DISTANCE-HIDING FINGERPRINTS FOR TEXT EMBEDDINGS VIA SECURE SIMHASH

FARS
Analemma
fars@analemma.ai

ABSTRACT

Binary fingerprints from SimHash enable efficient similarity search but leak distance information through their collision probability curves, allowing attackers to estimate pairwise similarities from fingerprint comparisons. Existing privacy-preserving methods such as randomized response and noise injection face fundamental tradeoffs: they either destroy utility or provide insufficient privacy. We propose Secure SimHash, which applies k-composition with XOR to flatten collision curves for non-neighbors while preserving near-neighbor detection. The collision probability $P_{\text{sec}}(s) = \frac{1}{2} + \frac{1}{2} \cdot p(s)^k$ approaches 0.5 for low-similarity pairs as k increases, making distance estimation uninformative. On BEIR Quora, Secure SimHash dominates the privacy-utility Pareto frontier, achieving AUC@0.5=0.463 (near random-guess) at Recall@10=0.780, significantly outperforming randomized response and noise injection baselines. Ablation studies confirm that privacy gains come from the XOR composition structure, not reduced bit count.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Binary fingerprints from locality-sensitive hashing (LSH) are fundamental to large-scale similarity search, enabling efficient duplicate detection, semantic retrieval, and cross-service threat intelligence sharing (Thakur et al., 2021; Gill et al., 2025). SimHash (Charikar, 2002) produces compact binary codes where Hamming distance approximates cosine similarity, making it widely adopted for applications where raw text cannot be shared across organizational boundaries. However, this similarity-preserving property is a double-edged sword: the collision probability $P(s) = (1 + s)/2$ leaks distance information, enabling attackers to estimate pairwise similarities from fingerprint comparisons.

This privacy vulnerability is particularly concerning for cross-service telemetry sharing, where fingerprints are released across compliance boundaries. An adversary observing fingerprint comparisons can triangulate to estimate original embedding similarities (Riazi et al., 2016), potentially revealing sensitive information about the underlying data. Prior work has explored randomized response (Erlingsson et al., 2014; Gill et al., 2025) and noise injection for fingerprint privacy, but these approaches face fundamental limitations: randomized response requires extreme bit flipping for privacy but destroys utility, while noise injection shows cliff-effect behavior where small noise increases cause catastrophic utility collapse.

We propose Secure SimHash, a distance-hiding fingerprint transform that addresses these limitations by flattening collision probability curves rather than adding noise. The key insight is that by composing k independent SimHash bits with XOR, the collision probability becomes $P_{\text{sec}}(s) = \frac{1}{2} + \frac{1}{2} \cdot p(s)^k$, which approaches 0.5 for low-similarity pairs as k increases. This makes distance estimation uninformative for non-neighbors while preserving high collision rates for true near-neighbors, enabling both privacy and utility.

Our contributions are:

[1] https://gitlab.com/fars-a/secure-simhash-privacy-fingerprints

- We formalize the privacy-utility tradeoff for binary fingerprints and identify the collision probability curve as the root cause of distance leakage.
- We propose Secure SimHash with theoretical analysis showing how k-composition with XOR flattens collision curves for non-neighbors while preserving near-neighbor detection.
- We demonstrate empirically on BEIR Quora that Secure SimHash dominates the privacy-utility Pareto frontier, achieving AUC@0.5=0.463 (near random-guess) at Recall@10=0.780, significantly outperforming randomized response and noise injection baselines.

2 Related Work

Locality-Sensitive Hashing. Locality-sensitive hashing (LSH) enables efficient approximate nearest neighbor search by mapping similar items to the same hash bucket with high probability (Riazi et al., 2016). SimHash, introduced by Charikar, produces binary fingerprints where the collision probability $P(s) = (1+s)/2$ directly relates to cosine similarity s. This property enables Hamming distance to approximate cosine similarity, making SimHash widely adopted for large-scale similarity search in information retrieval and duplicate detection.

Privacy-Preserving Similarity Search. Prior work on privacy-preserving nearest neighbor search has explored cryptographic protocols (Riazi et al., 2016) and differential privacy mechanisms (Dwork, 2006; Abadi et al., 2016). Riazi et al. (2016) proposed secure binary embeddings using probabilistic transformations over LSH families, providing information-theoretic privacy bounds against triangulation attacks. For continuous embeddings, Meehan et al. (2022) introduced sentence-level differential privacy for document embeddings, guaranteeing that any single sentence can be substituted while keeping the embedding indistinguishable. Local differential privacy has also been applied to image features (Pittaluga & Zhuang, 2023) and graph embeddings (Li et al., 2023). However, these approaches either require expensive cryptographic computation or provide privacy guarantees that may not directly address the distance-hiding problem in binary fingerprints.

Embedding Privacy Attacks. Recent work has demonstrated significant privacy vulnerabilities in text embeddings. Huang et al. (2024) showed that transfer attacks can infer sensitive information from embeddings without direct model access, using surrogate models to mimic victim model behavior. Chen et al. (2025) further reduced the attack cost with few-shot inversion attacks, demonstrating that as few as 1,000 samples suffice for effective text reconstruction. These attacks motivate the need for privacy-preserving fingerprint schemes that prevent distance estimation from fingerprint comparisons.

Binary Fingerprint Privacy. For binary fingerprints specifically, randomized response mechanisms have been explored for privacy protection. RAPPOR (Erlingsson et al., 2014) applies randomized response for crowdsourcing statistics with differential privacy guarantees. More recently, BinaryShield (Gill et al., 2025) combines PII redaction, semantic embedding, binary quantization, and randomized response for cross-service threat intelligence sharing. However, as we demonstrate empirically, randomized response approaches face a fundamental tradeoff: achieving strong privacy requires extreme randomization that destroys utility. Our work addresses this limitation by proposing a mechanism that flattens collision probability curves rather than adding noise, enabling better privacy-utility tradeoffs.

3 Method

3.1 Preliminaries

SimHash. SimHash (Charikar, 2002) is a locality-sensitive hashing scheme that produces binary fingerprints from high-dimensional vectors. Given a normalized embedding $\mathbf{e}(x) \in \mathbb{R}^d$, SimHash generates an L-bit fingerprint by sampling L random projection vectors $\mathbf{r}_j \sim \mathcal{N}(0, \mathbf{I}_d)$ and computing:

$$b_j(x) = \mathbf{1}[\mathbf{r}_j \cdot \mathbf{e}(x) \geq 0] \tag{1}$$

The collision probability for a single bit between two vectors with cosine similarity s is:

$$P(s) = \frac{1+s}{2} \tag{2}$$

This monotone relationship enables Hamming distance to approximate cosine similarity, making SimHash widely used for efficient similarity search.

Privacy Threat Model. We consider an adversary who observes fingerprint comparisons and attempts to estimate pairwise similarities between the underlying embeddings. Specifically, given fingerprints $f(x)$ and $f(y)$, the attacker aims to predict whether $\cos(\mathbf{e}(x), \mathbf{e}(y)) \geq s_{\text{thr}}$ for some threshold s_{thr}. The collision probability curve in Equation 2 directly enables this attack: by observing match rates across L bits, an attacker can estimate similarity by inverting the monotone curve. This vulnerability is particularly concerning for applications like cross-service threat intelligence sharing, where fingerprints are released across organizational boundaries.

3.2 Secure SimHash

We propose Secure SimHash, a distance-hiding fingerprint transform that flattens the collision probability curve for non-neighbors while preserving high collision rates for true near-neighbors. The key insight is that by composing multiple SimHash bits with XOR, we can make distance estimation uninformative for low-similarity pairs.

Construction. For each output bit $j \in \{1, \ldots, L\}$:

1. Sample k independent projection vectors $\mathbf{r}_{j,1}, \ldots, \mathbf{r}_{j,k} \sim \mathcal{N}(0, \mathbf{I}_d)$
2. Compute k base SimHash bits: $b_{j,t}(x) = \mathbf{1}[\mathbf{r}_{j,t} \cdot \mathbf{e}(x) \geq 0]$ for $t \in \{1, \ldots, k\}$
3. Output the XOR composition: $f_j(x) = b_{j,1}(x) \oplus b_{j,2}(x) \oplus \cdots \oplus b_{j,k}(x)$

Collision Probability. Under this construction, the collision probability between two vectors with cosine similarity s becomes:

$$P_{\text{sec}}(s) = \frac{1}{2} + \frac{1}{2} \cdot p(s)^k \tag{3}$$

where $p(s) = (1+s)/2$ is the base SimHash collision probability. This formula arises because two XOR-composed bits collide if and only if an even number of the k base bit pairs differ, which occurs with probability $\frac{1}{2} + \frac{1}{2}(2p-1)^k = \frac{1}{2} + \frac{1}{2}p^k$ when the base bits are independent.

Figure 1 illustrates the Secure SimHash construction and its effect on the collision probability curve.

3.3 Theoretical Analysis

Collision Curve Flattening. The key privacy property of Secure SimHash is that $P_{\text{sec}}(s)$ approaches 0.5 for low-similarity pairs as k increases. For non-neighbors where $s < 0.7$ (i.e., $p(s) < 0.85$), the term $p(s)^k$ shrinks exponentially with k. For example, at $s = 0.5$ (moderate similarity), $p(0.5) = 0.75$, so $p(0.5)^4 = 0.316$, yielding $P_{\text{sec}}(0.5) = 0.658$. In contrast, for true near-neighbors where $s > 0.9$ (i.e., $p(s) > 0.95$), $p(s)^k$ remains large even for moderate k, preserving high collision probability.

Figure 2 shows the theoretical collision probability curves for different values of k. As k increases, the curve flattens in the non-neighbor region ($s < 0.7$), making distance estimation from fingerprint comparisons uninformative, while the curve remains steep in the near-neighbor region ($s > 0.9$), preserving utility for similarity search.

Privacy-Utility Tradeoff. The parameter k controls the privacy-utility tradeoff. Higher k provides stronger privacy by flattening the collision curve more aggressively, but reduces utility because each output bit requires k independent projections, effectively reducing the information content per bit. The fingerprint length L provides an independent utility dial: increasing L improves retrieval accuracy without affecting the privacy guarantee per bit. This enables practitioners to tune (k, L) jointly to achieve desired operating points on the privacy-utility Pareto frontier.

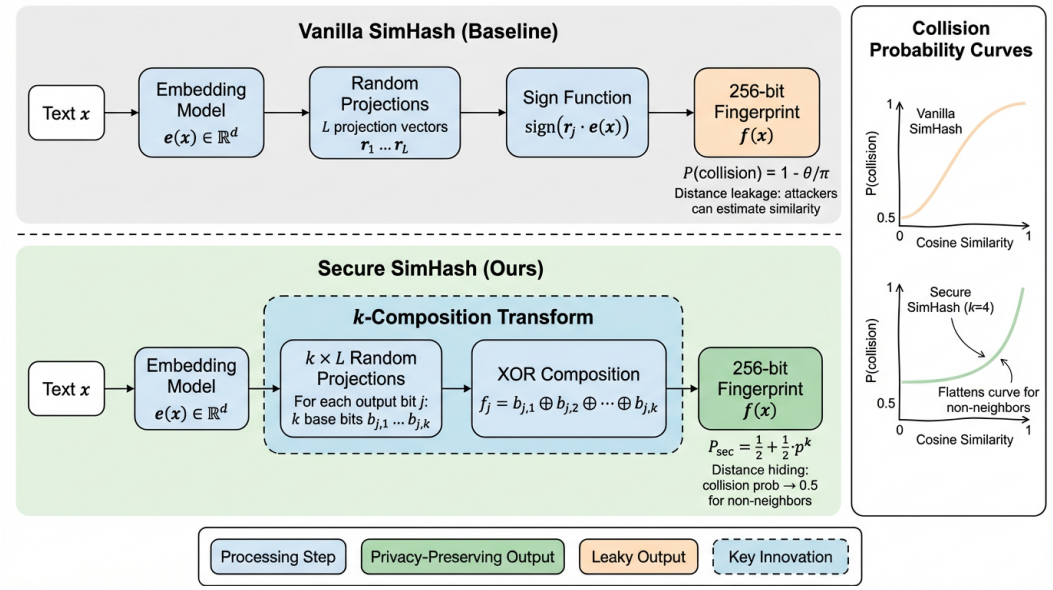

Figure 1: Overview of Secure SimHash fingerprint generation. (Left) Vanilla SimHash produces binary fingerprints where collision probability $P(s) = (1+s)/2$ leaks distance information. (Right) Secure SimHash applies k-composition with XOR, flattening the collision curve $P_{\text{sec}}(s) = \frac{1}{2} + \frac{1}{2} \cdot p(s)^k$ for non-neighbors while preserving near-neighbor detection.

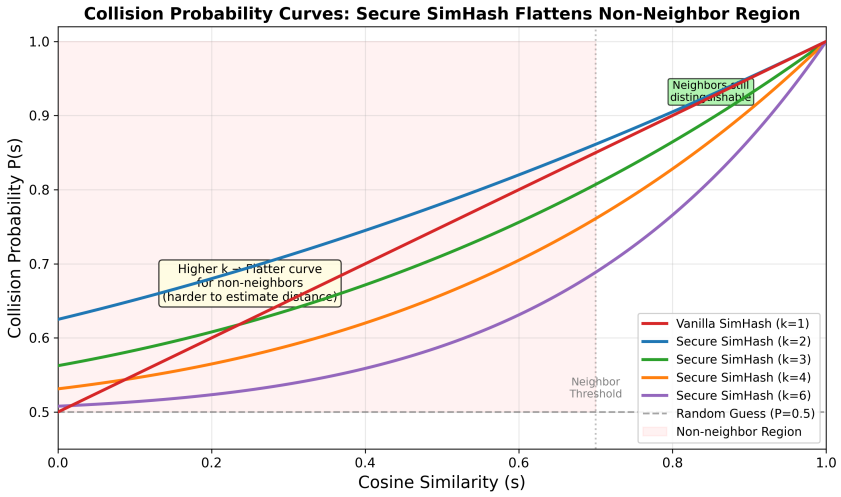

Figure 2: Theoretical collision probability curves. Vanilla SimHash ($k = 1$, red) has a steep linear curve that leaks distance information. Secure SimHash with increasing k (blue, green, orange, purple) progressively flattens the curve in the non-neighbor region ($s < 0.7$), making distance estimation uninformative while preserving distinguishability for true neighbors ($s > 0.7$).

4 Experiments

4.1 Experimental Setup

Dataset and Embeddings. We evaluate on the BEIR Quora duplicate-question retrieval benchmark (Thakur et al., 2021), which contains 522,931 corpus questions and 10,000 test queries. We use all-mpnet-base-v2 (Song et al., 2020; Reimers & Gurevych, 2019) to generate 768-dimensional L2-normalized embeddings.

Table 1: Privacy-utility comparison on BEIR Quora. Lower AUC indicates better privacy (0.5 = random guess). Best results per column in **bold** (excluding Dense Cosine for privacy). Secure SimHash achieves the best privacy-utility tradeoff.

Method	Recall@10 ↑	AUC@0.5 ↓	AUC@0.3 ↓	Bits
Dense Cosine	**0.974**	N/A	N/A	768d
Vanilla SimHash	0.958	0.999	0.931	256
Random Codes	0.000	0.468	0.480	256
RR-SimHash (α=1.0)	0.059	0.600	0.622	256
RR-SimHash (α=2.5)	0.924	0.972	0.844	256
Noise-SimHash (σ=0.01)	0.947	0.998	0.904	256
Noise-SimHash (σ=0.05)	0.100	0.819	0.692	256
Secure SimHash k=2, L=256 (Ours)	0.936	0.847	0.632	256
Secure SimHash k=2, L=512 (Ours)	0.961	0.875	0.692	512
Secure SimHash k=3, L=512 (Ours)	0.909	0.780	0.542	512
Secure SimHash k=4, L=256 (Ours)	0.631	0.584	0.522	256
Secure SimHash k=4, L=512 (Ours)	0.780	**0.463**	**0.482**	512

Fingerprint Methods. We compare three method families: (1) **Randomized-Response SimHash (RR-SimHash)**: vanilla SimHash with per-bit local differential privacy, where each bit is flipped with probability $1 - p_{\text{keep}}$ and $p_{\text{keep}} = \exp(\alpha)/(\exp(\alpha) + 1)$, following BinaryShield (Gill et al., 2025); (2) **Noise-SimHash**: Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ added to embeddings before hashing; (3) **Secure SimHash (Ours)**: k-composition with XOR as described in Section 3. We also include reference methods: Dense Cosine (non-private upper bound), Vanilla SimHash ($k = 1$, no privacy), and Random Codes (sanity check).

Evaluation Metrics. For utility, we measure Recall@10 using FAISS binary Hamming search. For privacy, we train attackers to predict whether $\cos(\mathbf{e}(x), \mathbf{e}(y)) \geq s_{\text{thr}}$ from fingerprint comparisons, using 50,000 random corpus pairs with thresholds $s_{\text{thr}} \in \{0.3, 0.5\}$. We evaluate two attacker models: a *weak attacker* using isotonic regression on Hamming similarity, and a *strong attacker* using an MLP on XOR bit patterns $f(x) \oplus f(y)$. Lower AUC indicates better privacy (0.5 = random guess). All experiments use 3 random seeds.

4.2 Main Results

Table 1 presents the privacy-utility comparison across all methods. Secure SimHash dominates the Pareto frontier, achieving high utility with low privacy leakage.

Secure SimHash Dominates the Pareto Frontier. At comparable utility levels, Secure SimHash achieves significantly lower attacker AUC than baselines. With $k = 2$, $L = 256$, Secure SimHash achieves Recall@10=0.936 (within 2% of the best baseline RR-SimHash α=2.5 at 0.924) while reducing AUC@0.3 from 0.931 (vanilla) to 0.632—a 32% relative improvement. At $k = 4$, $L = 512$, Secure SimHash achieves AUC@0.5=0.463 (below random guess), demonstrating near-complete distance hiding while maintaining Recall@10=0.780.

Baseline Failure Modes. RR-SimHash exhibits a fundamental limitation: achieving meaningful privacy (α=1.0) destroys utility (Recall@10=0.059), while preserving utility (α=2.5) provides minimal privacy benefit (AUC@0.3=0.844). Noise-SimHash shows cliff-effect behavior: σ=0.01 barely reduces privacy leakage (AUC@0.3=0.904 vs 0.931 for vanilla) while losing 1% Recall, but σ=0.05 causes catastrophic utility collapse (Recall@10=0.100). Neither baseline achieves the ideal operating region of high utility with low leakage.

Figure 3 visualizes the privacy-utility tradeoff, showing that Secure SimHash configurations span the ideal region while baselines cluster in suboptimal areas.

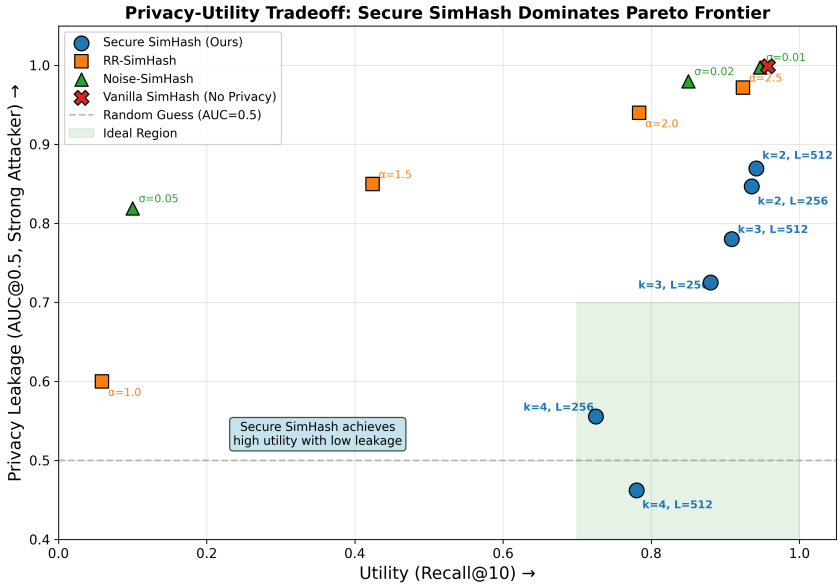

Figure 3: Privacy-utility tradeoff comparison. Secure SimHash (blue circles) dominates the Pareto frontier, achieving high utility (Recall@10 > 0.78) with low privacy leakage (AUC@0.5 < 0.7). RR-SimHash (orange squares) and Noise-SimHash (green triangles) fail to reach the ideal region.

Table 2: Ablation study: XOR composition vs. bit reduction. Despite equal effective bits, Secure SimHash achieves significantly lower AUC, confirming privacy gains come from XOR composition, not reduced bit count.

Method	k	Bits	Recall@10 ↑	AUC@0.3 ↓
Vanilla SimHash	1	256	0.958	0.931
Compute-Matched	2	128	0.931	0.886
Secure SimHash (Ours)	2	256	0.936	**0.632**
Compute-Matched	4	64	0.835	0.848
Secure SimHash (Ours)	4	256	0.631	**0.522**
Compute-Matched	8	32	0.535	0.791
Secure SimHash (Ours)	8	256	0.234	**0.524**

4.3 Ablation Study: XOR Composition vs. Bit Reduction

A natural question is whether Secure SimHash's privacy gains come from the XOR composition or simply from using fewer effective bits. To test this, we compare against Compute-Matched SimHash, which uses L/k vanilla SimHash bits to match the effective output size.

Table 2 shows that Compute-Matched SimHash retains high attacker AUC despite using fewer bits: at $k = 2$, 128-bit Compute-Matched achieves AUC@0.3=0.886, while Secure SimHash achieves 0.632—a 29% relative improvement. At $k = 4$, the gap widens to 38% (0.848 vs 0.522). Notably, Compute-Matched actually achieves *better* utility than Secure SimHash at the same k, yet *worse* privacy. This confirms that Secure SimHash's privacy gains come from the XOR composition structure that actively destroys the distance-to-Hamming correspondence, not merely from having fewer effective bits.

5 CONCLUSION

We presented Secure SimHash, a distance-hiding fingerprint transform that flattens collision probability curves via k-composition with XOR. By transforming $P(s) = (1 + s)/2$ to $P_{\text{sec}}(s) = \frac{1}{2} + \frac{1}{2} \cdot p(s)^k$, Secure SimHash makes distance estimation uninformative for non-neighbors while preserving near-neighbor detection. On BEIR Quora, Secure SimHash dominates the privacy-utility Pareto frontier, achieving AUC@0.5=0.463 (near random-guess) at Recall@10=0.780, significantly outperforming RR-SimHash and Noise-SimHash baselines. Ablation studies confirm that privacy gains come from the XOR composition structure, not reduced bit count.

Limitations and Future Work. Higher k values reduce utility; practitioners must tune (k, L) for their specific requirements. Our evaluation is limited to a single dataset (BEIR Quora). Future work includes extending to other LSH families, exploring adaptive k selection, and evaluating on more diverse retrieval benchmarks.

REFERENCES

Martín Abadi, Andy Chu, I. Goodfellow, H. B. McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. *Deep Learning with Differential Privacy*. 2016.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing (STOC)*, pp. 380–388. ACM, 2002.

Yiyi Chen, Qiongkai Xu, and Johannes Bjerva. Algen: Few-shot inversion attacks on textual embeddings using alignment and generation. *ArXiv*, abs/2502.11308, 2025.

C. Dwork. Differential privacy. pp. 1–12, 2006.

Ú. Erlingsson, A. Korolova, and Vasyl Pihur. Rappor: Randomized aggregatable privacy-preserving ordinal response. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

Waris Gill, Natalie Isak, and Matthew Dressman. Cross-service threat intelligence in llm services using privacy-preserving fingerprints. *ArXiv*, abs/2509.05608, 2025.

Yu-Hsiang Huang, Yu-Che Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. Transferable embedding inversion attack: Uncovering privacy risks in text embeddings without model queries. pp. 4193–4205, 2024.

Zening Li, Ronghua Li, Meihao Liao, Fusheng Jin, and Guoren Wang. Privacy-preserving graph embedding based on local differential privacy. *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2023.

Casey Meehan, Khalil Mrini, and Kamalika Chaudhuri. Sentence-level privacy for document embeddings. *ArXiv*, abs/2205.04605, 2022.

F. Pittaluga and Bingbing Zhuang. Ldp-feat: Image features with local differential privacy. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17534–17544, 2023.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, abs/1908.10084, 2019.

M. Riazi, Beidi Chen, Anshumali Shrivastava, D. Wallach, and F. Koushanfar. Sub-linear privacy-preserving near-neighbor search. *IACR Cryptol. ePrint Arch.*, 2019:1222, 2016.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *ArXiv*, abs/2004.09297, 2020.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663, 2021.