted scores, from which the client selects $i^*$; Round 2 fetches $\mathbf{D}_{i^*}$. Each proof is verified against the committed $\mathbf{A}$ or $\mathbf{D}$ before decryption.

Beyond query privacy, the client also needs guarantees on the integrity of the retrieved evidence. Indeed, the server currently decides what evidence reaches the user or the downstream language model. One that silently rescores documents or substitutes the returned passage can steer diagnoses or generated answers, and an encrypted-but-unverified protocol gives the clinician no way to notice. Existing systems provide one of these two guarantees but not both, and those that encrypt the query assume an "honest-but-curious" server, which contradicts the threat model that motivates encryption in the first place (§3). Retrieval integrity also bounds *indirect prompt injection* (Greshake et al., 2023) through the retrieval channel: once a corpus has been audited and committed, neither the provider nor an attacker with write access to the corpus can slip adversarial instructions into the passages that reach the reader, since the client accept only content consistent with the audited digest (§2).

**Background.** Homomorpic encryption (HE) (Gentry, 2009) is a form of encryption that allows computations on encrypted data without decrypting it first. The party holding the encryption key can decrypt the result of the computation, but the party performing the computation learns nothing about the underlying data. Over nearly two decades, HE has evolved from a theoretical topic to a vast field of privacy-preserving applications. For example, HE has been used to enable Private Information Retrieval (PIR) (Chor et al., 1998), where a client can retrieve an item from a server's database without revealing to the server which item it is has retrieved. It is crucial to understand that homomorphically encrypted ciphertexts must contain *noise* for security. If not carefully managed, noise accumulation over homomorphic operations can lead to incorrect computation results. As such, smaller noise allows for more operations over the ciphertexts, but guarantees a lower security level. This trade-off is central to our work and we perform in §4.2 the careful balancing act of choosing cryptographic parameters that provide adequate levels of both security and correctness for the tasks at hand.

**Our approach.** Our starting point is the observation that the two stages of dense retrieval, scoring and document fetch, reduce to a product between a public matrix held by the server (*e.g.,* provider) and a private, encrypted vector held by the client (*e.g.,* clinician). We instantiate both with a single maliciously secure primitive, verifiable matrix-vector multiplication under HE (Tremblay Thibault et al., 2026). The server cryptographically commits to a matrix, multiplies its committed matrix by the client's encrypted vector and returns the encrypted result with a succinct proof, which the client verifies before decrypting. The primitive guarantees integrity by construction, and we show in §2 that verifying before decrypting also closes the reaction-attack channel (Chillotti et al., 2016) that honest-but-curious designs leave open. The resulting two-round protocol, illustrated in Figure 1, is exact (every document is scored, with no clustering or hashing approximation), secure against a

malicious server (any manipulation of scores or of the returned document is detected except with negligible probability), client-lightweight (per round the client encrypts one vector, verifies one proof and decrypts one result) and modular (both rounds use one primitive behind one interface, so any faster instantiation may be swapped in directly).

**Setting.** The naive way to hide a query is to send the corpus to the client and have it search locally. That option is not available here: the clinical references that physicians consult are proprietary and licensed per seat, so the provider grants query-time access rather than a copy. As such, the setting we consider is a server-held corpus queried by a client that must not reveal its queries. We target deferred-response settings, such as literature requests to medical librarians or tumor-board preparation, where users expect a researched answer rather than an immediate response and minute-scale latency is acceptable; we evaluate the protocol's cost in this regime (§4.3).

**Contributions.**

1. We formalize private dense retrieval as requiring both query privacy and retrieval integrity against a malicious server, and reduce it to two instances of verifiable matrix-vector multiplication (§2). We give a two-round protocol realizing this functionality, with exact retrieval and a verify-before-decrypt mechanism that provably blocks reaction attacks (§2).
2. Using the protocol as an instrument, we experimentally locate the operating point of private dense retrieval at $b$=3 bits of quantization: retrieval quality and downstream RAG accuracy hold there for six encoders and four readers, on corpora of up to 2.68M passages (§4.1, §4.4).
3. We measure the cost of a private, verifiable query in time and dollars at corpus sizes up to $10^6$ chunks, and show that bit-width buys performance, with a $\approx 2\times$ increase between $b$=2 and $b$=3 (§4.2, §4.3). At $b$=3 a corpus the size of a clinical reference is served in one to three minutes, while the literature behind it takes hours to serve.

## 2 Private Dense Retrieval

The setting comprises two interactive parties. The server holds a corpus of $m$ documents, represented by an embedding matrix $\mathbf{A} \in \mathbb{Z}_p^{m \times n}$ (row $i$ is the quantized embedding of document $i$) and a content matrix $\mathbf{D} \in \mathbb{Z}_p^{m \times \ell}$ (row $i$ is the content of document $i$, padded to length $\ell$). The client holds a private query embedding $\mathbf{x} \in \mathbb{Z}_p^n$. Embeddings are quantized from $\mathbb{R}$ to $\mathbb{Z}_p$ by scalar quantization (Appendix A.6). This is further discussed in §4. This abstracts a common deployment pattern in RAG and enterprise search: the client trusts the embedding model that produced $\mathbf{x}$, but does not trust the retrieval service with the confidentiality of the query or the integrity of the returned evidence.

**Definition 1** (Private Dense Retrieval)**.** *A* private dense retrieval protocol *allows the client to (1) obtain the similarity scores* $\mathbf{s} = \mathbf{A} \cdot \mathbf{x} \in \mathbb{Z}_p^m$*, and (2) retrieve the document* $\mathbf{D}_{i^*}$ *for* $i^* = \arg\max_i s_i$*, subject to:*

1. ***privacy****: the server learns no information about* $\mathbf{x}$ *or* $i^*$ *beyond public parameters; and*
2. ***integrity****: the client accepts only scores and documents consistent with the server's committed database, except with negligible probability.*

The server is actively malicious: it may deviate arbitrarily from the protocol to learn the query or to alter the result. As in standard PIR, we do not require database privacy against the client; our confidentiality goal concerns the client's query. The definition extends to top-$k$ retrieval (§2).

**One primitive suffices.** Both requirements of Definition 1 reduce to verifiable multiplication of a public plaintext matrix by an encrypted vector. The primitive provides privacy (ciphertexts reveal nothing about the encrypted vector by semantic security under the GLWE assumption), soundness (no server strategy makes the client accept anything but the product with the committed matrix, except with negligible probability) and succinctness (digest and proof are small, and verifying is far cheaper than computing the product). Formal definitions are in Appendix A.4; Tremblay Thibault et al. (2026) give a concrete instantiation, which our protocol treats as a module. Improvements to the primitive thus transfer directly to our protocol.

**Protocol.** The protocol runs two rounds of the same sub-protocol (Figure 1). Once per database, the server commits to $\mathbf{A}$ and $\mathbf{D}$ and publishes the digests, which remain fixed for the database's lifetime. Updates require recomputing only the affected commitment, a cost amortized over many queries. In Round 1, the client sends $\mathbf{c}_1 \leftarrow \mathsf{Enc}(\mathsf{sk}_1, \mathbf{x})$ and the server returns $\mathbf{c}'_1 \leftarrow \mathbf{A} \cdot \mathbf{c}_1$ with a proof $\pi_1$ against the committed $\mathbf{A}$. From this the client decrypts the exact score vector and computes $i^* \leftarrow \arg\max_i s_i$ locally. In Round 2, the client sends $\mathbf{c}_2 \leftarrow \mathsf{Enc}(\mathsf{sk}_2, \mathbf{e}_{i^*})$, for $\mathbf{e}_{i^*} \in \{0,1\}^m$ the the one-hot encoding of $i^*$. The server returns $\mathbf{D}^\top \cdot \mathbf{c}_2$ with a proof $\pi_2$ against the committed $\mathbf{D}$, from which the client decrypts $\mathbf{D}_{i^*}$. Each round uses a fresh key and the client aborts unless verification succeeds. The rounds are linked only through $i^*$, which is computed locally. The server sees two semantically secure ciphertext vectors under independent keys and learns no information about $\mathbf{x}$ or $i^*$ beyond public parameters. The client's Round-1 work, namely downloading $m$ ciphertext slots, decrypting them and scanning for the maximum, is linear in $m$. We discuss scaling in §5.

**Verification before decryption.** That the client verifies before decrypting is essential to security. If it decrypted an unverified response, its observable behavior (abort, retry, or which document it fetches in Round 2) would become a function of the plaintext. This is the side channel that reaction attacks exploit (Chillotti et al., 2016). Verifying first makes that behavior depend only on the proof's validity, which the server already knows.

**Theorem 1** (Informal; proof sketch in Appendix C)**.** *If the primitive satisfies the properties of §2, the protocol realizes Definition 1: an actively malicious server learns no information about* $\mathbf{x}$ *or* $i^*$ *beyond public parameters, and cannot cause the client to accept scores or a document inconsistent with the committed database, except with negligible probability.*

**Top-$k$ retrieval.** RAG pipelines typically consume $k > 1$ passages. After Round 1 the client knows the full score vector, so it can select any $k$ indices and fetch each with an independent Round-2 query, multiplying the second round's workload by $k$. Batching the $k$ queries into one server request multiplies communication by $k$ but barely increases latency, since the online server work of Tremblay Thibault et al. (2026) is dominated by a step independent of the number of batched queries (Appendix D).

## 3 Related Work

**Encrypted and verifiable retrieval.** Table 1 situates our protocol relative to prior work. We take dense retrieval by maximum inner product search (Karpukhin et al., 2020; Reimers & Gurevych, 2019; Ni et al., 2022) as given and make the query private and the result verifiable. The natural tool for hiding the query is HE, as PPMI (Bae et al., 2025) does for private LLM interaction. Tiptoe (Henzinger et al., 2023) and PIR-RAG (Wang et al., 2025) instead search published cluster centroids locally and fetch documents by private information retrieval (PIR) (Chor et al., 1998), for which practical single-server lattice-based constructions include XPIR (Aguilar-Melchor et al., 2016). Other systems perform the search itself privately: SANNS (Chen et al., 2020), Servan-Schreiber et al. (2022), Compass (Zhu et al., 2024) and BiSON (Das et al., 2026) run approximate $k$-NN under two-party computation or oblivious RAM, at second-to-millisecond latencies on million- to billion-scale corpora. These systems assume an honest-but-curious server, or two non-colluding servers, and do

Table 1: Private and verifiable retrieval systems. *Exactness*: scores are exact inner products rather than approximate search. *Malicious security*: the stated privacy and/or integrity guarantees hold against an actively deviating server.

| | Privacy | Integrity | Exactness | Malicious security |
|---|---|---|---|---|
| Tiptoe (Henzinger et al., 2023) | ✓ | × | × | × |
| PIR-RAG (Wang et al., 2025) | ✓ | × | × | × |
| PPMI (Bae et al., 2025) | ✓ | × | ✓ | × |
| Compass (Zhu et al., 2024) | ✓ | × | × | × |
| VeriRAG (Lin et al., 2026) | × | ✓ | × | ✓ |
| zkRAG (Jiang et al., 2026) | × | ✓ | × | ✓ |
| **Ours** | ✓ | ✓ | ✓ | ✓ |

not provide integrity against an actively deviating server. In interactive encrypted protocols, such deviations can additionally enable reaction attacks that compromise privacy (Chillotti et al., 2016). Conversely, VeriRAG (Lin et al., 2026) and zkRAG (Jiang et al., 2026) prove in zero knowledge that the server ran an approximate search over a committed database, but send the query in the clear. Our work instead follows the verifiable-computation-over-ciphertext paradigm (Bois et al., 2021; Ganesh et al., 2023; Chatel et al., 2024; Tremblay Thibault et al., 2026), enabling query privacy and retrieval integrity while preserving exactness.

**Low-bit embeddings and the retrieval-metric/downstream gap.** Several methods have been proposed in recent years for compressing embeddings for more efficient downstream tasks, including binary hashing (Yamada et al., 2021), jointly learned product quantization (Zhan et al., 2021), Matryoshka representation learning (Kusupati et al., 2022), int8-trained commercial encoders (Reimers, 2024) and quantization-aware open ones (Yu et al., 2024). Clipping the quantization range to a percentile of the values, rather than to their maximum, is standard in post-training quantization (Banner et al., 2019). That literature primarily optimizes memory footprint and search efficiency, with representation size scaling linearly with bit-width. In our encrypted setting, every bit consumes a fixed share of a noise budget that only larger cryptographic parameters can compensate for (§4.2), which imposes a different cost model. Separately, ranking metrics are known to predict end-task RAG accuracy poorly, as retrieval and generation quality decouple (Salemi & Zamani, 2024; Cuconasu et al., 2024). Section 4.4 studies this decoupling as a function of bit-width, which also sets cryptographic cost, allowing us to select parameters based on downstream rather than retrieval quality alone.

# 4 Experimental Analysis

The protocol preserves exact dense retrieval over quantized embeddings: the only approximation is the $b$-bit quantization itself, while all subsequent arithmetic is computed exactly in $\mathbb{Z}_p$, with $p \geq 2NB^2$. In fact, the GLWE plaintext modulus $p$ (cf. §A.2) must be large enough so that $\mathbb{Z}_p$ contains the quantized embeddings and their inner products, without overflow. A larger $b$ requires a larger $p$, which drives up cryptographic parameters. Holding the retrieval task and security target fixed, the bit-width $b$ is therefore the key parameter coupling retrieval quality and cryptographic cost. Our experiments therefore ask how far $b$ can be reduced before the cryptographic savings cease to justify the loss in retrieval and downstream quality. We measure this trade-off holistically: §4.1 measures retrieval quality across bit-widths and corpus sizes, §4.2 gives the conversion from bit-width to cryptographic cost, §4.3 the concrete costs at scale and §4.4 the downstream performance.

**Corpora.** We evaluate on two biomedical retrieval corpora, chosen to match the setting of §1: *SciFact* (Wadden et al., 2020), scientific claim verification over biomedical abstracts ($m$=5,183 documents, 300 test claims), and *NFCorpus* (Thakur et al., 2021), medical retrieval pairing nutrition and health queries with linked PubMed documents ($m$=3,633, 323 queries). Both are BEIR (Thakur et al., 2021) tasks, so floating-point baselines are comparable with published numbers. The retrieval unit of the protocol is a chunk of at most 1,536 bytes: 512 matrix cells each containing 3 bytes (cf. Table 3). This allows more than half of documents to fit into a single chunk, all the while keeping the cost of the second round low. We cut at sentence boundaries with the title repeated in every chunk. This accommodates the clinical references of §1, which are typically several pages long. The downstream evaluation of §4.4 adds *BioASQ* (Krithara et al., 2023) for biomedical factoid question answering, and the scale study of §4.3 uses *Natural Questions* (Kwiatkowski et al., 2019), which also serves §4.4 as an out-of-domain control. BioASQ and Natural Questions are passage collections already fitting within one chunk. Retrieval quality in §4.1 is measured on the chunked corpora, ranked per chunk and scored per document, so that the relevance labels and the floating-point baselines remain the published document-level ones; Appendix F gives the same table on whole abstracts.

## 4.1 Retrieval quality under quantization

We evaluate six embedding models of similar performance on both corpora, so that training for low-bit readout is the variable. Two were trained for low-bit readout: Cohere `embed-english-v3.0`, trained for int8 and binary output (Reimers, 2024), and Snowflake `arctic-embed-l-v2.0`,

whose training was quantization-aware (Yu et al., 2024; Snowflake, 2024). Four were not: `bge-large-en-v1.5` (Xiao et al., 2023), `e5-large-v2` (Wang et al., 2022), `gte-large` (Li et al., 2023) and `mxbai-embed-large-v1` (Li & Li, 2023; Mixedbread AI, 2024). The last is a control: it advertises int8 and binary use without claiming quantization-aware training. For each $b \in \{2, \ldots, 8\}$ we run the full two-round protocol. Table 8 reports, for $b \in \{2, 3, 4, 8\}$, the fraction of floating-point nDCG@10 (Järvelin & Kekäläinen, 2002; Manning et al., 2008) each encoder retains. Appendix F gives every bit-width on whole abstracts (Table 12) and top-1 agreement at $b \in \{3, 4\}$ (Table 9). At a high level, the results show that down to $b$=4, all encoders retain a high percentage of retrieval quality compared to their floating-point counterparts.

**Appropriate quantization.** A $b$-bit symmetric quantizer maps a chosen magnitude, the scale $\tau$, to the top level $q_{\max} = 2^{b-1} - 1$ and rounds every coordinate to the nearest multiple of the step $\tau/q_{\max}$, so every coordinate within $[-\tau, \tau]$ is off by at most half a step. Max-abs scaling, the usual choice, sets the scale to the largest coordinate of the corpus matrix. BERT-family encoders carry one coordinate holding over $5\%$ of the squared norm of every embedding, and that one coordinate makes the step up to $4\times$ coarser than the other $1023$ need. Appendix F shows the effect: BGE-large clipped at $b$=3 retains 97 and $99\%$ of floating-point nDCG@10 on SciFact and NFCorpus, where max-abs retains 77 and $73\%$ at $b$=3 and only reaches 97 and $96\%$ at $b$=4. We thus use clipping quantization for our experiments.

**Quality at scale.** A larger index gives every query more near-misses to lose to (Reimers & Gurevych, 2021), so we repeat the sweep on Natural Questions from $10^4$ passages to the full 2.68M with five encoders, one of them trained for low-bit readout (Table 13, Appendix F). At $b$=4 four of the five retain at least $98.8\%$ of floating-point Recall@5 at every size, and e5 retains $97.3\%$ at 2.68M. At $b$=3 arctic, BGE and mxbai retain at least $98.4\%$ while gte and e5 fall to 94.5 and $92.0\%$ at 2.68M. At $b$=2 the loss grows with the corpus, from 1 to $4\%$ at $10^4$ to 10 to $23\%$ at 2.68M. The lead of the quantization-aware encoder grows with the corpus: arctic retains 0.8 to 2.7 points more Recall@5 than the conventional four at $10^4$ and 2.1 to 12.8 points more at 2.68M. As such, retrieval quality places the operating point at $b$=3. There, five of the six encoders lose 1 to $3\%$ of nDCG@10 (e5 loses $6\%$ on NFCorpus), and for arctic, BGE and mxbai the loss stays small as the corpus grows.

### 4.2 Quantization robustness lowers cryptographic cost

In this section we formalize how the bit-width $b$ sets the cryptographic parameters, and thus the cost, of Round 1. For both rounds, we fix the ciphertext modulus $q = 2^{64} - 2^{32} + 1$, at which the primitive of Tremblay Thibault et al. (2026) is most efficient, and target $\lambda \geq 128$ bits of security and a decryption failure probability of $2^{-64}$ per coefficient.

**Theorem 2** (Correctness of Round 1)**.** *Let $\mathbf{A} \in \mathbb{Z}^{m \times N}$ and $\mathbf{x} \in \mathbb{Z}^N$ have entries of absolute value less than $B$, let $p \geq 2NB^2$ and let $\mathbf{c} \leftarrow \mathsf{Enc}(\mathsf{sk}, \mathbf{x})$ be a GLWE encryption with ciphertext modulus $q$, plaintext modulus $p$ and Gaussian noise of standard deviation $\sigma$. For every $r > 0$, provided $q/\sigma \geq 2r\sqrt{N}\, B\, p$, each coordinate of $\mathsf{Dec}(\mathsf{sk}, \mathbf{A} \cdot \mathbf{c})$ equals that of $\mathbf{A} \cdot \mathbf{x}$ over the integers, except with probability $\varepsilon(r) \coloneqq 2\exp(-r^2/2)/(r\sqrt{2\pi})$.*

We defer the proof to Appendix A.3. In Round 1 the inner dimension $N$ is the embedding dimension $d = 2^{10}$, the quantized entries lie in $[-(2^{b-1}-1), 2^{b-1}-1]$ so $B = 2^{b-1}$, and we take $p = 2NB^2 = 2^{2b+9}$. At $q = 2^{64} - 2^{32} + 1$ and $r = 9.16$, for which $\varepsilon(r) \leq 2^{-64}$, Theorem 2 thus makes Round 1 correct provided $\sigma \leq \sigma_{\max}(b)$ where

$$\log_2 \sigma_{\max}(b) \;=\; \log_2 q - \log_2(2r) - \tfrac{1}{2}\log_2 N - (b-1) - (2b+9) \;\approx\; 46.8 - 3b. \qquad (1)$$

Every added bit of quantization thus costs three bits of noise budget, one through the entry bound $B$ and two through the plaintext modulus $p$. Security pulls in the other direction and calls for more noise. It admits no closed form, so we evaluate it with the lattice estimator (Albrecht et al., 2015) at $\sigma = \sigma_{\max}(b)$, for a ternary secret and GLWE dimensions $2^{10}$ and $2^{11}$ (Table 2). Dimension $2^{10}$, which matches the embedding dimension, reaches 128 bits at $b = 2$ only; every $b \geq 3$ must double the GLWE dimension $\kappa d$, by taking $d = 2^{11}$ or rank $\kappa = 2$. As such, the cryptography admits two regimes rather than a cost per bit: the smaller ring at $b$=2, and one doubling, paid once, from $b$=3 on. We measure what this doubling costs in §4.3.

Table 2: Quantization-cryptography exchange rate: Round 1 parameters at $q = 2^{64} - 2^{32} + 1$, $\kappa = 1$, ternary secret and $2^{-64}$ failure probability. $\lambda_{10}$ and $\lambda_{11}$ are the lattice-estimator security at $\sigma = \sigma_{\max}(b)$ for LWE dimensions $d = 2^{10}$ and $2^{11}$, and $d$ the smallest dimension at 128 bits.

| $b$ | $p$ | $\sigma_{\max}$ | $\lambda_{10}$ | $\lambda_{11}$ | $d$ |
|---|---|---|---|---|---|
| **2** | $2^{13}$ | $2^{40.8}$ | **138** | 284 | $\mathbf{2^{10}}$ |
| 3 | $2^{15}$ | $2^{37.8}$ | 122 | 253 | $2^{11}$ |
| 4 | $2^{17}$ | $2^{34.8}$ | 110 | 227 | $2^{11}$ |
| 5 | $2^{19}$ | $2^{31.8}$ | 100 | 206 | $2^{11}$ |
| 8 | $2^{25}$ | $2^{22.8}$ | 79 | 161 | $2^{11}$ |

### 4.3 Concrete costs of private dense retrieval

In what follows we quantify the cost of a private and verifiable query, isolate the share of that price paid for malicious security and identify the corpus sizes and workflows the protocol can serve today.

**Measured cost.** Table 3 reports the cost of the full protocol on the evaluation corpora and at scale, at both parameter sets of §4.2. At $b$=2 the server time is of 1.7 s on SciFact and on NFCorpus, and of 2.5 s from $b$=3 on, once the ring is doubled. At $10^6$ chunks the two are of 3.0 and 5.4 minutes. Round 2 multiplies $\mathbf{D}^\top \in \mathbb{Z}_p^{512\times m}$ where Round 1 multiplies $\mathbf{A} \in \mathbb{Z}_p^{m\times 1{,}024}$. Both matrices grow linearly in $m$ and the prover's cost follows this trend. As such, Round 2 costs $\approx 0.5\times$ Round 1 at every corpus size at $b$=2. Per round, the client uploads $\lceil N/d \rceil(\kappa{+}1)d\log_2 q$ bits of ciphertext and downloads $\lceil m'/d \rceil(\kappa{+}1)d\log_2 q$ bits plus the proof, where $m'$ is the output length of that round. Over both rounds, communication is of 528 KiB on SciFact and 496 KiB on NFCorpus at $b$=2, proofs included, and 20 KiB more once the ring is doubled. Most of the server time is spent computing the proof of correct computation. On the SciFact Round-1 shape at $d = 2^{10}$, the matrix-vector product over ciphertexts takes 15 ms without proof generation and 1.18 s with it. Detecting a malicious server rather than trusting it thus costs a factor of 77.

Table 3: Measured cost on the evaluation corpora and at scale, at $b$=2 (Round 1 at $d = 2^{10}$) and from $b$=3 on (Round 1 at $d = 2^{11}$); Round 2 runs at $d = 2^{12}$ with $\ell = 512$ (Table 5). Server time is measured on 48 AMD EPYC 9654 cores; client time is measured on a single core. Prices at the on-demand rate of a 48-core `hpc7a.24xlarge` EC2 instance ($7.20/h).

| | | | $b$=2 / $b\geq$3 | | |
|---|---|---|---|---|---|
| Corpus | Round 1: $\mathbf{A}$ | Round 2: $\mathbf{D}^\top$ | Client | Server | $/query |
| NFCorpus (5,761 chunks) | $5{,}761 \times 1{,}024$ | $512 \times 5{,}761$ | 510 ms / 914 ms | 1.7 s / 2.5 s | 0.003 / 0.005 |
| SciFact (7,510 chunks) | $7{,}510 \times 1{,}024$ | $512 \times 7{,}510$ | 510 ms / 914 ms | 1.7 s / 2.5 s | 0.003 / 0.005 |
| $m = 10^4$ | $10^4 \times 1{,}024$ | $512 \times 10^4$ | 907 ms / 1.7 s | 3.0 s / 5.0 s | 0.006 / 0.010 |
| $m = 10^5$ | $10^5 \times 1{,}024$ | $512 \times 10^5$ | 6.3 s / 12.5 s | 21.7 s / 33.8 s | 0.043 / 0.068 |
| $m = 10^6$ | $10^6 \times 1{,}024$ | $512 \times 10^6$ | 50.4 s / 1.7 min | 3.0 min / 5.4 min | 0.364 / 0.650 |

**The deployable frontier.** Table 3 presents the cost of a private dense retrieval query under our protocol for both $b \in \{2, 3\}$. By §4.1, $b$=3 is the smallest bit-width that keeps retrieval quality, so the following assumes $b$=3, i.e. the larger GLWE parameters. Today, UpToDate, the curated reference clinicians consult first, holds over 12,000 topics (Wolters Kluwer, 2026b), each the length of a review article, i.e., of the order of 2 to $5 \times 10^5$ chunks in all. A private query on a corpus this size costs one to three minutes and is affordable today. On the other hand, PubMed, the literature behind UpToDate, holds over 40 million abstracts (National Library of Medicine, 2026), one or two chunks each. A query on such a large corpus would take 4 to 6 hours of server time at this throughput and incur communication costs of approximately 1 GB. We judge this is too costly for the clinical setting, even in the deferred-response context. Two things can bring the cost down: faster cryptographic machinery, and encoders that keep their quality at $b$=2. The former lies with the cryptographic community. The latter is a matter of representation learning, and by §4.2 it is worth the one ring doubling, i.e. the factor of 1.5 to 1.8 between the two columns of Table 3. As

such, at $b$=3 the protocol serves a corpus of a few hundred thousand chunks in one to three minutes and $10^6$ in five, which covers a curated clinical reference and not the literature behind it.

### 4.4 Downstream task quality under quantization

In the RAG deployments we target, what matters is the accuracy of the answer a language model produces from the retrieved passages rather than the relevance ranking of these passages. The two need not degrade together, since a reader consuming $k$ passages tolerates a reordering provided the evidence it needs appears somewhere in its context (Salemi & Zamani, 2024; Cuconasu et al., 2024). We measure downstream accuracy as a function of $b$ and locate the smallest bit-width it tolerates.

**Methodology.** We follow the standard end-task protocol for retrieval-augmented generation (Karpukhin et al., 2020; Lewis et al., 2020; Petroni et al., 2021): the retriever returns the top-$k$ passages, a fixed reader answers from them, and only the retrieved set changes between the conditions we compare. A question-answering response is correct if it contains a reference answer (Mallen et al., 2023; Asai et al., 2024), and a claim-verification response if it matches the label. We test each bit-width against floating point with an exact McNemar test on paired per-query outcomes. Since a non-significant difference is not evidence of equivalence, we further test equivalence within $\pm 5$ points by two one-sided tests (TOST) on the same paired differences (Appendix E).

**Setup.** We use three endpoints and two encoders, one from each group of §4.1. *SciFact* is evaluated on its original task: the reader receives a claim and the top-$k$ retrieved abstracts and outputs SUPPORT, CONTRADICT or NOINFO, scored against the human labels on the 188 claims with evidence in the corpus, at $k$=5 and $k$=1. *BioASQ* is biomedical factoid QA over a corpus of 35,454 passages, scored by containment of a reference answer. *Natural Questions* is the out-of-domain control at scale, answered over the full 2.68M-passage corpus of §4.1. Each endpoint has 300 queries. We test the performance limits of the reader by measuring its performance in the closed book setting, where it answers from its own knowledge with no passages, and in the gold context setting, where it receives the labelled relevant passages in place of the retrieved ones. Retrieval uses the clipped quantizer with BGE and arctic, and only the retrieved set varies with $b$, so any change in accuracy is attributable to quantization. The reader is `phi-4` (14B) (Abdin et al., 2024). Appendix E repeats every measurement with `OLMo-2-7B` (OLMo Team, 2025) and `Qwen2.5` at 7B and 72B (Qwen Team, 2025), and gives the prompts and corpora.

**Downstream quality holds to $b$=3 and not to $b$=2.** Table 4 shows the result. On closed book, the reader reaches 19% on BioASQ, 31% on NQ and 59% on SciFact, with gold passages 68, 74 and 81%, and with floating-point retrieval it sits between the two, close to the gold row. The effects of quantization below are to be read against that span of 23 to 49 points. No cell shows a significant drop from floating point at $b$=4 or $b$=3, and every cell is equivalent to floating point within 5 points at both bit-widths. This holds at 2.68M passages. The reason is that the RAG endpoint consumes recall rather than rank. More precisely, at $b$=3 the top-5 set of BGE on NQ agrees with the floating-point one on 82% of positions, yet it contains a relevant passage for 70% of questions, as in floating point. Further, §4.1 showed Recall@5 retained at every corpus size. The three other readers agree: over all four, 32 of 32 reader-endpoint-encoder combinations are equivalent to floating point at $b$=4 and 30 of 32 at $b$=3, with a single significant drop of 3.3 points (Table 6).

At $b$=2, accuracy drops significantly in 20 of 32 combinations, by 3 to 9 points, and only 9 remain equivalent to floating point within 5 points. Again, poor recall is responsible for this drop in performance. On NQ, the top-5 set of BGE contains a relevant passage for 60% of questions at $b$=2 against 70% at $b$=3, and the reader answers UNKNOWN more often rather than answering wrongly. The loss grows with the corpus: Recall@5 retained at $b$=2 falls from 98 to 86% for BGE and from 99 to 90% for arctic between $10^4$ and 2.68M passages (Table 13). Interestingly, the loss is uneven. It is largest at $k$=1, where a single reordering at the top removes the only passage the reader sees, at 4 to 9 points for both encoders. At $k$=5 it depends on the encoder: on SciFact, BGE loses 3 to 6 points where arctic loses at most 2.1, with no significant drop for any of the four readers. This shows that the loss at $b$=2 is a property of how the encoder was trained.

Table 4: Downstream accuracy versus bit-width with the clipped quantizer (reader `phi-4`, $n$=300 queries per endpoint). SciFact: accuracy on the 188 evidence-bearing claims; BioASQ and NQ: answer containment, NQ retrieving over the full 2.68M-passage corpus. Closed book: the reader sees no passages; gold context: it sees the labelled relevant passages (at most $k$). **Bold** marks a significant drop from floating point (exact McNemar, $p < 0.05$). Other readers: Appendix E.

| | SciFact $k$=5 | | SciFact $k$=1 | | BioASQ $k$=5 | | NQ $k$=5, 2.68M | |
|---|---|---|---|---|---|---|---|---|
| $b$ | BGE | arctic | BGE | arctic | BGE | arctic | BGE | arctic |
| closed book | 0.585 | | 0.585 | | 0.187 | | 0.313 | |
| gold context | 0.814 | | 0.809 | | 0.680 | | 0.740 | |
| float | 0.734 | 0.713 | 0.766 | 0.739 | 0.633 | 0.647 | 0.643 | 0.643 |
| 8 | 0.734 | 0.718 | 0.771 | 0.739 | 0.633 | 0.643 | 0.657 | 0.637 |
| 6 | 0.729 | 0.713 | 0.771 | 0.739 | 0.633 | 0.640 | 0.650 | 0.640 |
| 4 | 0.723 | 0.718 | 0.771 | 0.739 | 0.637 | 0.637 | 0.650 | 0.627 |
| 3 | 0.707 | 0.713 | 0.761 | 0.755 | 0.643 | 0.640 | 0.623 | 0.640 |
| 2 | **0.676** | 0.697 | **0.686** | **0.681** | 0.613 | **0.607** | **0.573** | 0.600 |

Chunking the SciFact corpus as in §4 costs the conventional encoder one bit and the quantization-aware one none (Table 7, Appendix E). With BGE at $k$=5 the loss reaches 4.3 points at $b$=3, where arctic stays within 5 points of floating point down to $b$=2 on this smaller corpus.

As such, downstream accuracy places the operating point of private RAG at $b$=3 at every corpus size we measured, one bit above the $b$=2 at which §4.2 halves the GLWE parameters required for sufficient security. That frontier bit is decided by the encoder: with a conventional encoder the loss at $b$=2 is of 3 to 9 points, with a quantization-aware one it is within 2.1 points at $k$=5 on SciFact. Whether private RAG runs on the smaller GLWE parameters is thus a property of the encoder.

## 5 Discussion

**Limitations.** The protocol is a drop-in retrieval layer for deployments in which a licensed corpus is hosted by an untrusted service, the queries are sensitive and an answer is expected in minutes. Importantly, our contribution is not an end-to-end privacy-preserving RAG pipeline: we assume the embedding model and the readers run locally or in a trusted environment, and leave a fully private pipeline, which current technology does not yet allow, to future work. In terms of scale, three constraints bound our protocol. The first is the client: the Round-1 response, the Round-2 query and the client's decryption and argmax are all linear in $m$. At $m = 10^6$, Table 3 gives one to two minutes of sequential client work and roughly 16 MB in each direction. The client is kept single-threaded to model a computationally weak device. The second is the encoder, which decides whether a task can go below $b$=3. Our evidence is a controlled comparison of six encoders, of which only two were trained for low-bit readout. What they change is the score margin relative to the rounding step (§4.1); whether a training objective can widen it enough for $b$=2 at scale is an open problem we hand to representation learning. Very recent advances in this regard are promising (Wang et al., 2026). The third constraint is the cryptographic primitive: verifiable HE throughput limits web-scale deployment, but our modular design directly absorbs any improvement to it.

## 6 Conclusion

We have formulated private dense retrieval as a problem of privacy and integrity against a malicious server. We have reduced the underlying computations to verifiable multiplication of a committed matrix by an encrypted vector and used the resulting protocol to measure what privacy costs. With a clipped quantizer, retrieval quality and downstream accuracy hold at $b$=3 for six encoders and four readers on corpora of up to 2.68M passages. A private, verified query over a corpus the size of a clinical reference is then of one to three minutes. At 128 bits of security, only $b$=2 admits the smaller GLWE parameters; $b \geq 3$ costs a factor of 1.8 in server time. Which encoders keep their quality below three bits is set by their score margin relative to the quantization step, and only those trained for low-bit readout do so at $b$=2. Lowering the cost of privacy in dense retrieval is thus not only a cryptography problem, but also one of representation learning.

## AI USE STATEMENT

In this work, we used generative AI tools for feedback on research methodology. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, implement methods, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Additionally, we used generative AI tools to edit software code, generate tables and identify relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

All retrieval-side results use public corpora (SciFact, NFCorpus, Natural Questions, BioASQ) and publicly available encoders. The downstream readers are open-weight models run with structured decoding, so every number in §4.4 is reproducible without access to an API. The quantization code, the protocol implementation and the per-bit-width sweep outputs is be released with the paper[3]. Prompts are fixed across bit-widths within a reader are given in Appendix E. The cryptographic parameters, the correctness constraint and the security estimate are stated in closed form in §4.2 and Appendix A.2, so Table 2 is reproducible without running the protocol. Timings are single-node measurements on a named processor with a fixed thread count, and the benchmark binary asserts verification and exact decryption at every shape.

## ETHICS STATEMENT

This work aims to reduce the exposure of sensitive queries and is motivated by a concrete disclosure risk in clinical decision support. The corpora we use are public research datasets and contain no patient data. We emphasize that query privacy against the retrieval provider does not guarantee privacy over the rest of the technological stack. A protocol such as ours could be invoked to argue compliance while leakage persists elsewhere. The guarantee is exactly that of Definition 1 and no broader. Further, the integrity guarantee is relative to a committed digest. Deploying the protocol without independent attestation of that digest by peers or auditors offers the appearance of verification without its substance.

## REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Agence France-Presse. Hackers steal medical details of 15 million in france, 2026. URL https://www.france24.com/en/live-news/20260227-hackers-steal-medical-details-of-15-million-in-france.

Carlos Aguilar-Melchor, Joris Barrier, Laurent Fousse, and Marc-Olivier Killijian. XPIR: Private information retrieval for everyone. 2016(2):155–174, April 2016. doi: 10.1515/popets-2016-0010.

Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Math. Cryptol.*, 9(3):169–203, 2015. URL http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml.

American Medical Association. 2026 physician survey on augmented intelligence. AMA Center for Digital Health and AI, March 2026. URL https://www.ama-assn.org/system/files/physician-ai-sentiment-report.pdf.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] https://anonymous.4open.science/r/pdr-artifact

Yubeen Bae, Minchan Kim, Jaejin Lee, Sangbum Kim, Jaehyung Kim, Yejin Choi, and Niloofar Mireshghallah. Ppmi: Privacy-preserving llm interaction with socratic chain-of-thought reasoning and homomorphically encrypted vector databases, 2025. URL https://arxiv.org/abs/2506.17336.

Ron Banner, Yury Nahshan, and Daniel Soudry. Post-training 4-bit quantization of convolutional networks for rapid-deployment. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Alexandre Bois, Ignacio Cascudo, Dario Fiore, and Dongwoo Kim. Flexible and efficient verifiable computation on encrypted data. In *Public-Key Cryptography (PKC)*, pp. 528–558, 2021. doi: 10.1007/978-3-030-75248-4_19.

Sylvain Chatel, Christian Knabenhans, Apostolos Pyrgelis, Carmela Troncoso, and Jean-Pierre Hubaux. VERITAS: Plaintext encoders for practical verifiable homomorphic encryption. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2520–2534, 2024. doi: 10.1145/3658644.3670282.

Hao Chen, Ilaria Chillotti, Yihe Dong, Oxana Poburinnaya, Ilya Razenshteyn, and M. Sadegh Riazi. SANNS: Scaling up secure approximate $k$-nearest neighbors search. In *29th USENIX Security Symposium (USENIX Security)*, pp. 2111–2128, 2020.

Ilaria Chillotti, Nicolas Gama, and Louis Goubin. Attacking FHE-based applications by software fault injections. Cryptology ePrint Archive, Paper 2016/1164, 2016. URL https://eprint.iacr.org/2016/1164.

Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998. doi: 10.1145/293347.293350.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 719–729, 2024. doi: 10.1145/3626772.3657834.

Sankha Das, Rohan Ravi, Nishanth Chandran, and Divya Gupta. BiSON: Billion-scale oblivious nearest-neighbor search in milliseconds. Cryptology ePrint Archive, Paper 2026/1343, 2026. URL https://eprint.iacr.org/2026/1343.

Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. Rinocchio: SNARKs for ring arithmetic. *Journal of Cryptology*, 36(4):41, 2023. doi: 10.1007/s00145-023-09481-3.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pp. 169–178, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585062. doi: 10.1145/1536414.1536440. URL https://doi.org/10.1145/1536414.1536440.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023. URL https://api.semanticscholar.org/CorpusID:258546941.

Alexandra Henzinger, Emma Dauterman, Henry Corrigan-Gibbs, and Nickolai Zeldovich. Private web search with tiptoe. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 396–416, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613134. URL https://doi.org/10.1145/3600006.3613134.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.

Yanze Jiang, Xinyang Yang, Xuanming Liu, Yanpei Guo, and Jiaheng Zhang. zkRAG: Efficiently proving RAG retrieval in zero knowledge. Cryptology ePrint Archive, Paper 2026/709, 2026. URL https://eprint.iacr.org/2026/709.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. BioASQ-QA: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(170), 2023. doi: 10.1038/s41597-023-02068-4.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 30233–30249, 2022.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Xianming Li and Jing Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Chenqi Lin, Yubo Cui, Zhelei Zhou, Cheng Hong, Yufei Wang, Zhaohui Chen, and Meng Li. VeriRAG: Efficient zero-knowledge proofs for verifiable retrieval-augmented generation. Cryptology ePrint Archive, Paper 2026/637, 2026. URL https://eprint.iacr.org/2026/637.

Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43:1–43:35, 2013. doi: 10.1145/2535925. URL https://doi.org/10.1145/2535925.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England, 2008. ISBN 978-0-521-86571-5.

Mixedbread AI. mxbai-embed-large-v1 model card. https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1, 2024. Accessed September 2026.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12448–12460, 2023.

National Library of Medicine. About PubMed, 2026. URL https://pubmed.ncbi.nlm.nih.gov/about/. Accessed September 2026.

NBC News. Most U.S. doctors are quietly using this AI tool. few patients know about it, 2026. URL https://www.nbcnews.com/tech/tech-news/openevidence-ai-doctor-medical-physician-login-app-what-npi-uptodate-rcna341064.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022.

HHS Office for Civil Rights. Standards for privacy of individually identifiable health information. final rule, 2002.

OLMo Team. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2523–2544, 2021.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Nils Reimers. Cohere int8 & binary embeddings: Scale your vector database to large datasets. Cohere Blog, 2024. URL https://cohere.com/blog/int8-binary-embeddings.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Nils Reimers and Iryna Gurevych. The curse of dense low-dimensional information retrieval for large index sizes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 605–611. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-short.77.

Alireza Salemi and Hamed Zamani. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2395–2400, 2024. doi: 10.1145/3626772.3657957.

Sacha Servan-Schreiber, Simon Langowski, and Srinivas Devadas. Private approximate nearest neighbor search with sublinear communication. In *IEEE Symposium on Security and Privacy (S&P)*, 2022.

Snowflake. snowflake-arctic-embed-l-v2.0 model card. https://huggingface.co/Snowflake/snowflake-arctic-embed-l-v2.0, 2024. Accessed September 2026.

Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 377–390, 2020. doi: 10.1145/3372297.3417270.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

Louis Tremblay Thibault, Michael Walter, and Jiapeng Zhang. Practical SNARGs for matrix multiplications over encrypted data. Cryptology ePrint Archive, Paper 2026/027, 2026. URL https://eprint.iacr.org/2026/027.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550, 2020.

Baiqiang Wang, Qian Lou, Mengxin Zheng, and Dongfang Zhao. Pir-rag: A system for private information retrieval in retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2509.21325.

Hongyu Wang, Shuming Ma, Lingxiao Ma, Lei Wang, Wenhui Wang, Li Dong, Shaohan Huang, Huaijie Wang, Jilong Xue, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: 1-bit pre-training for large language models. *J. Mach. Learn. Res.*, 26(1), September 2026. ISSN 1532-4435.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Wolters Kluwer. Health system adoption of Wolters Kluwer UpToDate Expert AI now reaching approximately 2,500 U.S. hospitals and health systems, August 2026a. URL https://www.wolterskluwer.com/en/news/health-system-adoption-uptodate-expert-ai-reaching-2500-us-hospitals-health-system

Wolters Kluwer. What is UpToDate, 2026b. URL https://www.wolterskluwer.com/en/solutions/uptodate/about. Accessed September 2026.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*, 2023.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 979–986, 2021. doi: 10.18653/v1/2021.acl-short.123.

Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. Arctic-embed 2.0: Multilingual retrieval without compromise. *arXiv preprint arXiv:2412.04506*, 2024.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Jointly optimizing query encoder and product quantization to improve retrieval performance. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 2487–2496, 2021. doi: 10.1145/3459637.3482358.

Jinhao Zhu, Liana Patel, Matei Zaharia, and Raluca Ada Popa. Compass: Encrypted semantic search with high accuracy. Cryptology ePrint Archive, Paper 2024/1255, 2024. URL https://eprint.iacr.org/2024/1255.

# A Preliminaries

## A.1 Homomorphic Encryption

We rely on a linearly homomorphic encryption scheme based on the General Learning With Errors (GLWE) assumption.

## A.2 (G)LWE encryption

**Definition 2** (GLWE encryption scheme Lyubashevsky et al. (2013))**.** *Let $q$ be a prime ciphertext modulus, $p < q$ a plaintext modulus, $\kappa$ the GLWE dimension parameter and $d$ a power of two ring degree so that $\mathcal{R}_q := \mathbb{Z}[X]/\langle X^d + 1\rangle$ is a cyclotomic polynomial ring. We let $\chi_s$ and $\chi_e$ denote the secret key and error distributions over $\mathcal{R}_q$ respectively. Finally, we let $\mathcal{U}$ denote the uniform distribution, $\mathcal{R}_p$ denote the message space of the encryption scheme and $\Delta := \lfloor q/p \rceil$. A GLWE encryption scheme $\mathcal{E} := (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of the following algorithms:*

- $\mathsf{KGen}(1^\lambda)$*: Takes as input a security parameter $1^\lambda$ and returns a sampled secret key $\vec{s} \leftarrow\$ \chi_s^\kappa$.*
- $\mathsf{Enc}(\vec{s}, m)$*: Takes as input a secret key $\vec{s}$ and a message $m \in \mathcal{R}_p$ and returns a ciphertext $\vec{c} \in \mathcal{R}_q^{\kappa+1}$ computed as follows:*

$$\vec{a} \leftarrow\$ \mathcal{U}(\mathcal{R}_q)^\kappa, \quad e \leftarrow\$ \chi_e, \quad \vec{c} = (\vec{a}, \langle \vec{a}, \vec{s}\rangle + e + \Delta m) \in \mathcal{R}_q^{\kappa+1}.$$

- $\mathsf{Dec}(\vec{s}, \vec{c})$*: Takes as input a secret key* $\vec{s}$ *and a ciphertext* $\vec{c} = (\vec{a}, b) \in \mathcal{R}_q^{\kappa+1}$ *and returns a message* $m \in \mathcal{R}_p$ *computed as follows:*

$$m = \left\lfloor \frac{b - \langle \vec{a}, \vec{s} \rangle}{\Delta} \right\rceil$$

*This encryption scheme is secure under the GLWE assumption for appropriate choices of parameters. For a concrete parameter set, its security can be estimated using the widely used lattice estimator Albrecht et al. (2015).*

Table 5 gives the parameter sets we instantiate the protocol with, satisfying the correctness constraint of Theorem 2 with failure probability at most $2^{-64}$ per coefficient and at least $\lambda = 128$ bits of security for a ternary secret, as estimated with the lattice estimator Albrecht et al. (2015). Round 1 is shown at $b = 2$, the only bit-width the ring $d = 2^{10}$ admits, and at $b = 3$ and 4 with $d = 2^{11}$. The remaining rows of Table 2 follow the same pattern (rank $\kappa = 2$ at $d = 2^{10}$ has the same LWE dimension and the same security, but the implementation we measure supports rank 1 only). In each row $\sigma$ is set at the correctness ceiling of (1), so the security shown is the largest the row allows. Round 2 encrypts a one-hot vector over $m$ chunks with $B = p = 2^{24}$, so its ceiling is $\log_2 \sigma_{\max} = 64 - \log_2(2r) - \frac{1}{2}\log_2 m - 48$, i.e., $2^{5.3}$ at $m = 2^{13}$ and $2^{1.8}$ at $m = 2^{20}$. We use the minimal width $\sigma = 3.2$, which is correct up to $m = 2^{20}$ chunks and gives 223 bits at $d = 2^{12}$, the ring at which 128 bits are reached at that width; beyond $2^{20}$ chunks, each bit removed from $p$ buys four doublings of $m$.

Table 5: GLWE parameters per round: ring dimension $d$, rank $\kappa$, ciphertext modulus $q$, plaintext modulus $p$, noise standard deviation $\sigma$ and the resulting security parameter $\lambda$ (ternary secret) and maximum failure probability $\Pr[\mathsf{fail}]$ per coefficient. Round 2 is given for $m \leq 2^{20}$ chunks.

| Round | $d$ | $\kappa$ | $q$ | $p$ | $\sigma$ | $\lambda$ | $\Pr[\mathsf{fail}]$ |
|---|---|---|---|---|---|---|---|
| Round 1 ($b = 2$) | $2^{10}$ | 1 | $2^{64} - 2^{32} + 1$ | $2^{13}$ | $2^{40.8}$ | 138 | $2^{-64}$ |
| Round 1 ($b = 3$) | $2^{11}$ | 1 | $2^{64} - 2^{32} + 1$ | $2^{15}$ | $2^{37.8}$ | 253 | $2^{-64}$ |
| Round 1 ($b = 4$) | $2^{11}$ | 1 | $2^{64} - 2^{32} + 1$ | $2^{17}$ | $2^{34.8}$ | 227 | $2^{-64}$ |
| Round 2 | $2^{12}$ | 1 | $2^{64} - 2^{32} + 1$ | $2^{24}$ | $2^{1.7}$ | 223 | $2^{-64}$ |

The above encryption scheme is additively homomorphic and allows for multiplication by plaintext elements, provided the noise remains "small". More precisely, given two ciphertexts $\vec{c}_1 = (\vec{a}_1, b_1)$ and $\vec{c}_2 = (\vec{a}_2, b_2)$ encrypting messages $m_1, m_2 \in \mathcal{R}_p$ respectively, we have:

- Addition: $\vec{c}_{\text{add}} = \vec{c}_1 + \vec{c}_2 = (\vec{a}_1 + \vec{a}_2, b_1 + b_2)$ is a ciphertext encrypting $m_{\text{add}} = m_1 + m_2$.
- Scalar multiplication: For any plaintext $y \in \mathcal{R}_p$, $\vec{c}_{\text{scal}} = y \cdot \vec{c}_1 = (y \cdot \vec{a}_1, y \cdot b_1)$ is a ciphertext encrypting $m_{\text{scal}} = y \cdot m_1$.

These properties hold provided the accumulated noise in the resulting ciphertexts remains small enough to allow for correct decryption, i.e. $\|e\|_\infty \leq \Delta/2$.

### A.3 Proof of Theorem 2

*Proof.* Each score $s_i = \sum_j A_{ij} x_j$ satisfies $|s_i| \leq N(B-1)^2 < NB^2$, so the $2NB^2 - 1 \leq p$ values it can take are distinct modulo $p$. It remains to show that decryption returns $s_i \bmod p$. By linearity, $\mathbf{A} \cdot \mathbf{c}$ decrypts to $\mathbf{A} \cdot \mathbf{x} \bmod p$ provided the noise of every coefficient stays below $\Delta/2$ in absolute value. We neglect the rounding of $\Delta = \lfloor q/p \rfloor$, which shifts this threshold by at most $p/2$, and write $\Delta/2 = q/2p$. The ciphertext $\mathbf{c}$ carries $N$ independent noise coefficients $e_1, \ldots, e_N \sim \mathcal{N}(0, \sigma^2)$, and the coefficient of $\mathbf{A} \cdot \mathbf{c}$ carrying $s_i$ has noise $e'_i = \sum_j \pm A_{ij}\, e_{\pi(j)}$, for a permutation $\pi$ and signs fixed by the negacyclic ring structure. As such, $e'_i$ is Gaussian of variance $\sigma^2 \sum_j A_{ij}^2 \leq NB^2\sigma^2$, and the Gaussian tail bound gives $\Pr[|e'_i| > r\sqrt{N}B\sigma] \leq \varepsilon(r)$. The inequality $q/\sigma \geq 2r\sqrt{N}Bp$ implies that $r\sqrt{N}B\sigma \leq q/(2p)$, so every coordinate is correct except with probability $\varepsilon(r)$. □

### A.4 Secure Matrix-Vector Multiplication

We assume the existence of an efficient protocol for verifiable plaintext matrix–encrypted vector multiplication. Such a primitive can be instantiated with e.g. Tremblay Thibault et al. (2026). The protocol operates as follows:

1. The client encodes its input $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{Z}_p^N$ into ring elements and encrypts each component, obtaining a ciphertext vector $\mathbf{c}$ which is sent to the server.
2. The server holds a plaintext matrix $\mathbf{M} \in \mathbb{Z}_p^{m \times N}$. It computes the matrix-ciphertext product row-wise, yielding a result ciphertext vector $\mathbf{c}'$ of dimension $m$, which is returned to the client.
3. The client decrypts $\mathbf{c}'$ to obtain $\mathbf{M} \cdot \mathbf{x}$.

**Definition 3** (Passive security)**.** *The protocol is* passively secure *if the server's view (the ciphertext vector* $\mathbf{c}$*) reveals no information about* $\mathbf{x}$ *beyond public parameters and dimensions.*

**Definition 4** (Malicious security / Verifiability)**.** *The protocol is* maliciously secure *if, in addition to passive security, the client can verify that the server computed* $\mathbf{M} \cdot \mathbf{x}$ *correctly. That is, a malicious server cannot convince the client to accept an incorrect result* $\mathbf{y} \neq \mathbf{M} \cdot \mathbf{x}$ *except with probability* $\mathsf{negl}(\lambda)$.

The malicious security property is achieved via a verifiable computation mechanism which allows the client to check the server's computation without re-doing it.

### A.5 Embedding-Based Retrieval

Modern retrieval systems represent documents and queries as dense vectors (embeddings) in $\mathbb{R}^n$ using neural network encoders (Reimers & Gurevych (2019); Ni et al. (2022)). Given a corpus of $m$ documents, each document $i$ is mapped to an embedding $\mathbf{a}_i \in \mathbb{R}^n$. These embeddings are collected as rows of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. For a query with embedding $\mathbf{x} \in \mathbb{R}^n$, the relevance of each document is measured by the inner product $\langle \mathbf{a}_i, \mathbf{x} \rangle$, and the top-$k$ documents with highest scores are retrieved.

### A.6 Quantized Embeddings

To operate over $\mathbb{Z}_p$ rather than $\mathbb{R}$ we use *clipped symmetric scalar quantization*. Let $q_{\max} = 2^{b-1} - 1$ and let $\tau$ be the 99th percentile of $|\mathbf{A}|$ over all entries of the embedding matrix. Each coordinate $a$ is mapped to $\mathrm{clip}(\lfloor a\, q_{\max}/\tau \rceil, -q_{\max}, q_{\max})$, so the $1\%$ of coordinates beyond $\tau$ saturate. A query vector is quantized the same way with its own percentile. Every entry lies in $[-q_{\max}, q_{\max}]$.

The usual max-abs rule maps the single largest coordinate to $q_{\max}$. BERT-family encoders carry one coordinate holding 5 to $6\%$ of the squared norm of every embedding, so under max-abs that coordinate sets a step $4\times$ coarser than the bulk of the coordinates need. Clipping at a percentile is standard in post-training quantization (Banner et al., 2019). Table 10 compares clipping with max-abs on every encoder. The 99.9th and 95th percentiles land within a point of the 99th on both corpora.

**Compatibility with secure computation.** Quantized embeddings are natively integers in $\mathbb{Z}_p$, requiring no floating-point emulation, and the inner product of quantized vectors is computed exactly by the protocol. How closely it preserves the floating-point ranking is the subject of §4.1 and §4.3.

## B Protocol Message Flow

Figure 2 gives the message flow of the two-round protocol of §2.

## C Security of the Protocol

We sketch the proof of Theorem 1.

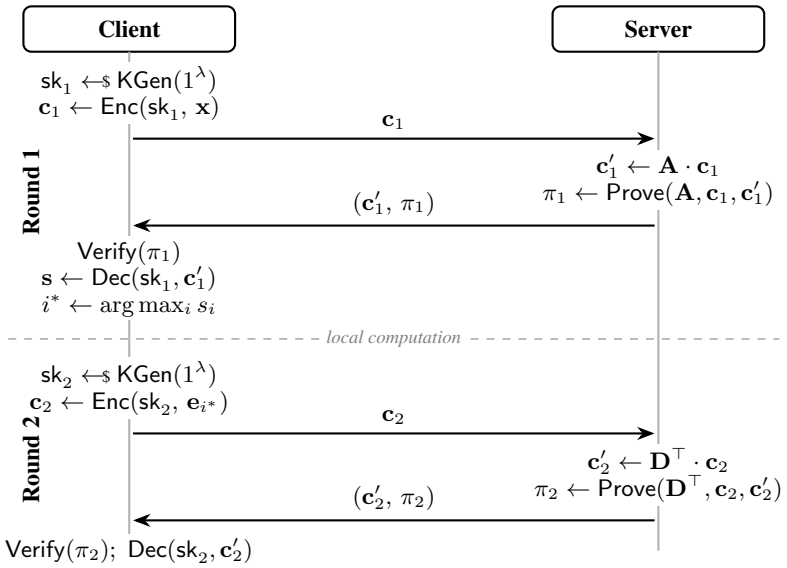



Figure 2: The private dense retrieval protocol. Round 1 retrieves encrypted similarity scores, Round 2 privately fetches the top document; both use the same verifiable matrix-vector sub-protocol, and the client verifies each proof before decrypting.

**Client privacy.** In each round the server observes only fresh ciphertexts, which by semantic security of the GLWE scheme are computationally indistinguishable from encryptions of any other vector of the same dimension. The two rounds use independent keys, so there is no cross-round linkage: the server's view in the two rounds is simulatable from public parameters alone. Because the client verifies each proof before decrypting, its observable behavior (continue or abort) is a function of the proof's validity only, which the server can compute itself. As such, the client's reactions leak nothing about the decrypted values, which rules out reaction attacks Chillotti et al. (2016).

**Integrity.** By soundness of the underlying protocol, any server deviation in round $i$, be it substituting the matrix, corrupting the ciphertext or fabricating the proof, causes $\pi_i$ to fail verification except with negligible probability:

$$\Pr\big[\mathsf{Verify}(\pi_i) = \mathsf{Accept} \ \wedge \ \mathbf{y}_i \neq \mathbf{M}_i \cdot \mathbf{x}_i\big] \leq \mathsf{negl}(\lambda),$$

where $(\mathbf{M}_1, \mathbf{x}_1) = (\mathbf{A}, \mathbf{x})$ and $(\mathbf{M}_2, \mathbf{x}_2) = (\mathbf{D}^\top, \mathbf{e}_{i^*})$. Each round is verified against its own commitment, so a union bound over the two rounds gives the claim for the composed protocol.

## D Batched Top-$k$ Retrieval

After Round 1, the client can identify the top-$k$ indices $i_1^*, \ldots, i_k^*$ and issue $k$ independent Round-2 queries. Using a single $k$-hot vector instead does not work: $\mathbf{D}^\top \cdot \sum_j \mathbf{e}_{i_j^*}$ yields the *sum* of the $k$ documents, from which the individual documents cannot be recovered. For moderate $k$, the $k$ queries batch into a single matrix–matrix product $\mathbf{D}^\top \cdot \mathbf{E}$, where $\mathbf{E} \in \{0,1\}^{m \times k}$ has one-hot columns.

**Why latency does not grow with $k$.** The verifiable matrix–vector multiplication of Tremblay Thibault et al. (2026) proves the relation $\mathbf{c}' = \mathbf{M} \cdot \mathbf{c}$ with a sum-check protocol compiled with a polynomial commitment scheme (PCS). Once per database, the server commits to a polynomial encoding of $\mathbf{M}$. To answer a query, it runs the sum-check, which reduces the claim $\mathbf{c}' = \mathbf{M} \cdot \mathbf{c}$ to the evaluation of the committed polynomial at a single random point fixed during the protocol, and then produces a PCS evaluation proof for that point. The evaluation proof is the dominant cost factor: it accounts for over $95\%$ of the server time.

Batching exploits the fact that the sum-check protocol is linear in the claim it proves. More precisely, given $k$ claims $\mathbf{c}'_j = \mathbf{M} \cdot \mathbf{c}_j$ about the same committed matrix, the verifier samples a random

challenge $\gamma$ and the prover proves the single claim $\sum_j \gamma^j \mathbf{c}'_j = \mathbf{M} \cdot \sum_j \gamma^j \mathbf{c}_j$ instead. If any of the $k$ claims is false, the combined claim is false as well except with probability at most $k/|\mathcal{C}|$ over the choice of $\gamma$, for $\mathcal{C}$ the challenge space. This additive soundness loss is negligible provided $|\mathcal{C}|$ is large enough. The batched claim is a matrix–vector relation of the same shape as a single query. As such, it is proved with one run of the sum-check and one PCS evaluation proof, regardless of $k$. The server still computes the $k$ products and the sum-check messages of the combined claim, whose cost grows with $k$, but the evaluation proof, which dominates, is generated once. The $k$ answers are all returned since the client needs each document, which is why communication grows by a factor $k$ whereas the proof is essentially of the same size as for a single query.

## E Readers, Prompts and Downstream Corpora

**Protocol description.** We evaluate every endpoint, encoder and condition $c \in \{\text{closed}, \text{gold}, \text{float}, 8, 6, 4, 3, 2\}$. The evaluation of §4.4 works as follows. (1) The corpus and the 300 queries are embedded once in floating point. For a bit-width $c = b$, both are quantized with the clipped quantizer of Appendix A.6. They are then scored in integers, as in Round 1, and the top-$k$ passages are kept. Ties are broken toward the lower index. The float condition uses the floating-point scores. The gold condition uses the labelled relevant passages, at most $k$ of them. The closed condition uses no passage. (2) We build one prompt from the template of the endpoint (below), with the passages in rank order. A prompt that is identical across conditions is answered once. (3) The reader answers with greedy decoding. Decoding is constrained to the three labels for claim verification and capped at 24 tokens for QA. (4) A QA answer is correct if the normalised text of any reference answer appears in the normalised response. A claim label is correct if it equals the human label. SciFact accuracy is computed over the 188 claims with evidence in the corpus. (5) We compare per-query correctness under each condition with the float condition on the same queries, using the tests described below. We record retrieval-side statistics alongside, namely top-$k$ overlap with float and whether a relevant passage is in the top $k$. The prompts and $k$ were fixed before any bit-width was run. Nothing is tuned on the test queries.

**Readers.** The body reports `phi-4` (14B, MIT licence) (Abdin et al., 2024). Table 6 adds `OLMo-2-7B-Instruct` (OLMo Team, 2025), whose training data is fully open, and the Qwen2.5 instruction-tuned models at 7B and 72B (Qwen Team, 2025). All are run locally with vLLM and greedy decoding. Under the one-word prompt, `OLMo-2` mostly abstains on claim verification. It answers NoInfo on 87% of SciFact prompts and never Contradict, so its absolute SciFact accuracy is low. It still reproduces the shape of the degradation curve, which is what we compare across readers.

**Prompts.** The claim-verification prompt states the claim and lists the retrieved abstracts. It asks for exactly one word from {Support, Contradict, NoInfo}. Decoding is constrained to those three strings. As such, there are no parse failures and no exclusions. The QA prompt states the question and lists the retrieved passages. It asks for the answer in a few words, or for exactly Unknown if the passages do not contain it. Responses are capped at 24 tokens. Prompts are fixed across bit-widths and encoders. Only the retrieved passages change. One SciFact $k$=5 prompt exceeds `OLMo-2`'s 4,096-token context and is excluded for that reader.

**BioASQ corpus.** We build the corpus from BioASQ-QA (Krithara et al., 2023) (`training11b`, 4,719 questions). For every question of every type, we group the evidence snippets by source document. We concatenate them into one passage per (question, document) pair. This gives 35,454 passages with a median length of 220 characters. Queries are a seeded sample of 300 factoid questions with a non-empty `exact_answer` and at most three gold passages. Of these, 190 have one gold passage, 54 have two and 56 have three. The cap keeps gold-in-top-$k$ from being trivially high. The relevant set of a question is its own passages. A response is correct if the normalised text of any reference answer synonym appears in it.

**NQ at scale.** The 300 Natural Questions test queries are the 100 of the original evaluation plus 200 more. These 200 are a seeded draw among the queries whose question matches an NQ-open answer list. Passages are retrieved from the full BEIR corpus of 2,681,468 Wikipedia passages. We use the same embeddings as the scale run of Appendix F.

**Statistics.** For every (reader, endpoint, encoder) and bit-width, we report the paired difference in accuracy from floating point, with a normal-approximation 95% interval on the per-query differences. Table 6 gives these differences at $b \in \{4, 3, 2\}$. We test for a drop with an exact McNemar test on the discordant pairs. A non-significant McNemar test is not evidence of equivalence. As such, we also run two one-sided tests (TOST) with equivalence margins of 5 and 3 points. A bit-width is declared equivalent to floating point when both one-sided tests reject at $p < 0.05$, i.e. when the 90% interval lies inside the margin. At $b$=4, all 32 combinations are equivalent within 5 points and 23 are equivalent within 3. At $b$=3, these counts are 30 and 17. At $b$=2, they are 9 and 2.

Table 6: Paired accuracy difference from floating point, in points with 95% interval, at $b$=4, 3 and 2. All readers, endpoints and encoders use the clipped quantizer ($n$=300 queries, 188 evidence-bearing claims for SciFact). $^*$: significant drop, exact McNemar $p < 0.05$. $^\dagger$: equivalent to floating point within $\pm 5$ points, TOST $p < 0.05$. "float" is the floating-point accuracy.

| Endpoint | Encoder | float | $b$=4 | $b$=3 | $b$=2 |
|---|---|---|---|---|---|
| *phi-4 (14B)* | | | | | |
| SciFact $k$=5 | BGE | 0.734 | $-1.1\ [-3.6, +1.5]^\dagger$ | $-2.7\ [-5.4, +0.1]^\dagger$ | $-5.9\ [-10.1, -1.6]^*$ |
| SciFact $k$=5 | arctic | 0.713 | $+0.5\ [-1.3, +2.3]^\dagger$ | $+0.0\ [-2.6, +2.6]^\dagger$ | $-1.6\ [-4.7, +1.5]^\dagger$ |
| SciFact $k$=1 | BGE | 0.766 | $+0.5\ [-0.5, +1.6]^\dagger$ | $-0.5\ [-2.3, +1.3]^\dagger$ | $-8.0\ [-12.1, -3.8]^*$ |
| SciFact $k$=1 | arctic | 0.739 | $+0.0\ [+0.0, +0.0]^\dagger$ | $+1.6\ [-0.2, +3.4]^\dagger$ | $-5.9\ [-10.1, -1.6]^*$ |
| BioASQ | BGE | 0.633 | $+0.3\ [-1.4, +2.1]^\dagger$ | $+1.0\ [-0.7, +2.7]^\dagger$ | $-2.0\ [-4.6, +0.6]^\dagger$ |
| BioASQ | arctic | 0.647 | $-1.0\ [-2.5, +0.5]^\dagger$ | $-0.7\ [-2.3, +0.9]^\dagger$ | $-4.0\ [-6.6, -1.4]^*$ |
| NQ (2.68M) | BGE | 0.643 | $+0.7\ [-2.1, +3.4]^\dagger$ | $-2.0\ [-5.3, +1.3]^\dagger$ | $-7.0\ [-11.5, -2.5]^*$ |
| NQ (2.68M) | arctic | 0.643 | $-1.7\ [-3.6, +0.3]^\dagger$ | $-0.3\ [-2.9, +2.2]^\dagger$ | $-4.3\ [-8.4, -0.3]$ |
| *OLMo-2 (7B)* | | | | | |
| SciFact $k$=5 | BGE | 0.213 | $-0.5\ [-3.3, +2.2]^\dagger$ | $-1.6\ [-3.9, +0.7]^\dagger$ | $-6.4\ [-10.7, -2.0]^*$ |
| SciFact $k$=5 | arctic | 0.198 | $-1.1\ [-3.2, +1.0]^\dagger$ | $-2.1\ [-4.7, +0.4]^\dagger$ | $-2.1\ [-5.4, +1.2]^\dagger$ |
| SciFact $k$=1 | BGE | 0.505 | $+1.1\ [-0.4, +2.5]^\dagger$ | $-1.1\ [-3.1, +1.0]^\dagger$ | $-4.8\ [-8.5, -1.1]^*$ |
| SciFact $k$=1 | arctic | 0.479 | $+0.0\ [-1.5, +1.5]^\dagger$ | $+0.5\ [-1.3, +2.3]^\dagger$ | $-3.7\ [-7.1, -0.3]$ |
| BioASQ | BGE | 0.503 | $+1.0\ [-1.5, +3.5]^\dagger$ | $+0.0\ [-3.1, +3.1]^\dagger$ | $-5.0\ [-9.3, -0.7]^*$ |
| BioASQ | arctic | 0.520 | $-0.3\ [-2.9, +2.2]^\dagger$ | $-1.3\ [-3.9, +1.3]^\dagger$ | $-2.7\ [-6.0, +0.7]$ |
| NQ (2.68M) | BGE | 0.497 | $+1.0\ [-2.1, +4.1]^\dagger$ | $-1.3\ [-4.9, +2.2]^\dagger$ | $-5.0\ [-9.5, -0.5]^*$ |
| NQ (2.68M) | arctic | 0.563 | $-0.3\ [-2.5, +1.8]^\dagger$ | $-3.3\ [-6.2, -0.4]^*$ | $-8.3\ [-12.2, -4.5]^*$ |
| *Qwen2.5-7B* | | | | | |
| SciFact $k$=5 | BGE | 0.681 | $-0.5\ [-3.7, +2.6]^\dagger$ | $-1.6\ [-4.7, +1.5]^\dagger$ | $-5.3\ [-9.9, -0.7]^*$ |
| SciFact $k$=5 | arctic | 0.654 | $-0.5\ [-2.3, +1.3]^\dagger$ | $+1.1\ [-1.5, +3.6]^\dagger$ | $+0.0\ [-3.3, +3.3]^\dagger$ |
| SciFact $k$=1 | BGE | 0.665 | $+0.5\ [-0.5, +1.6]^\dagger$ | $-1.1\ [-3.1, +1.0]^\dagger$ | $-9.0\ [-13.4, -4.7]^*$ |
| SciFact $k$=1 | arctic | 0.644 | $-0.5\ [-1.6, +0.5]^\dagger$ | $+0.0\ [-2.1, +2.1]^\dagger$ | $-5.9\ [-10.3, -1.4]^*$ |
| BioASQ | BGE | 0.477 | $-1.7\ [-4.0, +0.7]^\dagger$ | $-1.3\ [-3.9, +1.3]^\dagger$ | $-4.3\ [-7.8, -0.8]^*$ |
| BioASQ | arctic | 0.470 | $-0.3\ [-2.3, +1.6]^\dagger$ | $-1.0\ [-3.7, +1.7]^\dagger$ | $-1.7\ [-4.8, +1.5]^\dagger$ |
| NQ (2.68M) | BGE | 0.353 | $+1.7\ [-1.0, +4.4]^\dagger$ | $+1.7\ [-1.6, +4.9]^\dagger$ | $-1.3\ [-5.6, +2.9]^\dagger$ |
| NQ (2.68M) | arctic | 0.413 | $-1.7\ [-3.8, +0.5]^\dagger$ | $-2.3\ [-5.2, +0.5]^\dagger$ | $-5.3\ [-9.4, -1.2]^*$ |
| *Qwen2.5-72B* | | | | | |
| SciFact $k$=5 | BGE | 0.830 | $+0.5\ [-1.3, +2.3]^\dagger$ | $+0.5\ [-1.3, +2.3]^\dagger$ | $-2.7\ [-5.4, +0.1]^\dagger$ |
| SciFact $k$=5 | arctic | 0.819 | $+0.0\ [+0.0, +0.0]^\dagger$ | $+0.5\ [-1.3, +2.3]^\dagger$ | $-0.5\ [-3.3, +2.2]^\dagger$ |
| SciFact $k$=1 | BGE | 0.830 | $+0.5\ [-0.5, +1.6]^\dagger$ | $-1.1\ [-3.1, +1.0]^\dagger$ | $-9.0\ [-13.7, -4.4]^*$ |
| SciFact $k$=1 | arctic | 0.793 | $+0.0\ [+0.0, +0.0]^\dagger$ | $+0.0\ [-2.6, +2.6]^\dagger$ | $-7.4\ [-12.0, -2.9]^*$ |
| BioASQ | BGE | 0.580 | $+0.0\ [-0.9, +0.9]^\dagger$ | $+0.7\ [-0.9, +2.3]^\dagger$ | $-1.7\ [-4.5, +1.2]^\dagger$ |
| BioASQ | arctic | 0.597 | $+0.0\ [-1.3, +1.3]^\dagger$ | $-0.3\ [-1.5, +0.8]^\dagger$ | $-3.3\ [-6.1, -0.6]^*$ |
| NQ (2.68M) | BGE | 0.573 | $-1.7\ [-4.0, +0.7]^\dagger$ | $-2.7\ [-5.7, +0.4]$ | $-6.7\ [-10.5, -2.8]^*$ |
| NQ (2.68M) | arctic | 0.567 | $+0.3\ [-1.4, +2.1]^\dagger$ | $+0.0\ [-2.3, +2.3]^\dagger$ | $-5.0\ [-8.7, -1.3]^*$ |

**Chunked SciFact.** Table 7 gives the SciFact endpoint for `phi-4` when the reader receives the top-$k$ chunks of §4 instead of whole abstracts. Chunks are cut at sentence boundaries and hold at most 1,536 bytes including the title (`implem/scale/chunk_corpus.py`). The 5,183 abstracts give 7,510 chunks of 1,070 bytes on average. In floating point, a chunk of a gold abstract is in the top

5 for 79% of claims with BGE and 77% with arctic. For whole abstracts, these figures are 80 and 79%. NFCorpus, chunked the same way, gives 5,761 chunks, and 44% of its abstracts fit in one. In floating point, the reader does better on five chunks than on five abstracts, by 4.3 points with BGE and 2.1 with arctic. At $k$=1 it does about one point worse, since a single chunk may leave out the evidence sentence. Under quantization, the two encoders separate one bit earlier than on abstracts. The reason is that chunks of one abstract are near-duplicates, and quantization reorders them. Judged per document, the ranking survives exactly as on abstracts (Table 8). The reader, judged per chunk, achieves top-1 agreement at $b$=3 is 0.82 for BGE and 0.89 for arctic, against 0.87 and 0.93 per document (Appendix F). With BGE at $k$=5, the loss reaches 4.3 points at $b$=3 ($p = 0.008$). On abstracts, it was 2.7 points and not significant. With arctic at $k$=5, every bit-width down to $b$=2 stays within 5 points of floating point, as on abstracts. As such, chunking moves the operating point of the conventional encoder from $b$=3 to $b$=4. The quantization-aware encoder stays at $b$=2.

Table 7: SciFact with chunks against whole abstracts, reader `phi-4`. We give floating-point accuracy on the 188 evidence-bearing claims. We also give the paired difference from floating point at $b$=4, 3 and 2, in points with 95% interval. $^*$: significant drop, exact McNemar $p < 0.05$. $^\dagger$: equivalent within $\pm$5 points, TOST $p < 0.05$.

| Endpoint | Encoder | Unit | float | $b$=4 | $b$=3 | $b$=2 |
|---|---|---|---|---|---|---|
| SciFact $k$=5 | BGE | abstracts | 0.734 | $-1.1$ $[-3.6, +1.5]^\dagger$ | $-2.7$ $[-5.4, +0.1]^\dagger$ | $-5.9$ $[-10.1, -1.6]^*$ |
| | | chunks | 0.777 | $-2.7$ $[-5.0, -0.4]^\dagger$ | $-4.3$ $[-7.1, -1.4]^*$ | $-6.9$ $[-11.4, -2.5]^*$ |
| SciFact $k$=5 | arctic | abstracts | 0.713 | $+0.5$ $[-1.3, +2.3]^\dagger$ | $+0.0$ $[-2.6, +2.6]^\dagger$ | $-1.6$ $[-4.7, +1.5]^\dagger$ |
| | | chunks | 0.734 | $+1.6$ $[-0.7, +3.9]^\dagger$ | $+1.6$ $[-1.9, +5.1]^\dagger$ | $-1.1$ $[-5.2, +3.1]^\dagger$ |
| SciFact $k$=1 | BGE | abstracts | 0.766 | $+0.5$ $[-0.5, +1.6]^\dagger$ | $-0.5$ $[-2.3, +1.3]^\dagger$ | $-8.0$ $[-12.1, -3.8]^*$ |
| | | chunks | 0.755 | $-0.5$ $[-2.3, +1.3]^\dagger$ | $-2.1$ $[-4.7, +0.4]^\dagger$ | $-12.2$ $[-17.6, -6.9]^*$ |
| SciFact $k$=1 | arctic | abstracts | 0.739 | $+0.0$ $[+0.0, +0.0]^\dagger$ | $+1.6$ $[-0.2, +3.4]^\dagger$ | $-5.9$ $[-10.1, -1.6]^*$ |
| | | chunks | 0.729 | $+0.5$ $[-1.8, +2.9]^\dagger$ | $+0.0$ $[-3.0, +3.0]^\dagger$ | $-5.9$ $[-10.6, -1.1]^*$ |

## F The Encoder Suite and the Scale Run

In this appendix we give the retrieval-side numbers behind §4.1 and §4.3 for every encoder. We also compare them with the max-abs quantizer and explain why some encoders lose more quality than others.

**Encoders and prefixes.** Each encoder is run with the instruction prefix its model card prescribes. `bge-large-en-v1.5` and `mxbai-embed-large-v1` prepend "Represent this sentence for searching relevant passages: " to queries. `e5-large-v2` prepends "query: " and "passage: ". `arctic-embed-l-v2.0` prepends "query: ". `gte-large` uses no prefix, and Cohere takes an input type instead. We run `arctic-embed-l-v2.0` with a 512-token window instead of its native 8,192, to match the other encoders. Abstracts rarely exceed this window. The downstream evaluation of §4.4 uses the clipped quantizer and these prefixes.

**Retained quality under the clipped quantizer.** Table 9 gives floating-point nDCG@10 for every encoder and corpus. It also gives the fraction of it retained at $b$=4 and $b$=3, and top-1 agreement with the floating-point ranking at the same bit-widths. Table 10 gives the same quantities under max-abs scaling next to the clipped values.

**Why some encoders lose more than others.** Quantization adds a small rounding error to every score. A query keeps its top-1 document if the gap between its two best scores is larger than this error. Rounding to a step $s$ gives an error that is uniform on $[-s/2, s/2]$, whose variance is $s^2/12$. The error on a score therefore has standard deviation $\sqrt{(s_{\mathbf{A}}^2 + s_{\mathbf{x}}^2)/12}$, where $s_{\mathbf{A}}$ is the corpus step and $s_{\mathbf{x}}$ the query step. For each query, we divide the gap by this standard deviation. We then take the median over queries. Table 11 gives this ratio for every encoder at $b$=4. The larger the ratio, the more often the top-1 document survives. Over the 12 encoder-corpus pairs, the ratio and the measured top-1 agreement have Spearman correlation 0.94 at $b$=4 and 0.99 at $b$=3. The two encoders trained for low-bit readout have the largest ratios. BGE and mxbai come next, and e5 and gte have the

Table 8: Retrieval quality retained of our two-round protocol under clipped quantization, as the percentage of floating-point nDCG@10 capped at 100, for $b \in \{2, 3, 4, 8\}$ on SciFact and NFCorpus, ranked per chunk and scored per document.

| | SciFact | | | | | | NFCorpus | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | low-bit | | conventional | | | | low-bit | | conventional | | | |
| $b$ | arctic | Cohere | BGE | mxbai | e5 | gte | arctic | Cohere | BGE | mxbai | e5 | gte |
| 2 | 92.2 | 89.7 | 86.4 | 88.5 | 82.2 | 86.6 | 88.2 | 88.6 | 85.9 | 86.5 | 78.8 | 83.6 |
| 3 | 98.7 | 99.1 | 97.4 | 98.7 | 97.3 | 97.5 | 98.2 | 99.3 | 99.0 | 97.4 | 94.1 | 97.6 |
| **4** | 99.9 | 99.4 | 99.7 | 99.4 | 100.0 | 99.8 | 99.6 | 99.8 | 99.5 | 98.9 | 96.5 | 100.0 |
| 8 | 99.9 | 99.5 | 99.9 | 100.0 | 100.0 | 99.8 | 99.7 | 100.0 | 99.7 | 100.0 | 98.7 | 100.0 |

Table 9: The encoder suite under the clipped quantizer. Floating-point nDCG@10, the percentage of it retained at $b$=4 and $b$=3, and top-1 agreement with the floating-point ranking at the same bit-widths. ✓: trained for low-bit readout.

| | | SciFact | | | | | NFCorpus | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Encoder | low-bit | float | ret. 4 | ret. 3 | top-1 4 | top-1 3 | float | ret. 4 | ret. 3 | top-1 4 | top-1 3 |
| `arctic-embed-l-v2.0` | ✓ | 0.706 | 100.4 | 99.4 | 0.96 | 0.93 | 0.354 | 99.6 | 98.0 | 0.91 | 0.82 |
| `Cohere embed-v3` | ✓ | 0.718 | 99.6 | 98.7 | 0.96 | 0.91 | 0.387 | 99.5 | 98.9 | 0.86 | 0.78 |
| `bge-large-en-v1.5` | × | 0.746 | 99.4 | 97.1 | 0.94 | 0.87 | 0.381 | 99.5 | 98.6 | 0.83 | 0.71 |
| `mxbai-embed-large-v1` | × | 0.739 | 100.3 | 100.1 | 0.93 | 0.89 | 0.387 | 100.1 | 98.9 | 0.84 | 0.73 |
| `e5-large-v2` | × | 0.722 | 99.4 | 96.8 | 0.88 | 0.77 | 0.372 | 96.3 | 90.8 | 0.79 | 0.64 |
| `gte-large` | × | 0.743 | 99.7 | 97.8 | 0.94 | 0.81 | 0.383 | 100.0 | 95.2 | 0.81 | 0.62 |

smallest. The table also explains why clipping helps BGE and mxbai. Their largest coordinate holds over $5\%$ of the squared norm, against at most $1.9\%$ for the other encoders. Under max-abs scaling, this one coordinate sets the step. For BGE on SciFact, the step would be $0.305/7 \approx 0.044$ instead of the clipped $0.011$, i.e. about $4\times$ coarser.

**Chunk-level retrieval.** We first judge the chunked SciFact corpus per document, as in Table 8. Top-1 agreement at $b$=4 and $b$=3 is then $0.94$ and $0.87$ for BGE and $0.96$ and $0.92$ for arctic. These are the same values as on whole abstracts (Table 12). We then judge it per chunk, counting every chunk of a relevant abstract as relevant. Top-1 agreement is then $0.92$ and $0.82$ for BGE and $0.93$ and $0.89$ for arctic. The difference comes from chunks of one abstract trading places under rounding. The reader of §4.4 sees this reordering, but a document ranking does not.

**Quality at scale.** Tables 14 and 13 give the measurements behind the quality-at-scale paragraph of §4.1. They cover the five encoders of the suite that run locally, namely arctic, BGE, mxbai, e5 and gte. Cohere is hosted and was not run at this scale. The corpus is the BEIR Natural Questions passage collection (2,681,468 passages) with its 3,452 test queries. We draw subsets of $10^4$, $10^5$ and $10^6$ passages at random with a fixed seed. Each subset contains every passage relevant to any query. As such, nDCG is defined at every size. Embeddings are computed once per encoder on one H100 and stored in half precision. The clipped percentile is estimated from a random sample of 4,096 rows of each subset. Quantized scores are computed in single precision. This is exact for $b \leq 8$, since every partial sum is an integer below $2^{24}$. We also check the scores against 64-bit integer arithmetic. Ties are broken toward the lower index.

**Recall at the reader's $k$.** The sweep above scores at $k$=10. The readers of §4.4 consume five passages. As such, we repeated the sweep at $k$=5 and added $b$=2. Table 13 gives Recall@5 retained at every size. At $b$=4, four of the five encoders retain at least $98.8\%$ of floating-point Recall@5 at every size. The fifth, e5, retains $97.3\%$ at 2.68M. At $b$=3, arctic, BGE and mxbai retain at least $98.4\%$. At 2.68M, gte and e5 fall to $94.5$ and $92.0\%$. At $b$=2, the loss grows with the corpus. It goes from 1 to $4\%$ at $10^4$ to 10 to $23\%$ at 2.68M. The lead of arctic over each conventional encoder also grows, from $0.8$ to $2.7$ points at $10^4$ to $2.1$ to $12.8$ points at 2.68M. This is the retrieval-side counterpart of the downstream drop of §4.4. On the 300 NQ questions of that section, the top-5 set

Table 10: The quantizer lever. Each cell reads max-abs scaling → clipping at the 99th percentile. We give the percentage of floating-point nDCG@10 retained and top-1 agreement at $b$=4 and $b$=3. Nothing on the cryptographic side differs between the two columns of each pair.

| | $b$=4 | | $b$=3 | |
|---|---|---|---|---|
| Encoder | ret. (%) | top-1 | ret. (%) | top-1 |
| *SciFact* | | | | |
| `arctic-embed-l-v2.0` | 100.0 → 100.4 | 0.93 → 0.96 | 97.6 → 99.4 | 0.83 → 0.93 |
| `Cohere embed-v3` | 100.0 → 99.6 | 0.93 → 0.96 | 95.4 → 98.7 | 0.74 → 0.91 |
| `bge-large-en-v1.5` | 97.1 → 99.4 | 0.79 → 0.94 | 77.3 → 97.1 | 0.54 → 0.87 |
| `mxbai-embed-large-v1` | 99.5 → 100.3 | 0.81 → 0.93 | 82.3 → 100.1 | 0.57 → 0.89 |
| `e5-large-v2` | 96.3 → 99.4 | 0.79 → 0.88 | 82.4 → 96.8 | 0.56 → 0.77 |
| `gte-large` | 97.3 → 99.7 | 0.81 → 0.94 | 88.1 → 97.8 | 0.68 → 0.81 |
| *NFCorpus* | | | | |
| `arctic-embed-l-v2.0` | 98.5 → 99.6 | 0.83 → 0.91 | 96.2 → 98.0 | 0.69 → 0.82 |
| `Cohere embed-v3` | 98.5 → 99.5 | 0.83 → 0.86 | 96.4 → 98.9 | 0.61 → 0.78 |
| `bge-large-en-v1.5` | 96.0 → 99.5 | 0.67 → 0.83 | 73.4 → 98.6 | 0.35 → 0.71 |
| `mxbai-embed-large-v1` | 96.2 → 100.1 | 0.63 → 0.84 | 77.4 → 98.9 | 0.37 → 0.73 |
| `e5-large-v2` | 93.0 → 96.3 | 0.65 → 0.79 | 78.4 → 90.8 | 0.37 → 0.64 |
| `gte-large` | 96.3 → 100.0 | 0.62 → 0.81 | 85.0 → 95.2 | 0.42 → 0.62 |

Table 11: Score gap against rounding error at $b$=4, clipped quantizer. $\max|\mathbf{A}|$: largest corpus coordinate. $s_{\mathbf{A}}$: clipped step. top dim.: share of the squared norm in the largest coordinate. gap: median gap between the two best floating-point scores. gap/noise: median ratio of that gap to the rounding error. top-1: top-1 agreement with floating point.

| Encoder | Corpus | $\max\|\mathbf{A}\|$ | $s_{\mathbf{A}}$ | top dim. (%) | gap | gap/noise | top-1 |
|---|---|---|---|---|---|---|---|
| `arctic-embed-l-v2.0` | SciFact | 0.200 | 0.0122 | 0.5 | 0.0582 | 11.72 | 0.96 |
| `Cohere embed-v3` | SciFact | 0.221 | 0.0125 | 0.8 | 0.0445 | 8.83 | 0.96 |
| `bge-large-en-v1.5` | SciFact | 0.305 | 0.0110 | 5.5 | 0.0319 | 7.00 | 0.94 |
| `mxbai-embed-large-v1` | SciFact | 0.311 | 0.0111 | 5.4 | 0.0345 | 7.73 | 0.93 |
| `e5-large-v2` | SciFact | 0.170 | 0.0105 | 1.9 | 0.0149 | 3.48 | 0.88 |
| `gte-large` | SciFact | 0.172 | 0.0107 | 1.7 | 0.0189 | 4.38 | 0.94 |
| `arctic-embed-l-v2.0` | NFCorpus | 0.194 | 0.0123 | 0.5 | 0.0268 | 5.22 | 0.91 |
| `Cohere embed-v3` | NFCorpus | 0.216 | 0.0126 | 1.0 | 0.0211 | 4.19 | 0.86 |
| `bge-large-en-v1.5` | NFCorpus | 0.287 | 0.0109 | 5.7 | 0.0117 | 2.64 | 0.83 |
| `mxbai-embed-large-v1` | NFCorpus | 0.292 | 0.0110 | 5.8 | 0.0135 | 2.97 | 0.84 |
| `e5-large-v2` | NFCorpus | 0.165 | 0.0105 | 1.9 | 0.0063 | 1.50 | 0.79 |
| `gte-large` | NFCorpus | 0.175 | 0.0107 | 1.7 | 0.0073 | 1.67 | 0.81 |

of BGE contained a relevant passage for 70% of questions in floating point and 60% at $b$=2. This is 86% retained, against 86.1% measured here on all 3,452 queries.

Table 12: Table 8 on whole abstracts instead of chunks. This checks comparability against published document-level baselines. We give the percentage of floating-point nDCG@10 retained under clipped quantization for each bit-width $b$.

| | Trained for low-bit readout | | Conventional | | | |
|---|---|---|---|---|---|---|
| $b$ | arctic | Cohere | BGE | mxbai | e5 | gte |
| *SciFact* | | | | | | |
| 2 | 91.8 | 92.1 | 87.0 | 90.4 | 84.0 | 87.6 |
| 3 | 99.4 | 98.7 | 97.1 | 100.1 | 96.8 | 97.8 |
| **4** | 100.4 | 99.6 | 99.4 | 100.3 | 99.4 | 99.7 |
| 5 | 100.7 | 99.8 | 99.0 | 99.9 | 101.3 | 99.2 |
| 6 | 100.1 | 99.7 | 99.4 | 100.0 | 101.2 | 99.7 |
| 7 | 100.2 | 99.8 | 99.2 | 100.0 | 100.9 | 99.7 |
| 8 | 100.0 | 99.7 | 99.1 | 100.2 | 100.9 | 99.8 |
| *NFCorpus* | | | | | | |
| 2 | 87.2 | 88.7 | 83.2 | 87.4 | 75.0 | 82.7 |
| 3 | 98.0 | 98.9 | 98.6 | 98.9 | 90.8 | 95.2 |
| **4** | 99.6 | 99.5 | 99.5 | 100.1 | 96.3 | 100.0 |
| 5 | 99.2 | 99.8 | 100.6 | 100.7 | 97.8 | 98.7 |
| 6 | 99.9 | 99.7 | 100.4 | 101.0 | 98.5 | 99.8 |
| 7 | 99.9 | 99.8 | 100.5 | 100.7 | 98.6 | 99.7 |
| 8 | 99.6 | 99.7 | 100.4 | 100.8 | 98.5 | 99.7 |

Table 13: Recall@5 retained under the clipped quantizer versus corpus size on BEIR Natural Questions (3,452 queries), at $b \in \{2, 3, 4\}$. Floating-point Recall@5 is given for reference. Generated by `implem/analysis/scale_recall5_table.py`.

| | | | Recall@5 retained (%) | | |
|---|---|---|---|---|---|
| Encoder | $m$ | Recall@5 (float) | $b$=2 | $b$=3 | $b$=4 |
| `arctic-embed-l-v2.0` | $10^4$ | 0.988 | 99.2 | 99.9 | 100.0 |
| | $10^5$ | 0.967 | 97.0 | 99.9 | 99.9 |
| | $10^6$ | 0.848 | 93.1 | 99.5 | 99.7 |
| | $2.68{\cdot}10^6$ | 0.737 | 89.9 | 99.0 | 99.8 |
| `bge-large-en-v1.5` | $10^4$ | 0.982 | 97.7 | 99.9 | 100.1 |
| | $10^5$ | 0.944 | 94.7 | 99.2 | 99.5 |
| | $10^6$ | 0.794 | 88.3 | 98.7 | 98.8 |
| | $2.68{\cdot}10^6$ | 0.655 | 86.1 | 98.5 | 99.7 |
| `mxbai-embed-large-v1` | $10^4$ | 0.983 | 98.4 | 99.9 | 100.1 |
| | $10^5$ | 0.944 | 96.3 | 99.6 | 99.9 |
| | $10^6$ | 0.800 | 90.4 | 98.8 | 99.4 |
| | $2.68{\cdot}10^6$ | 0.661 | 87.7 | 98.4 | 100.6 |
| `e5-large-v2` | $10^4$ | 0.988 | 96.5 | 99.5 | 99.8 |
| | $10^5$ | 0.961 | 92.1 | 98.5 | 99.5 |
| | $10^6$ | 0.847 | 82.7 | 95.8 | 98.3 |
| | $2.68{\cdot}10^6$ | 0.745 | 77.1 | 92.0 | 97.3 |
| `gte-large` | $10^4$ | 0.986 | 97.2 | 99.7 | 99.8 |
| | $10^5$ | 0.945 | 94.4 | 99.4 | 100.0 |
| | $10^6$ | 0.795 | 85.9 | 96.3 | 99.9 |
| | $2.68{\cdot}10^6$ | 0.658 | 82.6 | 94.5 | 99.6 |

Table 14: Quality versus corpus size on BEIR Natural Questions with the clipped quantizer. We give nDCG@10 retained at $b$=3 and $b$=4. We also give the smallest $b$ that retains 99% of floating-point nDCG@10, or $> 8$ when no $b$ up to 8 does. Generated by `implem/analysis/scale_analysis.py`.

| | | nDCG@10 ret. (%) | | required $b$ |
|---|---|---|---|---|
| Encoder | $m$ | $b$=3 | $b$=4 | (top-$k$) |
| `arctic-embed-l-v2.0` | $10^4$ | 100.0 | 99.9 | 3 |
| | $10^5$ | 99.9 | 100.1 | 3 |
| | $10^6$ | 99.1 | 99.9 | 3 |
| | $2.68{\cdot}10^6$ | 98.5 | 99.7 | 4 |
| `bge-large-en-v1.5` | $10^4$ | 99.5 | 99.9 | 3 |
| | $10^5$ | 99.0 | 99.6 | 3 |
| | $10^6$ | 98.0 | 99.1 | 4 |
| | $2.68{\cdot}10^6$ | 98.1 | 99.3 | 4 |
| `mxbai-embed-large-v1` | $10^4$ | 99.8 | 100.0 | 3 |
| | $10^5$ | 99.5 | 100.0 | 3 |
| | $10^6$ | 98.9 | 100.1 | 4 |
| | $2.68{\cdot}10^6$ | 98.3 | 99.9 | 4 |
| `e5-large-v2` | $10^4$ | 99.0 | 99.7 | 3 |
| | $10^5$ | 97.7 | 99.3 | 4 |
| | $10^6$ | 93.9 | 98.1 | 8 |
| | $2.68{\cdot}10^6$ | 91.9 | 97.4 | $> 8$ |
| `gte-large` | $10^4$ | 99.4 | 99.9 | 3 |
| | $10^5$ | 98.4 | 99.9 | 4 |
| | $10^6$ | 95.7 | 99.9 | 4 |
| | $2.68{\cdot}10^6$ | 94.0 | 100.6 | 4 |