Locality Sensitive Hashing in Fourier Frequency Domain For Soft Set Containment Search

Indradyumna Roy[†] **Rishi Agarwal**[†]
Soumen Chakrabarti[†] **Anirban Dasgupta**[◇] **Abir De**[†]
[†]IIT Bombay, [◇]IIT Gandhinagar
{indraroy15, rishiagarwal18, soumen, abir}@cse.iitb.ac.in
anirbandg@cse.iitgn.ac.in

Abstract

In many search applications related to passage retrieval, text entailment, and subgraph search, the query and each 'document' is a set of elements, with a document being relevant if it contains the query. These elements are not represented by atomic IDs, but by embedded representations, thereby extending set containment to *soft* set containment. Recent applications address soft set containment by encoding sets into fixed-size vectors and checking for elementwise *vector dominance*. This 0/1 property can be relaxed to an asymmetric *hinge distance* for scoring and ranking candidate documents. Here we focus on data-sensitive, trainable indices for fast retrieval of relevant documents. Existing LSH methods are designed for mostly symmetric or few simple asymmetric distance functions, which are not suitable for hinge distance. Instead, we transform hinge distance into a proposed *dominance similarity* measure, to which we then apply a Fourier transform, thereby expressing dominance similarity as an expectation of inner products of functions in the frequency domain. Next, we approximate the expectation with an importance-sampled estimate. The overall consequence is that now we can use a traditional LSH, but in the frequency domain. To ensure that the LSH uses hash bits efficiently, we learn hash functions that are sensitive to both corpus and query distributions, mapped to the frequency domain. Our experiments show that the proposed asymmetric dominance similarity is critical to the targeted applications, and that our LSH, which we call FOURIERHASHNET, provides a better query time vs. retrieval quality trade-off, compared to several baselines. Both the Fourier transform and the trainable hash codes contribute to performance gains.

1 Introduction

Consider a corpus X of sets x (which we call 'documents') over some universe of discrete items, and let q be a query which is also a subset of this universe. We wish to retrieve those $x \in X$ which satisfy $q \subseteq x$. In most real-world applications, the items in the universe are not just opaque IDs, but are embedded in a rich feature space, demanding that the definition of "$q \subseteq x$" be generalized suitably.

We formalize the notion of *soft set containment* by writing $q = \{q_i\}$ and $x = \{x_i\}$ and the corresponding sets of item embeddings as $\{\vec{q}_i\}$ and $\{\vec{x}_i\}$. If q, x are sentences, $\vec{q}_i, \vec{x}_i$ may be per-word contextual embeddings output from a transformer. If q, x are graphs, $\vec{q}_i, \vec{x}_i$ may be contextual node embeddings, such as those output by a Graph Neural Network (GNN). These set-of-vector representations of q and x are generally of variable sizes. A suitable set encoding gadget, such as simple pooling [38, 30] or a trainable Deep Set [56] or Set Transformer [26] network, converts them to fixed-size vectors given by $\boldsymbol{q} = \mathrm{SetEnc}(\{\vec{q}_i\})$ and $\boldsymbol{x} = \mathrm{SetEnc}(\{\vec{x}_i\})$, with $\boldsymbol{x}, \boldsymbol{q} \in \mathbb{R}^K$. Several applications [48, 25, 10, 30] then use the test "$\boldsymbol{q} \leq \boldsymbol{x}$" (elementwise vector dominance) as a surrogate for testing if $q \subseteq x$.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

To convert the Boolean test for vector dominance, $\boldsymbol{q} \leq \boldsymbol{x}$, into a graded score suitable for ranking (and backpropagation), these applications [48, 25, 10, 30] use a form of (asymmetric) **hinge distance**

$$d(q, x) = \left\| [\boldsymbol{q} - \boldsymbol{x}]_+ \right\|_1 = \sum_k \max\{0, \boldsymbol{q}[k] - \boldsymbol{x}[k]\}. \tag{1}$$

$d(q, x) = 0$ when $\boldsymbol{q} \leq \boldsymbol{x}$ holds elementwise, and measures the extent of the constraint violation otherwise. A search system must retrieve the top-τ documents x with the smallest $d(q, x)$, given query q. Several example applications that fit into this framework are elaborated in Appendix B. Even if an application does not fit (1) exactly, our technique may help address other asymmetric distances.

Our goal When corpus X is large, it is impractical to evaluate (1) for each document x. Our goal is to retrieve these τ documents without explicitly evaluating $d(q, x)$ for all $x \in X$, within query time that scales slowly with $|X|$. To achieve this, we design an asymmetric Locality Sensitive Hashing (ALSH) method tailored for hinge distance (1), which then immediately addresses soft set-containment based search.

Prior work and their limitations When set elements are represented by atomic IDs, Bloom filters [34] and maximum inner product search (MIPS) can be used to find the best τ corpus items that are closest to being supersets [42, 55, 41, 2]. However, these techniques are designed specifically for items with opaque IDs, rather than contextual embeddings. LSH [7, 49, 17, 19, 1] has been established as a standard technique for fast approximate near-neighbor search (e.g., FAISS, DPR) in the space of contextual embeddings. However, they predominantly work for symmetric notions of relevance, such as Jaccard similarity, dot product, cosine similarity, or Hamming distance, rather than asymmetric distances like (1). Neyshabur and Srebro [33] propose a LSH suited for asymmetric relevance (ALSH), but it does not provide a satisfactory solution for (1), as our experiments show.

1.1 Our contributions

Responding to the above motivations, we present FOURIERHASHNET, a new LSH for hinge distance-based asymmetric distance measures. Specifically, we make the following contributions.

Scalable hinge distance search for soft set containment From several applications, we distil the strongly-motivated problem of fast top-τ retrieval using hinge distance (1), to capture soft set containment. To our knowledge, (A)LSH for hinge distance has not been explored till date.

Transformation of hinge distance to enable ALSH design One could leverage its shift-invariant property to apply a Fourier transform on the *negative* distance, express it as the dot product similarity between the corresponding Fourier features and then use Asymmetric LSH (ALSH) [33]. However, as we show in Section 3.1, using the negative distance leads to singularities of the underlying Fourier transform at some points. This in turn does not allow us to design an LSH for such measure. We circumvent this problem by a suitable transformation of hinge distance to a **dominance similarity**, whose Fourier transform is absolutely convergent.

Design of Fourier features Next, we propose a novel method of lifting the dense vectors to frequency domain, such that the dominance similarity in the original space can be expressed as the cosine similarity between the infinite dimensional Fourier features. However, our dominance similarity function is *not* a positive definite kernel. Hence, unlike Rahimi and Recht [37], we cannot apply Bochner theorem [40] to obtain finite dimensional Fourier features. Instead, we first scale the Fourier features with a sinc function and then obtain finite dimensional Fourier features via importance sampling.

Trainable hashcode design The cosine similarity between the sampled Fourier features is the unbiased estimate of our dominance similarity measure. This allows the use of conventional random hyperplane LSH. However, such an LSH is not guided by the underlying data distribution. To mitigate this limitation, we compute the hashcodes by feeding the Fourier features into a trainable neural hashing network. Prior approaches [50, 15] to trainable hashing encourage bucket balance over the entire corpus, regardless of the query workload. However, this approach is not optimal if most corpus items are irrelevant for most queries, as is usually the case. We propose a new loss function that encourages the best-match hash bucket for a query to include relevant documents and exclude irrelevant documents.

Experiments We show, through extensive experiments, that FOURIERHASHNET is more effective than existing LSH schemes, and that both frequency domain representations and the new trainable hashcode contribute to our gains.

2 Preliminaries

Notation Throughout, we will use $[K]$ to mean $\{1, \ldots, K\}$ or $\{0, \ldots, K-1\}$ as convenient. We use q to indicate a query and x to indicate a corpus 'document'. Their (possibly learnt) representations are denoted by $\boldsymbol{x}, \boldsymbol{q} \in \mathbb{R}^K$. For supervision, (q, x) may come with a binary relevance judgment $\mathrm{rel}(q, x) \in \{0, 1\}$. We have defined a potentially learnable distance $d(q, x)$ — a computable surrogate for $\mathrm{rel}(q, x)$ — above in Eqn. (1). One can define a similarity measure $\mathrm{sim}(q, x)$ by applying a monotonically decreasing function on the distance $d(q, x)$. We define $\iota = \sqrt{-1}$ and denote the set of corpus items as $X = \{x_1, x_2, ..., x_N\}$. We indicate the domain of query and corpus items as $\mathcal{Q}$ and $\mathcal{X}$ respectively. Given a function $s(t)$, its Fourier transform is the function $S : \mathbb{C} \to \mathbb{C}$ which satisfies $s(t) = \int_{-\infty}^{\infty} S(\iota\omega) e^{\iota\omega t} d\omega$, where ω is the frequency variable and $S(\iota\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(t) e^{-\iota\omega t} dt$. For a vector $\boldsymbol{q}$ or $\boldsymbol{x} \in \mathbb{R}^K$, the Fourier transform is synthesized using a frequency vector $\boldsymbol{\omega} \in \mathbb{R}^K$ of same dimension as $\boldsymbol{x}$ or $\boldsymbol{q}$. Here, a function $s(\boldsymbol{x})$ can be expanded as $s(\boldsymbol{x}) = \int_{\boldsymbol{\omega} \in \mathbb{R}^K} S(\iota\boldsymbol{\omega}) e^{-\iota\boldsymbol{\omega}^\top \boldsymbol{x}} d\boldsymbol{\omega}$.

2.1 Locality sensitive hashing

Indexing corpus items Given a set of corpus items $X = \{x_1, x_2, ..., x_N\}$, an LSH will hash each item x_i a number of L times, which is called the number of *trials*. For each trial $\ell \in [L]$, it prepares B *buckets*, which are indexed as the pair (ℓ, b) with $\ell \in [L]$ and $b \in [B]$. In the context of LSH, we draw L independent samples of hash functions $h^{(\ell)}$ from a single hash family $\mathcal{H}$, such that $h^{(\ell)} : \mathbb{R}^K \to [B]$. A corpus item x is inserted in the bucket indexed $(\ell, h^{(\ell)}(\boldsymbol{x}))$, for each $\ell \in [L]$.

Symmetric LSH Given a query q, a symmetric LSH computes bucket indexes $(\ell, h^{(\ell)}(\boldsymbol{q}))$ for all L using the *same* hash function $h^{(\ell)}$ used for indexing the corpus. Only those items x that are in bucket $(\ell, h^{(\ell)}(\boldsymbol{q}))$ are considered as *candidates*; overall, the candidates are in the union of these buckets. In the rest of the paper, we will describe retrieval for one bucket under one trial, with the understanding that L buckets will contribute candidates. An LSH exists if the query and corpus items are hashed in the same bucket with high (low) probability as long as their similarities are high (low). Formally, we define symmetric LSH as follows.

Definition 2.1 (Symmetric Locality Sensitive Hashing (LSH))**.** Given a domain of queries $\mathcal{Q}$ and corpus $\mathcal{X}$ with $\mathcal{Q}, \mathcal{X} \subset \mathcal{Z}$ and a similarity measure $\mathrm{sim} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$. A distribution over mappings $\mathcal{H} : \mathcal{Z} \to \mathbb{N}$ is said to be a (S_0, cS_0, p_1, p_2)-LSH for the similarity function sim if for all $q \in \mathcal{Q}$ and $x \in \mathcal{X}$ we have, with $p_1 > p_2$ and $c < 1$,

- if $\mathrm{sim}(q, x) \geq S_0$, then $\Pr_{h \sim \mathcal{H}}[h(q) = h(x)] \geq p_1$
- if $\mathrm{sim}(q, x) \leq cS_0$, then $\Pr_{h \sim \mathcal{H}}[h(q) = h(x)] \leq p_2$.

The hash family $\mathcal{H}$ is tailored to the specific choice of similarity function $\mathrm{sim}(q, x)$ (equivalently, the distance $d(q, x)$). When $\boldsymbol{q}, \boldsymbol{x} \in \mathbb{R}^K$ and $\mathrm{sim}(q, x) = \cos(\boldsymbol{q}, \boldsymbol{x})$, the choice of $\mathcal{H}$ corresponds to the uncountable set of all hyperplanes in K dimensions passing through the origin [9]. When $\mathrm{sim}(q, x)$ is the Jaccard similarity $|q \cap x|/|q \cup x|$, $\mathcal{H}$ is the space of *minwise independent* hash functions [7].

Asymmetric LSH (ALSH) In many applications, like the current setup (1), we have asymmetric similarity where $\mathrm{sim}(q, x) \neq \mathrm{sim}(x, q)$. In such cases, we employ two different hash families $\mathcal{G}$ and $\mathcal{H}$ to determine the bucket of query and corpus respectively. Formally, we define ALSH as follows:

Definition 2.2 (Asymmetric Locality Sensitive Hashing (ALSH) [33])**.** An asymmetric LSH is (S_0, cS_0, p_1, p_2)-ALSH for a similarity function $\mathrm{sim}(\bullet, \bullet)$ over $\mathcal{Q}$, $\mathcal{X}$ if we have two different distributions over mappings $\mathcal{G}$ and $\mathcal{H}$ such that, with $p_1 > p_2$ and $c < 1$,

- if $\mathrm{sim}(q, x) \geq S_0$ then $\Pr_{g \sim \mathcal{G}, h \sim \mathcal{H}}[g(q) = h(x)] \geq p_1$
- if $\mathrm{sim}(q, x) \leq cS_0$ then $\Pr_{g \sim \mathcal{G}, h \sim \mathcal{H}}[g(q) = h(x)] \leq p_2$.

As an example, given $\|\boldsymbol{x}\| \leq 1$, consider $\mathrm{sim}(q, x) = \boldsymbol{q}^\top \boldsymbol{x}/||\boldsymbol{q}||_2$, which can be re-written as $\cos(\alpha(\boldsymbol{q}), \beta(\boldsymbol{x}))$, where $\alpha(\boldsymbol{q}) = [0; \boldsymbol{q}/\|\boldsymbol{q}\|_2], \beta(\boldsymbol{x}) = [\sqrt{1 - \|\boldsymbol{x}\|_2^2}; \boldsymbol{x}]$. Thus, we can apply random hyperplane hash on both $\alpha(\boldsymbol{x})$ and $\beta(\boldsymbol{x})$ to construct $g(q) = \mathrm{sign}(\mathbf{w} \cdot \alpha(\boldsymbol{q}))$ and $h(x) = \mathrm{sign}(\mathbf{w} \cdot \beta(\boldsymbol{x}))$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. If $\|\boldsymbol{x}\|$ is unbounded, no ALSH exists for $\mathrm{sim}(q, x) = \boldsymbol{q}^\top \boldsymbol{x}/||\boldsymbol{q}||_2$ [33]. In (S_0, cS_0, p_1, p_2)-ALSH, retrieval of items with similarity score more than S_0 out of a database of items having a similarity score less than cS_0 will admit time-complexity $O(n^\rho \log n)$ and space complexity $O(n^{1+\rho})$ where $\rho = \log p_1 / \log p_2$ [33].

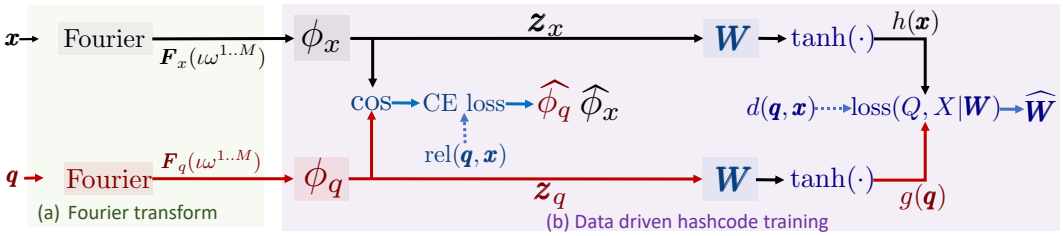


Figure 1: Outline of FOURIERHASHNET. **(a)** Given the input query embedding $\boldsymbol{q}$ and corpus embedding $\boldsymbol{x}$, we apply the asymmetric transformation in Eq. (10) to obtain corresponding Fourier features $\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M})$ and $\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})$. **(b)** We use the generated Fourier features as inputs, to train asymmetric Fourier transformation networks ϕ_q and ϕ_x using Eq. (11). This generates transformed Fourier representations $\boldsymbol{z}_q = \phi_q(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M}))$ and $\boldsymbol{z}_x = \phi_x(\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M}))$, which are in turn used to train the random hyperplanes $\boldsymbol{W}$ using Eq. (13). The trained $\widehat{\phi_q}$, $\widehat{\phi_x}$ and $\widehat{\boldsymbol{W}}$ thus obtained are used to generate final hashcodes$g(q) = \text{sign}(\widehat{\boldsymbol{W}}\widehat{\phi_q}(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M})))$, $h(x) = \text{sign}(\widehat{\boldsymbol{W}}\widehat{\phi_x}(\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})))$.

2.2 Problem statement

Given the set of training queries Q and corpus X, with supervised relevance scores $\text{rel}(q, x) \in \{0, 1\}$ and the surrogate score $d(q, x)$ defined in Eq. (1), we aim to design an LSH of the distance $d(q, x)$ which can efficiently retrieve top-τ corpus items for any new query q'.

Why are existing methods not suitable? As we discussed in Section 2.1, relevance metrics for popular LSHs are mostly symmetric, *e.g.*, cosine, dot-product, and Jaccard similarity. In particular, Jaccard similarity, although commonly used in set-related applications, is not suitable for our problem, where we define $\text{rel}(q, x) = 1$ when $q \subseteq x$ and 0 otherwise — it is possible that there exists a higher overlap between q and x when $q \not\subseteq x$, and a lower overlap when $q \subseteq x$. E.g., suppose $q = \{a, b\}$, $x_1 = \{a, b, c, d, e\}$, and $x_2 = \{b\}$. Here, $\text{rel}(q, x_1) = 1$ and $\text{rel}(q, x_2) = 0$. However, the Jaccard similarity $J(q, x)$ is not able to reflect the order of $\text{rel}(q, x)$ since $J(q, x_1) = 2/5 < J(q, x_2) = 1/2$.

As discovered by Charikar [9, Lemma 1], the similarity functions in symmetric LSH are inversely related to a *metric*, which must satisfy symmetry and triangle inequality. Although a query normalized dot product similarity appears asymmetric, it can be expressed using cosine similarity. This readily allows us to use a random hyperplane based (asymmetric) LSH. In contrast, it is not immediately apparent how to find such a connection for our asymmetric hinge distance (1).

3 FOURIERHASHNET: A new ALSH for hinge distance search

Overview of our approach We design an ALSH for $d(q, x)$ in three steps. In the first step, we construct a suitable dominance similarity function $\text{sim}(q, x)$ from $d(q, x)$ in such a way that there exists a probability distribution $p : \mathbb{R}^K \to [0, 1]$ and bounded Fourier representations $\boldsymbol{F}_q(\iota\boldsymbol{\omega})$ and $\boldsymbol{F}_x(\iota\boldsymbol{\omega})$ of both query q and corpus items x such that

$$\text{sim}(q, x) = \int_{\boldsymbol{\omega}\in\mathbb{R}^K} \boldsymbol{F}_q(\iota\boldsymbol{\omega})^\top \boldsymbol{F}_x(\iota\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega} = \mathbb{E}_{\boldsymbol{\omega}\sim p(\bullet)}[\boldsymbol{F}_q(\iota\boldsymbol{\omega})^\top \boldsymbol{F}_x(\iota\boldsymbol{\omega})] \tag{2}$$

In the second step, we approximate the expected value of the $\boldsymbol{F}_q(\iota\boldsymbol{\omega})^\top \boldsymbol{F}_x(\iota\boldsymbol{\omega})$ using a finite sample of Fourier features. This allows us to apply random hyperplane LSH , similar to asymmetric dot product LSH. However, these hyperplanes are drawn from an isotropic Gaussian distribution in a data-oblivious manner, which results in suboptimal bucket distribution in terms of accuracy-efficiency trade off. To tackle this issue, in the third step, we train the random hyperplanes $\boldsymbol{W}$ which takes the Fourier features as input and give (soft) binary hashcodes, which are optimized to effectively trade off between accuracy and efficiency. Next, we provide the details of the above three steps.

3.1 Design of dominance similarity function $\text{sim}(q, x)$ from hinge distance

Limitations of simple choices of dominance similarity function $\text{sim}(q, x)$ A dominance similarity function $\text{sim}(q, x)$ is inversely related to the hinge distance $d(q, x)$. Chierichetti and Kumar [11] characterized that, any function of a similarity measure is LSHable, if and only if this function is a probability generating function. However, this characterization applies only to symmetric LSH and no such guiding principle is available for an ALSH. In this context, one can experiment with simple designs of sim that are inversely related to d. An immediate choice is $\text{sim}(q, x) = -d(q, x)$. However, if we allow $\boldsymbol{q}$ and $\boldsymbol{x}$ to be any vector from $\mathbb{R}^K$, then $\text{sim}(q, x)$ is not bounded. Finding ALSH for unbounded similarity measures is extremely difficult if not impossible. For example, no

ALSH exists even for dot product similarity in two or more dimensions [33]. Moreover, suppose we express $\mathrm{sim}(q,x)$ using the Fourier expansion

$$\mathrm{sim}(q,x) = \sum_{k\in[K]} \int_{-\infty}^{\infty} S(\iota\omega_k) e^{\iota\omega_k(\boldsymbol{q}[k]-\boldsymbol{x}[k])} d\omega_k. \tag{3}$$

Then, $S(\iota\omega)$, *i.e.*, the Fourier transform of the function $s(t) = -[t]_+$ used in each dimension, has a singularity at $\omega = 0$. In particular, we have $S(\iota\omega) = -\iota\delta'(\omega)/2 + 1/2\pi\omega^2$. Here, $\delta'(\omega)$ is the derivative of Dirac delta functional. Thus, $S(\iota\omega)$ becomes unbounded as $\omega \to 0$. These issues eventually prevent us from designing bounded Fourier features $\boldsymbol{F}_q(\iota\boldsymbol{\omega})$ and $\boldsymbol{F}_x(\iota\boldsymbol{\omega})$ for Eq. (2).

$\mathrm{sim}(q,x)$ with bounded Fourier transform The key reason for which $S(\iota\omega)$ becomes unbounded as $\omega \to 0$ is that the function $s(t) = -[t]_+$ is unbounded at $t \to \infty$. However, in practice, the embeddings are bounded and we have a bounded difference $|\boldsymbol{q}[k] - \boldsymbol{x}[k]| \leq T$. Thus, it is reasonable to ignore the effect of $[\boldsymbol{q}[k] - \boldsymbol{x}[k]]_+$ when $|\boldsymbol{q}[k] - \boldsymbol{x}[k]| > T$. To this end, we compute $\mathrm{sim}(q,x)$ as

$$\mathrm{sim}(q,x) = \sum_{k\in[K]} s(\boldsymbol{q}[k] - \boldsymbol{x}[k]), \qquad \text{where } s(t) = \begin{cases} T - t & \text{if } 0 \leq t \leq T, \\ T & \text{if } -T \leq t < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In practice, we choose T as a hyperparameter greater than $\max_k |\boldsymbol{q}[k] - \boldsymbol{x}[k]|$. Upon restricting the computation within this domain, one can immediately show that $\mathrm{sim}(q,x) = KT - d(q,x)$.

3.2 Computation of finite dimensional Fourier features for dominance similarity $\mathrm{sim}(q,x)$

Fourier transform of $s(t)$ We next compute the Fourier representation $S(\iota\omega)$ of $s : \mathbb{R} \to \mathbb{R}$ (4).

Proposition 3.1. *(Proven in Appendix D) $s(t)$ specified in Eq. (4) has Fourier transform*

$$S(\iota\omega) = \underbrace{\frac{T\sin(\omega T)}{2\pi\omega} + \frac{\sin^2(\frac{\omega T}{2})}{\pi\omega^2}}_{\mathrm{Re}(S(\iota\omega))} + \iota \underbrace{\left[\frac{\sin(\omega T)}{2\pi\omega^2} - \frac{T\cos(\omega T)}{2\pi\omega}\right]}_{\mathrm{Im}(S(\iota\omega))} \tag{5}$$

While the Fourier transform of $-[t]_+$ is unbounded as $\omega \to 0$, here, $S(\iota\omega)$ is bounded everywhere.

Computation and sampling of Fourier features Once we compute $S(\iota\omega)$ using Eq. (5), we use Eq. (3) to compute $\mathrm{sim}(q,x)$ as follows:

$$\mathrm{sim}(q,x) = \sum_{k\in[K]} \int_{-\infty}^{\infty} [\mathrm{Re}(S(\iota\omega_k)) + \iota\mathrm{Im}(S(\iota\omega_k))] e^{\iota\omega_k(\boldsymbol{q}[k]-\boldsymbol{x}[k])} d\omega_k \tag{6}$$

Now, we define $\boldsymbol{S}_q(\iota\boldsymbol{\omega})$ and $\boldsymbol{S}_x(\iota\boldsymbol{\omega})$, the query and corpus specific Fourier representations:

$$\begin{aligned} \boldsymbol{S}_q(\iota\omega_k) = \Big[& u_k \sqrt{|\mathrm{Re}(S(\iota\omega_k))|}\big[\cos(\omega_k\boldsymbol{q}[k]), \sin(\omega_k\boldsymbol{q}[k])\big], \\ & v_k \sqrt{|\mathrm{Im}(S(\iota\omega_k))|}\big[-\sin(\omega_k\boldsymbol{q}[k]), \cos(\omega_k\boldsymbol{q}[k])\big]\Big] \\ \boldsymbol{S}_x(\iota\omega_k) = \Big[& \sqrt{|\mathrm{Re}(S(\iota\omega_k))|}\big[\cos(\omega_k\boldsymbol{x}[k]), \sin(\omega_k\boldsymbol{x}[k])\big], \\ & \sqrt{|\mathrm{Im}(S(\iota\omega_k))|}\big[\cos(\omega_k\boldsymbol{x}[k]), \sin(\omega_k\boldsymbol{x}[k])\big]\Big] \end{aligned} \tag{7}$$

Here, $u_k = \mathrm{sign}(\mathrm{Re}(S(\iota\omega_k)))$, $v_k = \mathrm{sign}(\mathrm{Im}(S(\iota\omega_k)))$. They ensure that the dot-product $\boldsymbol{S}_q(\iota\omega_k)^\top \boldsymbol{S}_x(\iota\omega_k)$ equals to the real part of the integrand in the RHS of Eq. (6). Since the dominance similarity $\mathrm{sim}(q,x)$ is a real quantity, the imaginary part of the RHS integrates to zero. Therefore, using the dot product of the vectors $\boldsymbol{S}_q(\iota\omega_k)$ and $\boldsymbol{S}_x(\iota\omega_k)$, which are purely real, we can express

$$\mathrm{sim}(q,x) = \int_{-\infty}^{\infty} \sum_{k\in[K]} \boldsymbol{S}_q(\iota\omega_k)^\top \boldsymbol{S}_x(\iota\omega_k) d\omega_k = \int_{\boldsymbol{\omega}\in\mathbb{R}^K} \boldsymbol{S}_q(\iota\boldsymbol{\omega})^\top \boldsymbol{S}_x(\iota\boldsymbol{\omega}) d\boldsymbol{\omega} \tag{8}$$

Here, $\boldsymbol{S}_\bullet(\iota\boldsymbol{\omega}) = [\boldsymbol{S}_\bullet(\iota\omega_1), .., \boldsymbol{S}_\bullet(\iota\omega_K)]$, $\boldsymbol{\omega} = [\omega_1, .., \omega_K]$. Note that, the expression of $\boldsymbol{S}_q(\iota\omega_k)$ is different from $\boldsymbol{S}_x(\iota\omega_k)$ in Eq. (7). This maintains the asymmetry in the final dot product in Eq. (8).

Inspired by the seminal work of Rahimi and Recht [37], several works [46, 35] have exploited Fourier transformations to approximate various functions using inner product between the feature maps. However, the functions that Rahimi and Recht [37] considered are shift invariant positive definite kernels. This allowed them to leverage Bochner's theorem [40] which establishes that the Fourier transformation of these kernels are probability distributions. However, in Eq. (8), there is no such readily available probability distribution. In response, we attempt to find out a probability distribution

$p(\boldsymbol{\omega})$ which allows us to draw samples using an importance sampling like procedure, as follows:

$$\text{sim}(q, x) = \mathbb{E}_{\boldsymbol{\omega} \sim p(\boldsymbol{\omega})} \left[\boldsymbol{F}_q(\iota\boldsymbol{\omega})^\top \boldsymbol{F}_x(\iota\boldsymbol{\omega}) \right], \text{ where, } \boldsymbol{F}_q(\iota\boldsymbol{\omega}) = \frac{\boldsymbol{S}_q(\iota\boldsymbol{\omega})}{\sqrt{p(\boldsymbol{\omega})}}, \boldsymbol{F}_x(\iota\boldsymbol{\omega}) = \frac{\boldsymbol{S}_x(\iota\boldsymbol{\omega})}{\sqrt{p(\boldsymbol{\omega})}}, \quad (9)$$

Let $\{\boldsymbol{\omega}^j\}_{j=1}^M \sim p(\boldsymbol{\omega})$ be M i.i.d random samples. We compute the Monte Carlo estimate as follows:

$$\text{sim}(q, x) \approx \frac{1}{M} \sum_{j \in [M]} \boldsymbol{F}_q(\iota\boldsymbol{\omega}^j)^\top \boldsymbol{F}_x(\iota\boldsymbol{\omega}^j) \propto \cos(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M}), \boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})) \quad (10)$$

Here, $\boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^{1..M}) = [\boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^1), .., \boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^M)]$. Note that, as suggested by Eqs. (7) and (9), $||\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1...M})||_2 = ||\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})||_2 = \sum_{j=1}^M \sum_{k=1}^K \frac{|\text{Re}(S(\iota\omega_k^j))| + |\text{Im}(S(\iota\omega_k^j))|}{p(\omega_k^j)}$. Thus, the value is independent of the query or corpus, which leads to the proportionality relation. We choose the probability distribution $p(\boldsymbol{\omega})$ guided the proportionality constant $||\boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^{1..M})||$ and set $p(\boldsymbol{\omega}) = \prod_{k \in [K]} p(\omega)$, where $p(\omega) \propto |\text{Re}(S(\iota\omega))| + |\text{Im}(S(\iota\omega))|$. However, the integral of these terms may not be bounded. Therefore, we set the support of $p(\omega)$ between $[-\omega_{\max}, \omega_{\max}]$, thus eliminating the higher frequency terms. The effect on the overall score is small.

3.3 Trainable hashing network

Random hyperplane LSH Eq. (10) provides an asymmetric transformation on the input query-corpus pair, which maps it into the cosine similarity space, thus allowing for Random Hyperplanes hashing. We sample H spherically symmetrically distributed normal vectors $\{\mathbf{w}_i\}_{i=1}^H$, *i.e.*, $\mathbf{w}_i \sim \mathcal{N}(0, \mathbb{I})$, each perpendicular to a random hyperplane passing though the origin. For each query q and the corpus x, we can generate H-bit hashcodes $g(q), h(x) \in \{\pm 1\}^H$ from the Fourier features (10) as follows: $g(q)[i] = \text{sign}(\mathbf{w}_i^\top \boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1...M}))$ and $h(x)[i] = \text{sign}(\mathbf{w}_i^\top \boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1...M}))$. Consequently, we can index the given corpus with N items, into a hash table with 2^H buckets. For each query q, we restrict our search within bucket $b = g(q)$. If the corpus items are uniformly distributed across all buckets, then it enables sub-quadratic time retrieval with $N/2^H$ comparisons (per trial).

Data driven hashcode generation The above random hyperplane LSH approach suffers from two distinct limitations: (1) the quality of Monte Carlo approximation obtained in Eq. (10), depends on the suitability of $p(\boldsymbol{\omega})$, and (2) the hyperplanes are data oblivious. Data oblivious hyperplanes provide the best efficiency if the corpus embeddings are uniformly spread over the K dimensional sphere, which allows the random hyperplanes to evenly allocate the corpus items across different hashcodes. However, in practice, the spatial distribution of the embeddings is not uniform. This results in a skewed distribution of the corpus items across the hash buckets.

To tackle the first problem, we improve the quality of the Fourier features through a trainable nonlinear transformation. Here, we use two networks ϕ_q and ϕ_x which takes the Fourier features for the query and corpus, *i.e.*, $\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M})$ and $\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})$ as input and outputs corresponding transformed Fourier representations $\boldsymbol{z}_q = \phi_q(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M}))$ and $\boldsymbol{z}_x = \phi_x(\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M}))$. We train ϕ_q and ϕ_x by minimizing a BCE loss on $\{\cos(\boldsymbol{z}_q, \boldsymbol{z}_x), \text{rel}(q, x)\}$ pairs for $q \in Q$ and $x \in X$ as follows:

$$\min_{\phi_q, \phi_x} \sum_{q \in Q, x \in X} -\big[\text{rel}(q, x) \log(1 + \cos(\boldsymbol{z}_q, \boldsymbol{z}_x)) + (1 - \text{rel}(q, x) \log(1 - \cos(\boldsymbol{z}_q, \boldsymbol{z}_x))\big] \quad (11)$$

Next, we train the random hyperplanes $\boldsymbol{W} = [\mathbf{w}_1, \mathbf{w}_2, ..]$ using the transformed Fourier features $\{\boldsymbol{z}_q\}$ and $\{\boldsymbol{z}_x\}$. The final hashcodes $g(q)$ and $h(x)$ are obtained as $g(q) = \text{sign}(\widehat{\boldsymbol{W}}\boldsymbol{z}_q)$, $h(x) = \text{sign}(\widehat{\boldsymbol{W}}\boldsymbol{z}_x)$, where $\widehat{\boldsymbol{W}}$ are the final trained random hyperplanes. For training purposes, we use $\tanh(\boldsymbol{W}\bullet)$ as a smooth surrogate of $\text{sign}(\boldsymbol{W}\bullet)$. The loss function $\text{loss}(Q, X \mid \boldsymbol{W})$ used to train $\boldsymbol{W}$ consists of three components.

(1) Collision minimizer For any query q, our goal is to ensure that assigned bucket contains only positive items. Assuming corpus items are uniformly distributed across buckets, we ensure that for any query q, the $N/2^H$ most relevant items $X_{q\checkmark}$ measured in terms of $d(q, x)$ will have higher amount of bit overlap than rest of the items

Algorithm 1 FOURIERHASHNET

function Train(X,Q, $\{\text{rel}(q, x)\}_{q \in Q, x \in X}$)
  Draw $\boldsymbol{\omega}^{1...M} \sim p(\boldsymbol{\omega})$
  Compute $\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M})$, $\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})$ (Eq. (9))
  Train ϕ_q, ϕ_x from $\text{rel}(q, x)$, $\boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^{1..M})$ (Eq. (11))
  Train $\boldsymbol{W}$ by minimizing the loss (13)
  Return $\widehat{\phi_x}, \widehat{\phi_q}, \widehat{\boldsymbol{W}}$

function Index($\{\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})\}_{x \in X}$)
  Require: Trained networks $\widehat{\phi_x}, \widehat{\boldsymbol{W}}$
  $h(x) \leftarrow \text{sign}(\widehat{\boldsymbol{W}}\widehat{\phi_x}(\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M}))) \; \forall x \in X$
  for $x \in X$ **do**
    hash x to bucket $b = h(x)$
  Return the bucket sets B

function Retrieve(q')
  Require: Trained networks $\widehat{\phi_q}, \widehat{\boldsymbol{W}}$
  Compute $\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1...M})$ based on $\boldsymbol{q}'$
  $g(q') \leftarrow \text{sign}(\widehat{\boldsymbol{W}}\widehat{\phi_q}(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1...M})))$
  Rank all x in the bucket $b = g(q')$ based on the distance $d(q', x)$ to obtain the list $\text{List}_{q'}$.
  Return $\text{List}_{q'}$

$X_{q\boldsymbol{\times}}$. Here, $X_{q\checkmark}$ and $X_{q\boldsymbol{\times}}$ indicate positive and negative silver instances (not gold instances) indicating top $N/2^H$ items *in terms of the (possibly trained) hinge distance* $d(q, x)$. We encode this by minimizing the following ranking loss.

$$\Delta_1 = \sum_{q \in Q} \sum_{x \in X_{q\checkmark}, x' \in X_{q\boldsymbol{\times}}} \left[1 + \tanh(\boldsymbol{W}\boldsymbol{z}_q)^\top \tanh(\boldsymbol{W}\boldsymbol{z}_{x'}) - \tanh(\boldsymbol{W}\boldsymbol{z}_q)^\top \tanh(\boldsymbol{W}\boldsymbol{z}_x)\right]_+ \tag{12}$$

This loss encourages that $\tanh(\boldsymbol{W}\boldsymbol{z}_q)^\top \tanh(\boldsymbol{W}\boldsymbol{z}_x) > \tanh(\boldsymbol{W}\boldsymbol{z}_q)^\top \tanh(\boldsymbol{W}\boldsymbol{z}_{x'}) + 1$, *i.e.*, the number of common bits between q and $x \in X_{q\checkmark}$ is atleast one more than the same between q and x'.

(2) Fence Sitting We set fence sitting loss as $\Delta_2 = \sum_{x \in X} |\|\tanh(\boldsymbol{W}\boldsymbol{z}_x)| - 1\|_1$. This prevents the optimizer from arriving at a trivial solution by setting all hashcodes to zero.

(3) Bit Balance We set the bit balance loss as $\Delta_3 = \sum_{i \in [H]} |\sum_{x \in X} \tanh(\boldsymbol{W}\boldsymbol{z}_x)[i]|$. This enforces that each position should have an equal number of $+1$ and -1 , thus ensuring that each random hyperplane evenly splits the set of points. Finally, we estimate $\boldsymbol{W}$ by minimizing the loss, with $\lambda_\bullet$ as hyperparameters such that $\sum_i \lambda_i = 1$, which is given as follows:

$$\text{loss}(Q, X \mid \boldsymbol{W}) = \lambda_1\Delta_1 + \lambda_2\Delta_2 + \lambda_3\Delta_3, \tag{13}$$

Algorithm 1 summarizes the overall procedure.

Difference from existing trainable LSH LSH training has been extensively studied [50, 15], with Fence Sitting and Bit Balance losses being well known. However, the Collision Minimizer loss differs significantly from existing approaches. Current techniques seek to ensure load balance across hash buckets for all corpus items, including the ones that may not be relevant to most queries. This is unnecessary for query workloads which touch upon only a small subset of the corpus to generate the best responses. In contrast, our Collision Minimizer loss ensures that only the top-most bucket for any given query allows relevant items and explicitly denies irrelevant items. Thus, it is informed by the query workload, rather than assuming load balance for all items in the corpus. Such an approach may result in balanced bucket loads, but not necessarily.

4 Experiments

In this section, we provide a comprehensive evaluation of our method against several baselines and ablations on four datasets. Appendix F describes additional experiments.

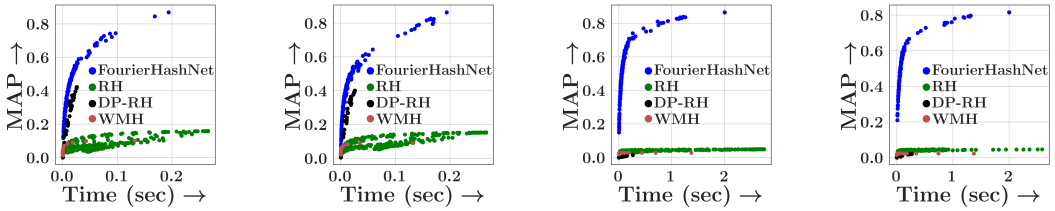
4.1 Experimental setup

Datasets We experiment on datasets sampled from anonymized real-world Web log data, *viz*, MSWEB and MSNBC. MSWEB [5] is generated using logs from *www.microsoft.com*, containing records of the areas of the website visited by the users. MSNBC [8] is a collection of logs of user page requests from *msnbc.com*. In both cases, a record (either q or x), is a passage that is regarded as a bag of words. Given a collection V of such word bags, ($|V| = 11234$ for MSWEB and $|V| = 111290$ for MSNBC), we sample $|Q| = 500$ bags from V, designating them as queries, and designate the rest as corpus items $X = V \backslash Q$. Consistent with typical information retrieval application scenarios [47], we generate gold relevance labels based on (multi)set containment for MSWEB (MSNBC). (Additional methods for evaluation are explored in Appendix F.) We build the query set Q, such that the number of relevant items $N_{q\oplus} = |\{x \in X : \text{rel}(q, x) = +1\}| \in [5, 500]$ for each query q. We create four datasets by changing average relevance counts per query, $\overline{N}_{q\oplus}$. They are: (1) MSWEB-1 where $\overline{N}_{q\oplus} = 35.624$. (2) MSWEB-2 where $\overline{N}_{q\oplus} = 20.392$. (3) MSNBC-1 where $\overline{N}_{q\oplus} = 24.09$ (4) MSNBC-2 where $\overline{N}_{q\oplus} = 19.78$ The set of queries Q is partitioned into 20% training set Q_{tr}, 20% validation set Q_{dev} and 60% test set Q_{test}.

Design of query and corpus embeddings $\boldsymbol{q}, \boldsymbol{x}$ We begin with a pre-trained sentence transformer model [38] to obtain 768 dimensional dense contextual representations $\textbf{feature}_q$ and $\textbf{feature}_x$ for the each word in bags q and x. Embeddings of words belonging to a bag are fed into a deep set [56] network to obtain a bag representation $\boldsymbol{q}, \boldsymbol{x} \in \mathbb{R}^K$, with $K = 294$ (chosen via hyperparameter sweep). To train the parameters inside the deep set network, we use $\boldsymbol{q}, \boldsymbol{x}$ to compute the proposed asymmetric hinge distance $d(q, x)$ (1), feed it into a trainable sigmoid layer σ and minimize

$$\textstyle\sum_{q,x} \text{BCE}\left(\text{rel}(q, x), \sigma(-d(q, x))\right) \tag{14}$$

which uses a BCE loss on the gold relevance labels. Once we obtain $\boldsymbol{q}$ and $\boldsymbol{x}$, we use Algorithm 1 to obtain trained $\widehat{\phi_q}$, $\widehat{\phi_x}$ and $\widehat{\boldsymbol{W}}$ (Train$(\cdot)$), which are then used for indexing (Index$(\cdot)$).



(a) MSWEB-1 (b) MSWEB-2 (c) MSNBC-1 (d) MSNBC-2

Figure 2: Effect of different similarity measures on LSH, measured in terms of variation of MAP vs. average query time (in sec) for all methods. Here, the final score used for ranking the relevant items is that similarity score for which the LSH is designed for.

Evaluation Given a test query $q \in Q_{\text{test}}$ and a set of N'_q candidate corpus items, we rank them in increasing order of their hinge distances $d(q, x)$. Then we evaluate the average precision (AP) for the query and average over queries to report mean average precision (MAP) — see Appendix E.6.

4.2 Effect of different similarity measures on LSH

Setup Here, we compare FOURIERHASHNET against the three LSH baselines, *viz*, Random hyperplane (RH) [9], Dot product LSH (DP-RH) [33] and Weighted MinHash (WMH) [12], that are tailored towards cosine similarity, dot product similarity and Weighted Jaccard Similarity, respectively. For each LSH method, we train the embeddings $\boldsymbol{q}$, $\boldsymbol{x}$ and the final hashcodes $g(q)$ and $h(x)$ using the same networks, as in our method. Furthermore, we set the final relevance measure for ranking to be the similarity score for which the LSH is designed.

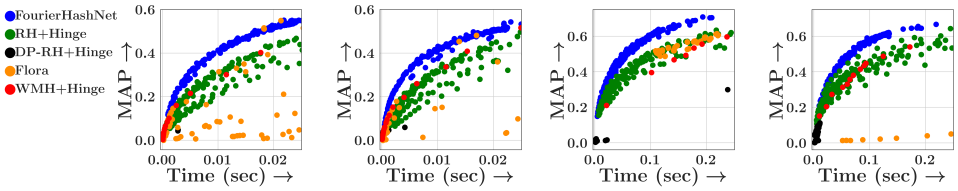
Results We vary the mixing hyperparameters λ_1 and λ_2 in our loss (13) and the number of buckets B to explore the tradeoff between accuracy (MAP) and average query time. In Figure 2, we summarize the results. We observe that: **(1)** FOURIERHASHNET outperforms all the baselines by providing significantly better time-vs.-MAP trade-off across all datasets. In MSWEB datasets, all the baselines except DP-RH show poor performance. All baselines perform poorly for the MSNBC dataset. We remark that cosine similarity, dot product or weighted Jaccard similarity are not suited for vector dominance search. Therefore, the maximum possible MAP obtained by them are severely constrained. **(2)** In MSWEB datasets, DP-RH performs moderately, by achieving a MAP value around 0.4–0.42 within 0.03 seconds (average query time). This is because dot product can be computed significantly faster than all the other distance/similarity measures. In particular, it is $\sim$7.5$\times$ faster than our hinge distance (1), $\sim$10.3$\times$ faster than cosine, and $\sim$5.1$\times$ faster than Jaccard similarity.

4.3 Applying baseline LSHs on hinge distance guided embeddings

Setup In the preceding experiments, we used the similarity score corresponding to each LSH method for final candidate ranking. The baselines performed poorly, which may result from a poor choice of final similarity score or the LSH method. To tease these apart, we set the final similarity function to be dominance similarity, irrespective of the LSH method used to filter candidates. Indeed, Shrivastava and Li [43] showed that an LSH not tailored to the final scoring function may still provide an effective filter. We compare against four such possible baselines.

Given the embeddings $\boldsymbol{q}, \boldsymbol{x}$ trained (14) using hinge distance, we feed them into the four baselines, each of which trains a hashing network in a different way. **(1)** RH+Hinge: We train a set of random hyperplanes represented by $\boldsymbol{W}$ and compute the hashcodes as $h(q) = \text{sign}(\boldsymbol{W}\boldsymbol{q})$ and $h(x) = \text{sign}(\boldsymbol{W}\boldsymbol{x})$. **(2)** DP-RH+Hinge: We train random hyperplanes $\boldsymbol{W}$ for these embeddings to compute the hashcodes as $g(q) = \text{sign}(\boldsymbol{W}[0, \boldsymbol{q}/||\boldsymbol{q}||])$ and $h(x) = \text{sign}(\boldsymbol{W}[\sqrt{T^2 - ||\boldsymbol{x}||^2}, \boldsymbol{x}])$. **(3)** WMH+Hinge: We use the best performing WMH implementation from DrHash toolkit [54] to obtain the hashcodes. **(4)** FLORA[15]: We train asymmetric hash networks (net_1, net_2) using an end-to-end data-driven approach, which minimizes bit balance and decorrelation loss, along with a consistency loss which predicts the final similarity score using $\cos(\text{net}_1(\boldsymbol{q}), \text{net}_2(\boldsymbol{x}))$.

Results Figure 3 compares the performance of FOURIERHASHNET, RH+Hinge, DP-RH+Hinge, WMH+Hinge and FLORA in terms of MAP for MSWEB and MSNBC datasets. Here we analyze the section of the trade-off curve which provides $\geq$10X speedup compared to exhaustive search. The complete tradeoff curve is provided in Appendix F. **(1)** The newly designed baselines are now seen to perform significantly better than those used in the previous experiments with Figure 2. However FOURIERHASHNET still outperforms all the baselines. **(2)** RH+Hinge, despite achieving



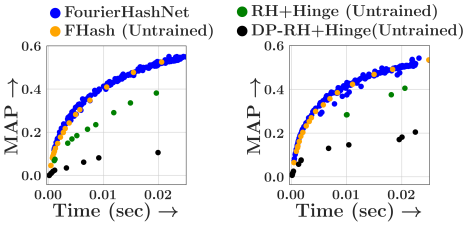
(a) MSWEB-1 (b) MSWEB-2 (c) MSNBC-1 (d) MSNBC-2

Figure 3: Trade-off between query time and accuracy (MAP) for MSWEB datasets where there is $\geq$10X speedup compared to exhaustive search. We apply different LSH methods on hinge distance guided embeddings, *viz*, RH+Hinge, DP-RH+Hinge, WMH+Hinge, FLORA and FOURIERHASHNET; and, then use the hinge distance to finally rank the retrieved items.

the second highest scores in many cases, is seen to suffer from a large variance in performance within any given time budget. This would make it difficult to tune the hyperparameters to achieve the requisite performance v/s retrieval speed trade-off. **(3)** DP-RH+Hinge is seen to have a significantly worse performance than FOURIERHASHNET everywhere. This indicates that DP-RH is ill-suited to asymmetric hinge distance based retrieval.**(4)** We observe that for the same amount of query time invested, FLORA's MAP can lag ours by over 10%, particularly when faster average query times are required. FLORA's hyperparameter tuning is also more delicate, with there being unsuccessful settings (where MAP grows very slowly with query time) very close to relatively successful ones.

4.4 Ablation study on hashcode training

Setup To perform ablation study on our proposed hashcode training method, we propose an alternative FHASH (UNTRAINED). This applies our Fourier features followed by a *data oblivious* random hyperplane LSH, without any data driven hashcode training.



(a) MSWEB-1 (b) MSWEB-2

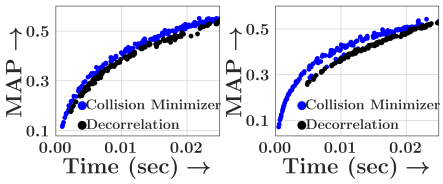
Figure 4: Effect of untrained RH

Results In Figure 4, we compare the complete design of our method, *i.e.*, FOURIERHASHNET and FHASH (UNTRAINED) against the untrained versions of RH+Hinge and DP-RH+Hinge. We make the following observations: **(1)** Benefit of Fourier transformation: The MAP vs time trade-off curve of FHASH (UNTRAINED), consistently dominates all the baselines across both datasets. **(2)** Benefit of hashcode training: Compared to FHASH (UNTRAINED), we observe that FOURIERHASHNET allows for significantly more choices of trade-off points, where higher MAP is required.

4.5 Ablation study on collision minimizer

Setup Here, we replace the collision minimizer in loss$(Q, X \mid \boldsymbol{W})$ (13) with decorrelation loss which encourages hashcodes to be dissimilar: $\Delta_1 = \sum_{x \neq y} |\tanh(\boldsymbol{W}\boldsymbol{z}_x)^\top \tanh(\boldsymbol{W}\boldsymbol{z}_y)|$, a commonly used loss in prior work [50, 15].

Results Figure 5 compares the performance of the two variations of the losses in terms of MAP, for MSWEB datasets. We observe that: **(1)** Our loss containing the collision minimizer term performs better than its variant which uses the decorrelation loss. In MSWEB-2, latter provides a MAP of 0.4 in 0.014 secs, which our loss achieves in 50% of the time. **(2)** Our method allows for greater freedom in navigating the performance vs average query time trade-off, as seen in MSWEB-2, as it is more spread out across the time axis.



(a) MSWEB-1 (b) MSWEB-2

Figure 5: Collision minimizer vs. decorrelation.

5 Conclusion

We have presented FOURIERHASHNET, an ALSH for asymmetric hinge distance, strongly motivated by text, image and graph retrieval applications. By converting hinge distance to a proposed dominance similarity and applying a suitable Fourier transform to the dominance similarity, we can estimate the

distance as an inner product over an importance-sampled spectrum, which further enables the use of a trainable LSH in the frequency domain. Experiments show that FOURIERHASHNET dramatically speeds up queries while preserving or improving retrieval accuracy. Since we propose a hashing method for order embeddings which effectively captures set containment, it would be intriguing to explore the possibility of extending this approach to box embeddings. Box embeddings are known to model more complex set operations like set overlap and set difference [39, 13], making them an interesting avenue for future research. One limitation of FOURIERHASHNET compared to simple symmetric LSHs is the increase in computational cost to compute the Fourier transform. One can explore other types of transformations to mitigate this cost.

Locality Sensitive Hashing in Fourier Frequency Domain For Soft Set Containment Search
(Appendix)

A Limitations of our work

(1) In Eq. (9), the probability distribution $p(\boldsymbol{\omega})$ is determined based on the proportionality constant $||\boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^{1..M})||$, and we set $p(\boldsymbol{\omega}) = \prod_{k\in[K]} p(\omega)$, where $p(\omega) \propto |\mathrm{Re}(S(\iota\omega))| + |\mathrm{Im}(S(\iota\omega))|$. We note that this choice of distribution is not informed by the data distribution. Doing so may further improve FOURIERHASHNET.

(2) In our approach, the dominance similarity function is represented as an expectation of inner products of functions in the frequency domain, as described in Eq. (10). The accuracy of this representation relies on the quality of Monte Carlo approximations, which is influenced by the number of $\boldsymbol{\omega}$ samples used. Our experimental findings, presented in Figure12, suggest that it may be necessary to generate up to 100 samples per dimension to reduce the approximation error to acceptable levels. A better choice of $p(\boldsymbol{\omega})$ may reduce the number of samples needed.

(3) During our experimental investigations, we discovered that the computation of our proposed dominance similarity score is approximately 7.5X slower, compared to the dot product similarity on our datasets. This aligns with earlier observations where matrix subtraction operations have been known to be significantly slower than dot product. This disparity in computation speed represents another potential limitation of our approach, which could be addressed by exploring alternative designs for the dominance similarity function.

B Example applications of soft set containment

Natural Language Inference (NLI) In (NLI) [4], q and x are sentences, regarded as sequences of words as items. A transformer network [14, 38] converts each sentence to an embedding vector. We infer $x \Longrightarrow q$ if $\boldsymbol{q} \leq \boldsymbol{x}$ [25]. Consider now a claim verification application that, given a claim as a query and a Web-scale corpus, needs to quickly retrieve passages that best support the given claim. This application exactly motivates FOURIERHASHNET.

Market basket Given a basket of supermarket items, we may query purchase logs for frequent supersets to make recommendations. The corpus contains itemsets purchased in the past, each is a 'document' x. The query q is the current basket. Hard set containment tests for $q \subset x$. However, we would like to 'soften' items in q (say, from one toothpaste brand to another. Each item has a short textual description, which is passed into BERT [38] and the `[CLS]` embedding read out as the item representation. An itemset is then embedded using a (suitably fine-tuned) set encoder (such as Deep Set), giving us $\boldsymbol{q}$ and $\boldsymbol{x}$. To the extent $\boldsymbol{q} \leq \boldsymbol{x}$, we regard the query basket as "soft-contained" in the document basket.

Knowledge graph (KG) completion Vendrov et al. [48] embedded types and entities in a KG to vectors such that if two types are related via t_1 is-subtype-of t_2 then $\vec{t_1} \leq \vec{t_2}$ was encouraged, and if entity e is-instance-of t, then $\vec{e} \leq \vec{t}$ was encouraged in suitable loss functions. Later, these *order* embeddings were generalized to *box* embeddings where is-subtype-of and is-instance-of were modeled as boxes in high dimensions contained in other boxes [10]. These models naturally motivate fast retrieval using dominance similarity.

Subgraph isomorphism search Here we expect a corpus graph x will be relevant if query graph q is a *subgraph* of x, i.e., that x has a subgraph that is *isomorphic* to q. In reality, we want to score highly corpus graphs that have subgraphs *almost* isomorphic to the query graph. A graph neural network (GNN) [24, 18] can build suitable contextual embeddings $\boldsymbol{q}$ and $\boldsymbol{x}$ for the entire graphs, which can be used to test for approximate subgraph isomorphism, *i.e.*, $\boldsymbol{q} \leq \boldsymbol{x}$. There are several applications of subgraph isomorphism search. In material and drug design, there are large molecule databases. A researcher wishes to predict properties of a new query molecule by retrieving similar molecules in the database. Each molecule is modeled as a modest-sized graph with nodes (atom, DNA bases) and edges (valence, etc.). In image search [21], the query q may be a graph fragment, e.g., $\langle$person, feeding, pet$\rangle$, and the goal is to find corpus graphs x where q is approximately a subgraph [30], *e.g.*, x can contain $\langle$man, feeding, dog$\rangle$.

C Further discussion on related work

In this section, we discuss existing work related to each of the three major components of our work—set embeddings, use of frequency domain for computing representations, and locality sensitive hashing.

Neural set embeddings Motivated by various machine learning questions that are better formalized by using a set of items as primitive, there has also been a recent line work on set embeddings. Zaheer et al. [56] consider models that act on sets and characterize the structure of such permutation-invariant models, but do not consider asymmetric query based measures e.g., containment. Skianis et al. [44] casts the similarity measurement between sets as a combinatorial flow problem, which in turn is approximated by a linear program. Lee et al. [26] propose the Set Transformer, a model that uses self-attention to model interactions among the input set elements. Our model is different from the existing line of work in focusing on asymmetric metrics that can measure containment, as well as in using the frequency domain representation of the metric to build a scalable LSH.

Application of frequency domain transformation in machine learning One of the most celebrated uses of the frequency domain representation was by [37], who proposed using random Fourier features for shift-invariant kernels. Since dot-product kernels are not shift-invariant, Bochner's theorem, a key tool in creating random Fourier features, does not apply. Hence, alternative random feature-based approximations have been proposed, primarily focusing on polynomial kernels [22, 36, 3, 45, 29]. All of the above work is on symmetric kernels. For our asymmetric dominance similarity function, we design a sampling distribution by taking into account the individual frequency-level coefficients of the Fourier representation.

Locality sensitive hashing The third main pillar of our work is locality sensitive hashing (LSH) which enables efficient retrieval. Answering queries using sketches or hashes in order to measure the similarity or containment of documents has a long history, pioneered by Broder [6]. In more recent years, semantic search or vector search, sometimes called "dense passage retrieval" [23] employing scalable near-neighbor search engines, has emerged as a credible, often more powerful, alternative [32, 28] to standard information retrieval schemes, as the vector embeddings can capture more nuanced contextualization and semantics. While there are a number of variants of LSH, including multi-probe [31], our presentation and experimentation focus on a single-probe setting in which the hashing hyperplanes are learned from the data. In particular, we build upon the asymmetric hash constructions in Shrivastava and Li [42] and Neyshabur and Srebro [33].

D Proofs of the technical results

D.1 Proof of Proposition 3.1

Proof. Consider two functions $\textsc{Box}_{a,b}$ (for arbitrary positive constants a, b) and $\textsc{ReLU}_T$ defined as follows:

$$\textsc{Box}_{a,b}(t) = \begin{cases} a \text{ if } -b \leq t \leq b \\ 0 \text{ else} \end{cases} \tag{15}$$

$$\textsc{ReLU}_T(t) = \begin{cases} t \text{ if } 0 \leq t \leq T \\ 0 \text{ else} \end{cases} \tag{16}$$

Observe that s can be written in terms of these new functions as follows: $s(t) = \textsc{Box}_{T,T}(t) - \textsc{ReLU}_T(t)$. By linearity of Fourier transform,

$$S(\iota\omega) = \mathcal{F}_{\textsc{Box}_{T,T}}(\iota\omega) - \mathcal{F}_{\textsc{ReLU}_T}(\iota\omega) \tag{17}$$

where, $\mathcal{F}_f(\iota\omega)$ denotes Fourier transform of $f(t)$ for any function f. Now let us compute $\mathcal{F}_{\textsc{ReLU}_T}(\iota\omega)$.

$$\mathcal{F}_{\textsc{ReLU}_T}(\iota\omega) = \frac{1}{2\pi}\int_0^T te^{-\iota\omega t}dt \tag{18}$$

$$= -\frac{1}{2\pi\omega^2} + \frac{e^{-\iota\omega T}}{2\pi\omega^2} + \frac{\iota T e^{-\iota\omega T}}{2\pi\omega} \tag{19}$$

$$= -\frac{1}{2\pi\omega^2} + \frac{\cos(\omega T) - \iota\sin(\omega T)}{2\pi\omega^2} + \frac{T(\iota\cos(\omega T) + \sin(\omega T))}{2\pi\omega} \tag{20}$$

$$= \frac{-2\sin^2(\omega T/2)}{2\pi\omega^2} + \frac{T\sin(\omega T)}{2\pi\omega} - \iota\frac{\sin(\omega T)}{2\pi\omega^2} + \iota\frac{T\cos(\omega T)}{2\pi\omega} \tag{21}$$

Since $\textsc{Box}_{T,T}(t)$ is a rectangular pulse, its Fourier transform is

$$\mathcal{F}_{\textsc{Box}_{T,T}}(\iota\omega) = 2T\frac{\sin(\omega T)}{2\pi\omega} \tag{22}$$

Substituting the above Eqs. (21) and (22) into Eq. (17), we get $S(\iota\omega)$ as follows.

$$S(\iota\omega) = \underbrace{2T\frac{\sin(\omega T)}{2\pi\omega}}_{\textsc{Box}_{T,T}(\iota\omega)} - \underbrace{\left[\frac{-2\sin^2(\omega T/2)}{2\pi\omega^2} + \frac{T\sin(\omega T)}{2\pi\omega} - \iota\frac{\sin(\omega T)}{2\pi\omega^2} + \iota\frac{T\cos(\omega T)}{2\pi\omega}\right]}_{G(\iota\omega)} \tag{23}$$

$$= \overbrace{T\frac{\sin(\omega T)}{2\pi\omega} + 2\frac{\sin^2(\frac{\omega T}{2})}{2\pi\omega^2}}^{\mathrm{Re}(S(\iota\omega))} + \iota\underbrace{\left[\frac{\sin(\omega T)}{2\pi\omega^2} - \frac{T\cos(\omega T)}{2\pi\omega}\right]}_{\mathrm{Im}(S(\iota\omega))} \tag{24}$$

□

E Additional details about the experimental setup

E.1 Dataset Generation

We obtain the MSWEB[1] and MSNBC[2] datasets from the UCI Machine Learning repository. Both of the datasets contain anonymized logs of real world user web activity. Each data item in MSWEB is a set of text snippets denoting areas of the website *www.microsoft.com* visited by an user within a specified time frame. Similarly, MSWEB consists of multi-sets denoting user page requests under various news categories at *www.msnbc.com*. Overall, we regard each data set as a collection of items, each item being a bag of words. For each data set, we sample a subset of items and designate them as queries, and the remaining items are designated as corpus items. The (binary) query-corpus relevance for MSWEB-1 and MSWEB-2 are governed by set containment, while for MSNBC-1 and MSNBC-2 we use multi-set (bag) containment. I.e., x is relevant for q iff $q \subseteq x$. To test the ability of FOURIERHASHNET to retrieve semantically similar items close to the gold items, we report not only on MAP based on gold labels but also scores of the top-10 candidates (Figure 3). The dataset characteristics are summarized in Table 6. We create datasets which differ greatly in terms of corpus size (10734 for MSWEB-1 and MSWEB, 110790 for MSNBC-1 and MSNBC-2), as well as span a range of average query selectivity between 1.7×10^{-4} and 3.3×10^{-3}. We set aside 100 query graphs each for training and validation, and use the remaining 300 for testing.

Dataset	$\lvert Q\rvert$	$\lvert X\rvert$	$\frac{\sum_{q\in Q x\in X}\text{rel}(q,x)]}{\lvert Q\rvert}$	$\frac{\min_{q\in Q x\in X}\text{rel}(q,x)]}{\lvert Q\rvert}$	$\frac{\max_{q\in Q x\in X}\text{rel}(q,x)]}{\lvert Q\rvert}$	$\frac{\sum_{q\in Q x\in X}\text{rel}(q,x)]}{\lvert Q\rvert\lvert C\rvert}$
MSWEB-1	500	10734	35.624	9	327	0.0033
MSWEB-2	500	10734	20.392	9	49	0.0019
MSNBC-1	500	110790	24.09	9	44	0.00022
MSNBC-2	500	110790	19.78	9	34	0.00017

Table 6: Dataset statistics. From left to right: Datasets name, number of queries, number of corpus, the average number of relevant corpus items per query, the minimum num of relevnt corpus items per query, the maximum number of corpus items per query and the average query selectivity.

E.2 Learning Representations for q and x for the baselines

During experiments for Section 4.2, in each of the baseline method (Cosine similarity, Dot product and Weighted Jaccard), we use the respective similarity scoring functions, and minimize a pairwise ranking loss on the gold relevance labels to learn the deep set network. We observed that the pairwise ranking loss performs better than the BCE loss for the baselines. The margin enabled pairwise ranking loss is specified as

$$\text{Loss} = \sum_{q\in Q}\sum_{\substack{x\in X_{q\checkmark}\\ x'\in X_{q\text{✗}}}} \big[\text{margin} + \text{sim}(q,x') - \text{sim}(q,x)\big]_+. \tag{25}$$

where sim is the choice of similarity scoring baseline, $X_{q\checkmark}, X_{q\text{✗}}$ are the set of relevant and irrelevant corpus items for the query q. We use the best performing margin among $\{1, 0.1\}$.

E.3 Sampling from arbitrary distribution

One key component of FOURIERHASHNET is sampling $\boldsymbol{\omega}^{1\ldots M} \sim p(\boldsymbol{\omega})$. We have chosen to set $p(\omega) \propto |\text{Re}(S(\iota\omega))| + |\text{Im}(S(\iota\omega))|$, with the support set between $p(\omega)$ between $[-100, 100]$. The samples are drawn using Inverse Transform Sampling.

E.4 Details about fourier transformation network

In our experiments, we generate $M = 10$ samples for $\boldsymbol{\omega}$. The neural networks ϕ_q and ϕ_x are linear layers which output 10 dimensional transformed Fourier representations $\boldsymbol{z}_q = \phi_q(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M}))$ and $\boldsymbol{z}_x = \phi_x(\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M}))$. These are trained using the BCE loss specified in Eq. (11).

E.5 Details about hashcode generation network

We use the same hashcode training procedure for FOURIERHASHNET, as well as the DP-RH and RH baselines. In all three cases, we generate 64 dimensional hashcodes. For FOURIERHASHNET, the random hyperplanes $\boldsymbol{W}$ are trained on 10 dimensional trained Fourier representations $\boldsymbol{z}_q, \boldsymbol{z}_x$. For RH we use the original embeddings $\boldsymbol{q}$ and $\boldsymbol{x}$. For DP-RH, we use the augmented embeddings $g(q) = \text{sign}(\boldsymbol{W}[0, \boldsymbol{q}/||\boldsymbol{q}||])$ and $h(x) = \text{sign}(\boldsymbol{W}[\sqrt{T^2 - ||\boldsymbol{x}||^2}, \boldsymbol{x}])$.

[1] `https://archive.ics.uci.edu/ml/machine-learning-databases/msweb-mld`

[2] `https://archive.ics.uci.edu/ml/machine-learning-databases/msnbc-mld/`

E.6 AP and MAP measurements

Suppose a query q is associated with $N_{q\oplus}$ relevant corpus items (as judged by humans). Suppose the system provides a ranking over all N corpus items, and the relevant items occur at ranks $r_1, \ldots, r_{N_{q\oplus}}$. Then AP for query q is defined as $(1/N_{q\oplus}) \sum_{i=1}^{N_{q\oplus}} (i/r_i)$. This is because, up to position r_i, we have seen i relevant items, which means we can shorthand i/r_i as prec@i (precision at i). We can rewrite the sum as $\frac{1}{N_{q\oplus}} \sum_{r=1}^{N} \text{prec@}r \times \text{rel@}r$, where N is the size of the whole corpus, and $\text{rel@}r$ is the (0/1) relevance of the item at rank r. In case the retrieval algorithm does not assess all N corpus items, but stops with the best L hash buckets, which contain, say, N'_q items, we should use the following formula for AP: $\frac{1}{N_{q\oplus}} \sum_{r=1}^{N'_q} \text{prec@}r \times \text{rel@}r$. Note that we should still divide by $N_{q\oplus}$, otherwise an algorithm that maps the query to a densely relevant but small bin, which fails to retrieve most relevant items, might be rewarded in an unfair manner.

E.7 Top-10 score measurement

In Appendix F, we provide additional experiments where we compare FOURIERHASHNET with all the baselines not only in terms of MAP, but also in terms of the Top-10 score. We use the sum of Top-10 scores normalized in $[0, 1]$ via the sigmoid transformation used in Eq. (14): $\text{Top-10}(q) = \sum_{x \in \text{Top-10}(N'_q)} \sigma(-d(q, x))$. Any hashing protocol is expected to retrieve the corpus items, which have the highest similarity scores with respect to any given query. The Top-10 score evaluates it independently of how the retrieved items match with true relevant items. Therefore, the Top-10 scores provide an evaluation mechanism that is independent of the gold relevance labels and solely relies on the scores dictated by the trained embeddings. This offers a valuable means of assessing performance without being influenced by subjective human judgments.

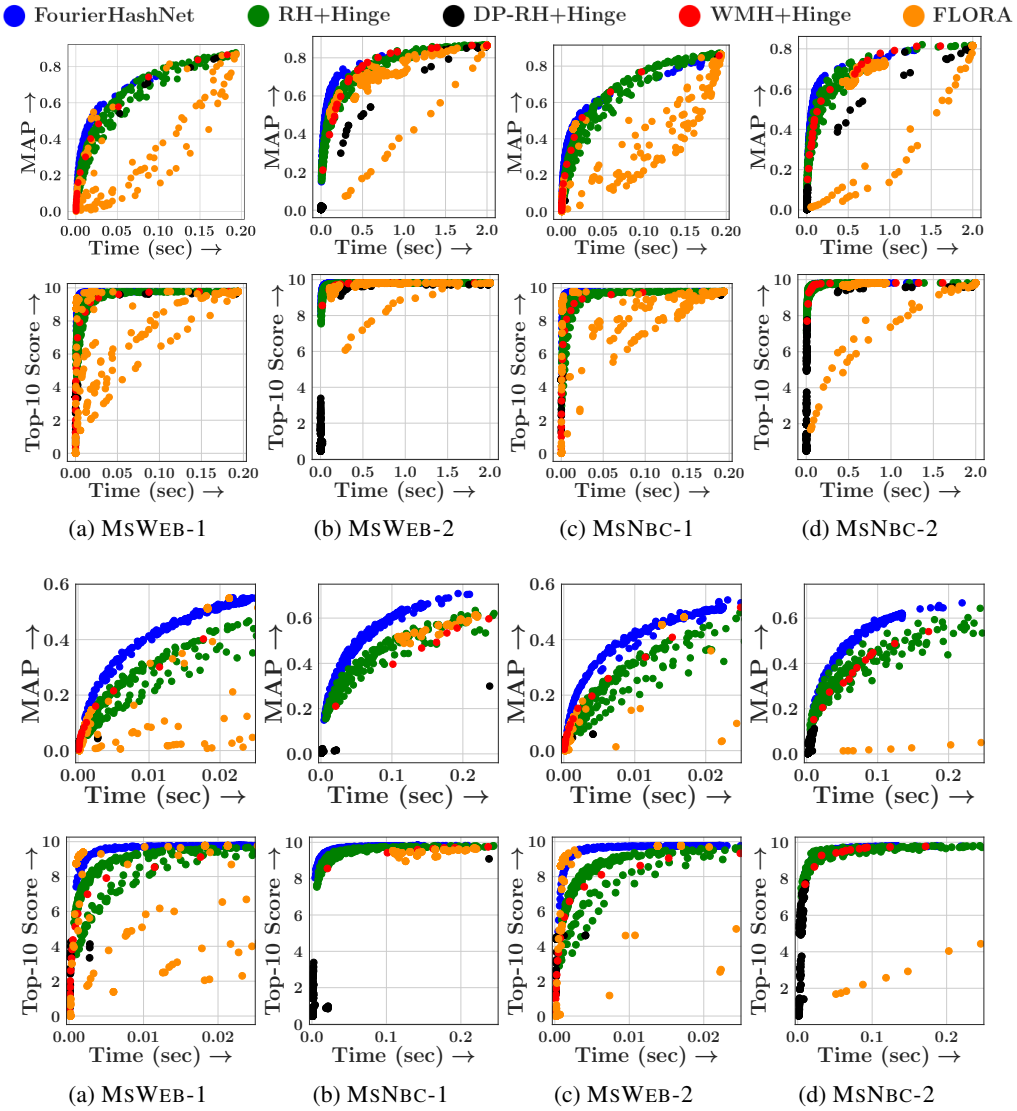
E.8 Licenses

We utilize a publicly available pre-trained sentence transformer model [38], which is licensed under the Apache License 2.0. Additionally, we employ the DrHash toolkit [54] for various implementations of the baseline Weighted Minhash (WMH) algorithms. The DrHash toolkit is publicly available under the MIT License. We duly acknowledge the original authors of the baseline methods in our citations.

F Additional experiments

F.1 Applying baseline LSHs on hinge distance guided embeddings

In continuation of the results reported in Figure 3, in Figure 7, we present the complete view as well as the zoomed versions of the trade-off curves for all datasets.



(a) MSWEB-1 (b) MSWEB-2 (c) MSNBC-1 (d) MSNBC-2

(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

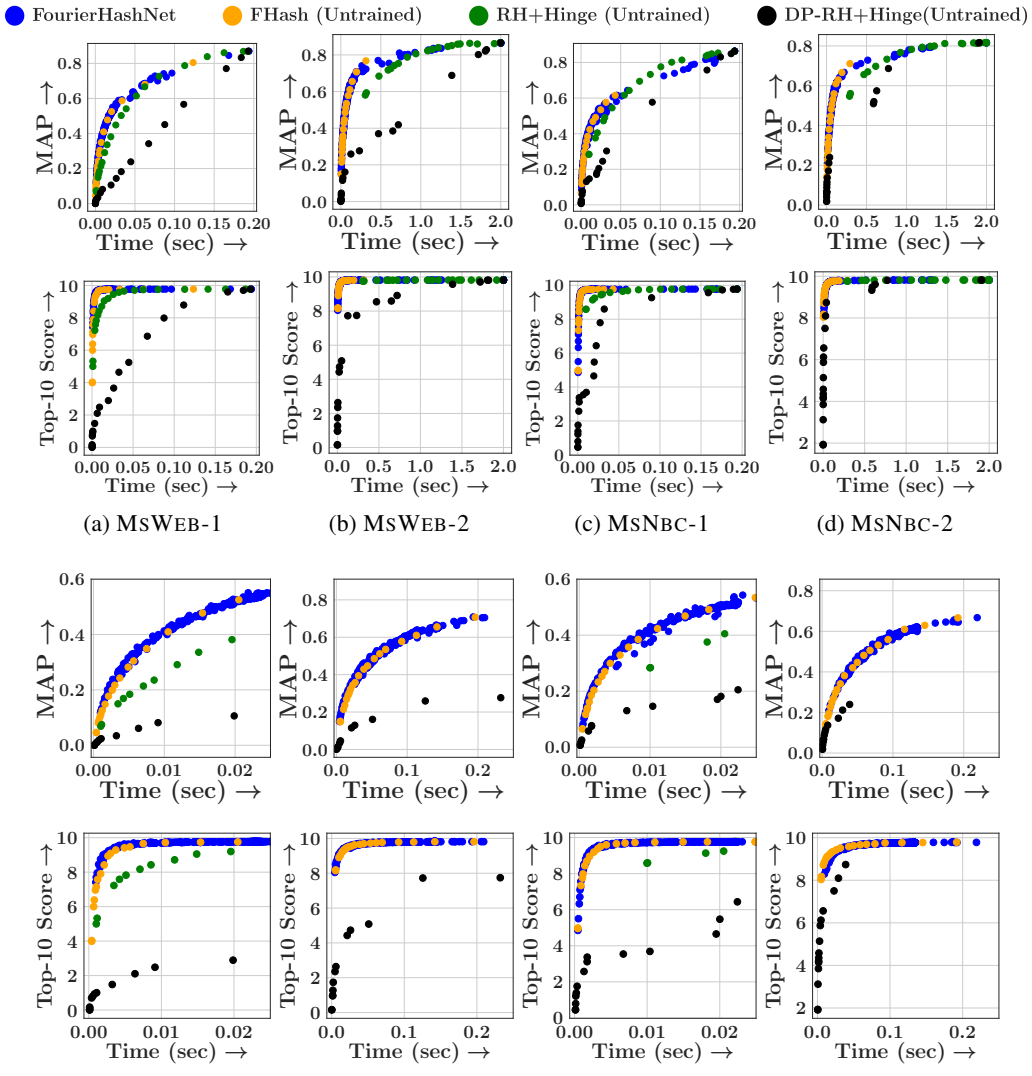
Figure 7: Trade-off between average query time and accuracy (MAP and Top-10 scores) for MSWEB and MSNBC datasets (First two rows: complete view across full time axis, last two rows: Zoomed version of first two rows until the average query time there is $\geq$ 10X speedup compared to exhaustive search). We apply different LSH methods on hinge distance guided embeddings similar to Figure 3, then use the hinge distance to finally rank the retrieved items.

Beyond the observations made in Figure 3, we make the following additional observations.
(1) The complete view for both Top-10 score and MAP score, clearly demonstrates FLORA's high sensitivity to hyperparameter tuning. FLORA is seen to have the highest variance in scores for any given time budget across all the baselines. In terms of Top-10 score, while FLORA is marginally ahead of FOURIERHASHNET in a few instances in the MSWEB datasets, it is significatly outperformed by FOURIERHASHNET in the MSNBC datasets. This may be due to the significantly higher average query selectivity in the MSNBC datasets.
(2) In terms of Top-10 score, FOURIERHASHNET achieves the maximum possible value 4$\times$ faster than the nearest competitor RH+Hinge, in MSWEB-1 and MSWEB-2. In terms of MAP score, RH+Hinge and WMH+Hinge achieve a maximum MAP of 0.5 in MSWEB-1 and MSWEB-2, and

0.62 in MSNBC-1 and MSNBC-2. However, FOURIERHASHNET achieves the same MAP values 1.33× faster in MSWEB-1 and MSWEB-2, and 2× faster in MSNBC-1 and MSNBC-2.
(3) Interestingly, the gap between FOURIERHASHNET and RH+Hinge seems to widen for MSWEB-2, when compared to MSWEB-1. This is possibly due to the presence of several queries in MSWEB-1, which have ≥300 relevant corpus items. This affords RH+Hinge a greater opportunity to fetch high scoring items, which is not the case in MSWEB-2.

F.2 Ablation study on hashcode training

In continuation of the results reported in Figure 4, in Figure 8, we present the complete view as well as the zoomed versions of the trade-off curves for all datasets.



(a) MSWEB-1 (b) MSWEB-2 (c) MSNBC-1 (d) MSNBC-2

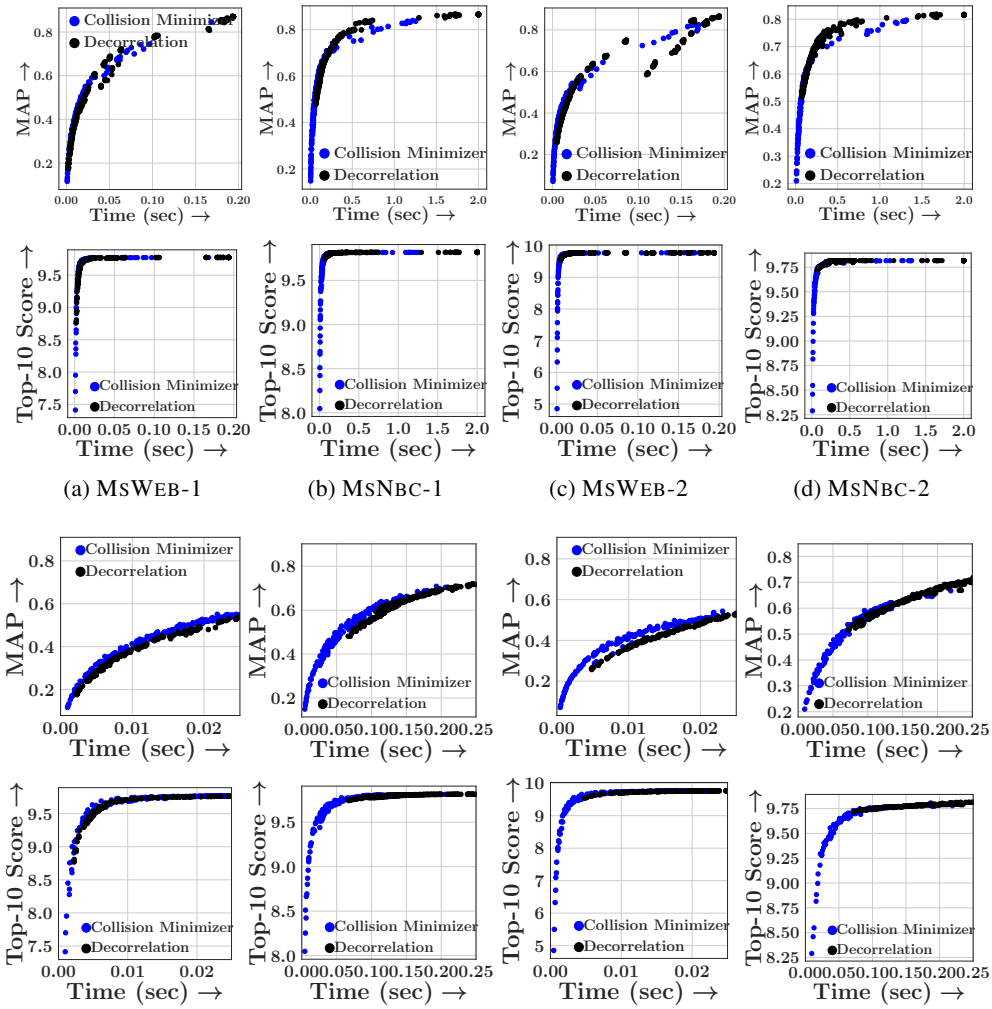
(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

Figure 8: Trade-off between average query time and accuracy (MAP and Top-10 scores) for MSWEB and MSNBC datasets (First two rows: complete view across full time axis, last two rows: Zoomed version of first two rows until the average query time there is ≥ 10X speedup compared to exhaustive search). We compare FHASH (UNTRAINED) against the untrained versions of RH+Hinge and DP-RH+Hinge, as well as against FOURIERHASHNET.

Beyond the observations made in Figure 4, we make the following additional observations.
(1) In terms of both Top-10 score and MAP score, FHASH (UNTRAINED) clearly outperforms both RH+Hinge and DP-RH+Hinge, across all four datasets. This strongly highlights the advantage of our Fourier feature generation method.

(2) In every setup, FOURIERHASHNET enables a wider range of options for accuracy score vs retrieval time trade-off.

F.3 Ablation study on collision minimizer (13)



(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

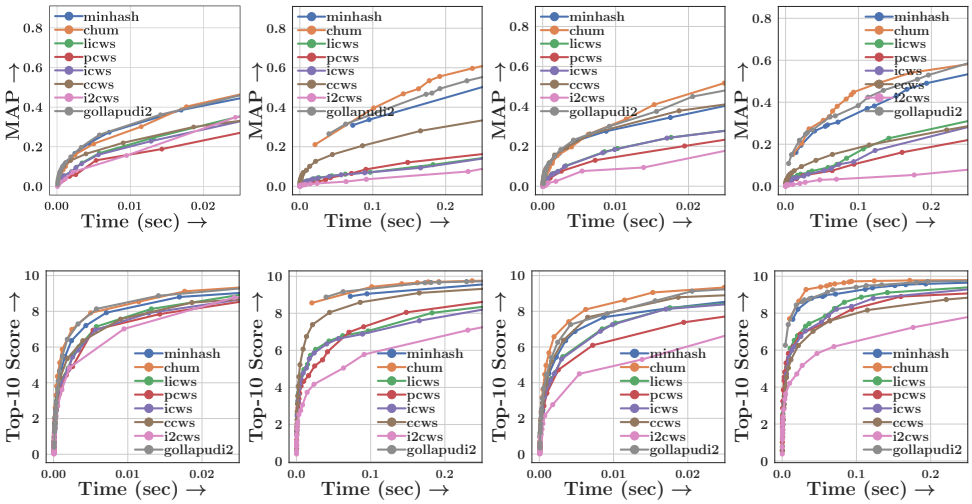
(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

Figure 9: Trade-off between mean query time and accuracy (MAP and Top-10 scores) for MSWEB and MSNBC datasets (First two rows: complete view across full time axis, last two rows: Zoomed version of first two rows until the mean query time there is $\geq$10X speedup compared to exhaustive search). We compare our hashcode training loss loss$(Q, X \mid \boldsymbol{W})$ (13), against a variant which replaces the collision minimizer component Δ_1 with a decorrelation loss $\sum_{x \neq y} |\tanh(\boldsymbol{W}\boldsymbol{z}_x)^\top \tanh(\boldsymbol{W}\boldsymbol{z}_y)|$.

In continuation of the results reported in Figure 5, we present the complete view as well as the zoomed versions of the trade-off curves for all datasets. Beyond the observations made in Figure 5, we make the following additional observations.
(1) In MSWEB-2, the alternative variant shows a sudden plunge in MAP performance trade-off at around 0.1 seconds. This type of discontinuous drop is not observed in any of our cases.
(2) In MSWEB-1, there is a variation of 0.1 MAP at around 0.05 seconds. Such high variability is not observed for any of our trade-off curves.

F.4 Identifying best performing Weighted Minhash algorithm for our datasets

For implementation of baseline Weighted Minhash (WMH) algorithm, we use the best performing WMH implementation available in the DrHash toolkit [54] to obtain the hashcodes. We comapare across the 8 available baselines in the toolkit: minhash [7], chum [12], icws [20], pcws [52], licws [27], ccws [51], i2cws [53] and gollapudi2 [16].

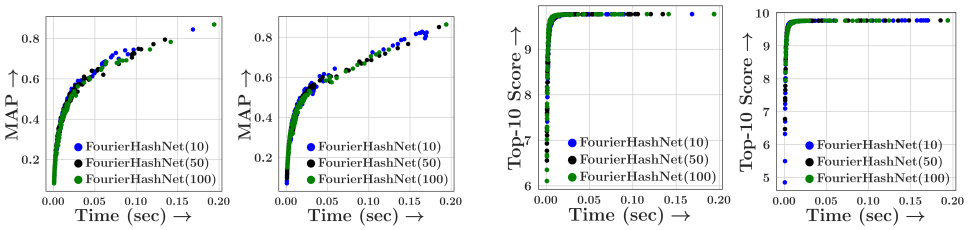


(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

Figure 10: We compare performance of Weighted Minhash variations, in terms of trade-off between mean query time and accuracy (MAP and Top-10 scores) for MSWEB and MSNBC datasets, until the query time there is $\geq$10X speedup compared to exhaustive search.

We make the following observations.
(1) Across all four of our datasets, for both MAP and Top-10 score, the top 3 performers are minhash, gollapudi2 and chum. The remaining algorithms are often significantly worse in performance, as can be seen for MAP in all 4 datasets and for Top-10 in MSNBC-2.
(2) Among the top 3 performers, chum is seen to be the best perform in terms for both MAP and Top-10 score, in MSWEB-2, MSNBC-1 and MSNBC-2. In MSWEB-1, chum is tied with minhash and gollapudi2 for the top position.
Driven by these observations, we choose chum as the representative baseline for WMH in our experiments.

F.5 Effect of M, number of samples of ω on FOURIERHASHNET performance



(a) MSWEB-1 (b) MSWEB-1 (a) MSWEB-1 (b) MSWEB-2

Figure 11: Effect of M, number of ω samples, on the trade-off between mean query time and accuracy (MAP and Top-10 scores) for MSWEB datasets.

Here we check the impact of varying the number of samples (M) for $\boldsymbol{\omega}$. We consider three different values of M, *i.e.*, 10, 50 and 100, for generating the fourier features $\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M})$ and $\boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M})$ which are then fed into the neural networks ϕ_q and ϕ_x for learning the transformed Fourier representations $\boldsymbol{z}_q$ and $\boldsymbol{z}_x$, using the BCE loss specified in Eq. (11). Finally, we train the random hyperplanes $\boldsymbol{W}$ and check the MAP and Top-10 score performances for the three variations - FOURIERHASHNET(10), FOURIERHASHNET(50) and FOURIERHASHNET(100). We observe that the final performance trade-off of both MAP and Top-10 scores, remains roughly the same across all three variants. This shows that trainable Fourier transformation is able to compensate for the quality of Monte Carlo approximations affected by the number of $\boldsymbol{\omega}$ samples M.

Next, we investigate how well the MC estimates of the Fourier features approximate the dominance similarity function $\text{sim}(q, x)$. Here, we set the dimension of $\boldsymbol{q}$ and $\boldsymbol{x}$ as $K = 1$. We set $T = 20$ and we sample $\boldsymbol{q}, \boldsymbol{x} \sim \text{Unif}[-20, 20]$. Finally, we compute $\widehat{\text{sim}}_M(q, x) = \frac{||\boldsymbol{F}_\bullet(\iota\boldsymbol{\omega}^{1..M})||^2}{M} \cos(\boldsymbol{F}_q(\iota\boldsymbol{\omega}^{1..M}), \boldsymbol{F}_x(\iota\boldsymbol{\omega}^{1..M}))$ and measure the variation of $\epsilon_{\text{sim}} = \mathbb{E}_{\boldsymbol{q},\boldsymbol{x}\sim\text{Unif}[-20,20]}[|\text{sim}(q, x) - \widehat{\text{sim}}_M(q, x)|]$ with M, the number of samples of ω. Figure 12 summarizes the results, which shows as M increases, ϵ_{sim} decreases.

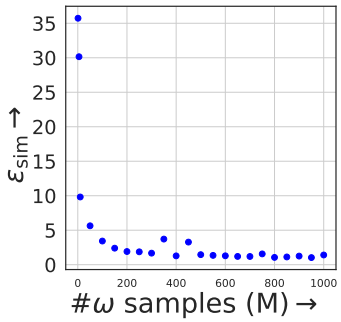


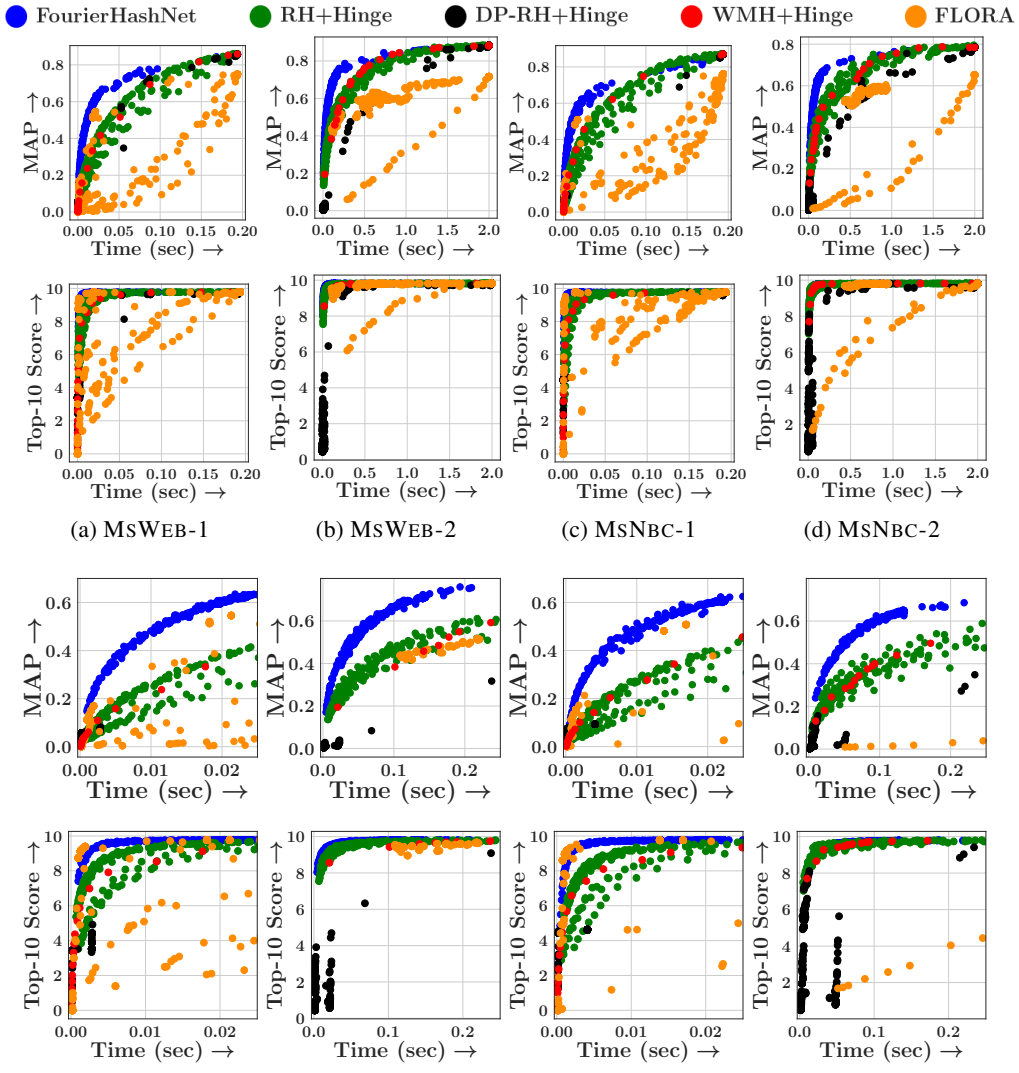
Figure 12: Variation of $\epsilon_{\text{sim}} = \mathbb{E}_{\boldsymbol{q},\boldsymbol{x}\sim\text{Unif}[-20,20]}[|\text{sim}(q, x) - \widehat{\text{sim}}_M(q, x)|]$ with M, the number of ω samples.

F.6 Applying baseline LSHs on hinge distance guided embeddings with noisy labels

In certain applications, the accuracy of ground truth labels can be compromised by noise or subjective human judgments of relevance. We evaluate the performance of FOURIERHASHNET and the baselines in a noisy label setup to test its robustness.

Starting with the hinge distance guided embeddings, we initially rank the corpus items based on their dominance similarity scores. Subsequently, we intentionally flip the labels of the bottom-ranking 10% of positive labels to negative, and an equal number of highest ranked negatively labeled items to positive. This simulation reflects a plausible scenario since the lowest ranked positive items and the highest ranked negative items are particularly susceptible to misclassification in real-world settings. Furthermore, this approach ensures that the average query selectivity for each dataset remains unchanged.

As before, we apply different the LSH methods on hinge distance guided embeddings, *viz*, RH+Hinge, DP-RH+Hinge, WMH+Hinge, FLORA and FOURIERHASHNET; and, then use the hinge distance to finally rank the retrieved items.



(a) MSWEB-1 (b) MSWEB-2 (c) MSNBC-1 (d) MSNBC-2

(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

Figure 13: Trade-off between average query time and accuracy (MAP and Top-10 scores) for MSWEB and MSNBC datasets (First two rows: complete view across full time axis, last two rows: Zoomed version of first two rows until the average query time there is $\geq$ 10X speedup compared to exhaustive search). We apply different LSH methods on hinge distance guided embeddings similar to Figure 7 , and then use hinge distance to rank retrieved items. Evaluations are conducted using noisy labels.

We make the following observations.
(1) In terms of MAP score, FOURIERHASHNET continues to outperform all other baselines across all four datasets. Furthermore, when comparing the performance in the presence of noise, as depicted in Figure 13, to the corresponding results obtained in the noiseless setting illustrated in Figure 7, we observe that FOURIERHASHNET outperforms all its competitors by a significantly higher margin in the presence of noise.

(2) In terms of Top-10 score, we note that the results in Figure 13 for the noisy setup are identical to the results presented in the noiseless setting shown in Figure 7. This observation indicates that the evaluation based on Top-10 score is unaffected by label noise. This supports the argument made in Appendix E.7 that Top-10 score evaluation enables a more subjective assessment of performance, focusing on the quality of the embeddings themselves.

F.7 Abridged Proof of ALSH for OpenReview

F.8 Proof of ALSH

Definition F.1 (Asymmetric Locality Sensitive Hashing (ALSH) [33])**.** An asymmetric LSH is (S_0, cS_0, p_1, p_2)-ALSH for a similarity function $\text{sim}(\bullet, \bullet)$ over $\mathcal{Q}$, $\mathcal{X}$ if we have two different distributions over mappings $\mathcal{G}$ and $\mathcal{H}$ such that, with $p_1 > p_2$ and $c < 1$,

- if $\text{sim}(q, x) \geq S_0$ then $\Pr_{g\sim\mathcal{G}, h\sim\mathcal{H}}[g(q) = h(x)] \geq p_1$
- if $\text{sim}(q, x) \leq cS_0$ then $\Pr_{g\sim\mathcal{G}, h\sim\mathcal{H}}[g(q) = h(x)] \leq p_2$.

Written in a way so that easily can be adapted to openreview.

We write the proof that our procedure is indeed an ALSH.

Theorem F.2. *Assume that $sim(q, x) > sim_{min} > 0$ for all q, x; $\cos^{-1}$ is Lipschitz in our context with Lipschitz constant $L_{\cos}$; $p(\omega_k^j) \propto |Re(S(\omega_k^j))| + |Im(S(\omega_k^j))|$ with I being the proportionality constant. Then, the mapping $g(q)[i] = \text{sign}(\boldsymbol{w}_i^\top \boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M}))$ and $h(x)[i] = \text{sign}(\boldsymbol{w}_i^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M}))$ where $\boldsymbol{w_i} \sim N(0, \mathbb{I})$ constitutes an ALSH if*

$$M > \frac{4L_{\cos}^2}{K\pi^2 \left[\cos^{-1}\left(\frac{c\cdot sim_{min}}{KI}\right) - \cos^{-1}\left(\frac{sim_{min}}{KI}\right)\right]^2}, \tag{26}$$

Justification about assumptions $sim(q, x) > sim_{min} > 0$: T in Eq. 4 can be made large to ensure this. $\cos^{-1}$ is Lipschitz in our context with Lipschitz constant $L_{\cos}$: This is in general not true, because $d\cos^{-1}(x)/dx = 1/\sqrt{1-x^2}$ can be unbounded near $x \in \pm 1$. However, in our case, this will be attained if $\boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M}) = \pm \boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M})$. Eq 7 suggests that it would happen iff $\cos\omega_k q[k] = \pm\cos\omega_k x[k]$ and $\cos\omega_k q[k] = \pm\sin\omega_k x[k]$. For a general q and x, the above conditions are infeasible or give a fixed value of ω_k. However, the likelihood that ω_k takes that fixed value is very low since ω_k is drawn from a continuous distribution.

Proof We have: $||\boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M})||_2^2 = ||\boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M})||_2^2 = \sum_{i\in[M], k\in[K]} \frac{|Re(S(\omega_k^j))| + |Im(S(\omega_k^j))|}{p(\omega_k^j)} = MKI$. We use the following relationship: $Pr_{g,h}[g(q) = h(x)] = \mathbb{E}_{\boldsymbol{\omega^j}}[Pr_{g,h}[g(q) = h(x)|\boldsymbol{\omega}^j]]$.

$$Pr_{g,h}[g(q) = h(x)|\boldsymbol{\omega}^i] = 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{\boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M})^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M})}{||\boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M})||||\boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M})||}\right) \tag{27}$$

$$= 1 - \frac{1}{\pi}\cos^{-1}\left(\int_\omega \underbrace{\frac{\boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}}{||\boldsymbol{F}_q(\boldsymbol{\omega}||||\boldsymbol{F}_x(\boldsymbol{\omega})||}}_{KI}\right) \tag{28}$$

$$- \frac{1}{\pi}\cos^{-1}\left(\frac{\boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M})^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M})}{MKI}\right) + \frac{1}{\pi}\cos^{-1}\left(\int_\omega \underbrace{\frac{\boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}}{||\boldsymbol{F}_q(\boldsymbol{\omega}||||\boldsymbol{F}_x(\boldsymbol{\omega})||}}_{KI}\right) \tag{29}$$

$$= 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{sim(q, x)}{KI}\right) \tag{30}$$

$$- \frac{1}{\pi}\cos^{-1}\left(\frac{\boldsymbol{F}_q(\boldsymbol{\omega}^{1\cdots M})^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\cdots M})}{MKI}\right) + \frac{1}{\pi}\cos^{-1}\left(\int_\omega \frac{\boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}}{KI}\right) - - - - - - - (A) \tag{31}$$

Note that:

$$- \cos^{-1}\left(\frac{\boldsymbol{F}_q(\boldsymbol{\omega}^{1\ldots M})^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\ldots M})}{MKI}\right) + \cos^{-1}\left(\int_\omega \frac{\boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}}{KI}\right) \tag{32}$$

$$\leq \frac{L_{\cos}}{KI}\left|\boldsymbol{F}_q(\boldsymbol{\omega}^{1\ldots M})^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\ldots M})/M - \int_\omega \boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}\right| \tag{33}$$

$$= \frac{L_{\cos}}{KI}\left|\sum_{j\in[M]} \boldsymbol{F}_q(\boldsymbol{\omega}^j)^\top \boldsymbol{F}_x(\boldsymbol{\omega}^j)/M - \int_\omega \boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}\right| \tag{34}$$

Taking expectation wrt $\boldsymbol{\omega}$

$$\mathbb{E}\left[- \cos^{-1}\left(\frac{\boldsymbol{F}_q(\boldsymbol{\omega}^{1\ldots M})^\top \boldsymbol{F}_x(\boldsymbol{\omega}^{1\ldots M})}{MKI}\right) + \cos^{-1}\left(\int_\omega \frac{\boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}}{KI}\right)\right] \tag{35}$$

$$\leq \frac{L_{\cos}}{KI}\mathbb{E}\left|\sum_{j\in[M]} \boldsymbol{F}_q(\boldsymbol{\omega}^j)^\top \boldsymbol{F}_x(\boldsymbol{\omega}^j)/M - \int_\omega \boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega}) p(\boldsymbol{\omega}) d\boldsymbol{\omega}\right| \tag{36}$$

$$\leq \frac{L_{\cos}}{KI}\left(\text{Variance}\left[\sum_{j\in[M]} \boldsymbol{F}_q(\boldsymbol{\omega}^j)^\top \boldsymbol{F}_x(\boldsymbol{\omega}^j)/M\right]\right)^{1/2} \qquad (\mathbb{E}[|Z|] \leq \sqrt{\mathbb{E}[|Z|^2]}) \tag{37}$$

$$= \frac{L_{\cos}}{KI\sqrt{M}}\left(\text{Variance}\left[\boldsymbol{F}_q(\boldsymbol{\omega})^\top \boldsymbol{F}_x(\boldsymbol{\omega})\right]\right)^{1/2} \tag{38}$$

$$= \frac{L_{\cos}}{KI\sqrt{M}}\sqrt{K}\left(\text{Variance}\left[\boldsymbol{F}_q(\omega_k)^\top \boldsymbol{F}_x(\omega_k)\right]\right)^{1/2} \leq \frac{L_{\cos}}{\sqrt{KM}} \tag{39}$$

The last inequality follows from bound on the variance due the following:

$$\boldsymbol{F}_q(\omega_k)^\top \boldsymbol{F}_x(\omega_k) \tag{40}$$

$$= \boldsymbol{S}_q(\omega_k)^\top \boldsymbol{S}_x(\omega_k)/p(\omega_k) \quad \text{(Eq 9 in the paper)} \tag{41}$$

$$= \frac{Re[\boldsymbol{S}(\omega)]\cos\omega_k(q[k]-x[k]) - Im[\boldsymbol{S}(\omega)]\sin\omega_k(q[k]-x[k]))}{(1/I)[|Re[\boldsymbol{S}(\omega)]| + |Im[\boldsymbol{S}(\omega)]|]} \tag{42}$$

$$\leq I\frac{|Re[\boldsymbol{S}(\omega_k)]| + |Im[\boldsymbol{S}(\omega_k)]|}{|Re[\boldsymbol{S}(\omega)]| + |Im[\boldsymbol{S}(\omega)]|} = I \tag{43}$$

Putting (39) into (A) we have

$$\mathbb{E}[Pr_{g,h}[g(q) = h(x)|\boldsymbol{\omega}^j]] \leq p_2 = 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{sim(q,x)}{KI}\right) + L_{\cos}/\pi\sqrt{KM} \tag{44}$$

Similarly, putting a lower bound using Lipschitz constant $L_{\cos}$, we will have:

$$\mathbb{E}[Pr_{g,h}[g(q) = h(x)|\boldsymbol{\omega}^j]] \geq p_1 = 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{sim(q,x)}{KI}\right) - L_{\cos}/\pi\sqrt{KM} \tag{45}$$

Now if M is as large as

$$M > \frac{4L_{\cos}^2}{K\pi^2\left[\cos^{-1}\left(\frac{c\cdot sim_{min}}{KI}\right) - \cos^{-1}\left(\frac{sim_{min}}{KI}\right)\right]^2}, \tag{46}$$

then we have $p_1 > p_2$

>* It would be much more interesting (and appeal to a wider audience) if this algorithm can be generalized into a broader "truncate - transform - sample" template. Such a template might apply directly to other applications. > *Is it possible to generalize the sampling process (bottom of page 5) into a generic algorithm with theoretical guarantees

Indeed, our algorithm can be generalized to a broader context which does include a wide variety of scoring functions, including Box embedding based volume scores, facility location scores used by colbert, etc. We elaborate them in the following.

First, we note that our algorithm can be extended to any shift-invariant scoring function. By shift invariant scoring function, we mean scores of the form: $a(\boldsymbol{q} - \boldsymbol{x})$. Note that if we shift the query and

corpus embedding by the same vector $\boldsymbol{\delta}$, the score remains the same. Assume that $q[k]$ and $x[k]$ are bounded with $||\boldsymbol{q}||_\infty \leq q_{\max}$ and $||\boldsymbol{x}||_\infty \leq x_{\max}$. This would allow us to build a truncated function sim such that $sim(q,x) = s(\boldsymbol{q}-\boldsymbol{x}) = a(\boldsymbol{q}-\boldsymbol{x}) - a_{\min}$ when $||\boldsymbol{q}||_\infty \leq q_{\max}$ and $||\boldsymbol{x}||_\infty \leq x_{\max}$ and zero otherwise (In our case, $a = -d(q,x)$ and $a_{min} = -KT$).

This would lead us to develop an absolutely convergent (because of trunctation) Fourier transform as follows:

$$S(\boldsymbol{\omega}) = \frac{1}{(2\pi)^K} \int_{\boldsymbol{t}\in\mathbb{R}^K} s(t) e^{-i\boldsymbol{\omega}^\top \boldsymbol{t}} d\boldsymbol{t} \tag{47}$$

This allows us to compute $s(\boldsymbol{q}-\boldsymbol{x})$ using the inverse Fourier transform as:

$$s(\boldsymbol{q}-\boldsymbol{x}) = \int_{\boldsymbol{\omega}\in\mathbb{R}^K} S(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^\top(\boldsymbol{q}-\boldsymbol{x})} d\boldsymbol{\omega} = \int_{\boldsymbol{\omega}\in\mathbb{R}^K} \boldsymbol{S}_q(\boldsymbol{\omega})^\top \boldsymbol{S}_x(\boldsymbol{\omega}) d\boldsymbol{\omega} \tag{48}$$

Here,

$$\boldsymbol{S}_q(\boldsymbol{\omega}) = \Big[\text{Sign}(Re(S(\boldsymbol{\omega}))\sqrt{|Re(S(\boldsymbol{\omega}))|}\big[\cos(\boldsymbol{\omega}^\top\boldsymbol{q}), \sin(\boldsymbol{\omega}^\top\boldsymbol{q})\big], \text{Sign}(Im(S(\boldsymbol{\omega}))\sqrt{|Im(S(\boldsymbol{\omega})|}\big[-\sin(\boldsymbol{\omega}^\top\boldsymbol{q}), \cos(\boldsymbol{\omega}^\top\boldsymbol{q})\big]\Big] \tag{49}$$

and

$$\boldsymbol{S}_x(\boldsymbol{\omega}) = \Big[\sqrt{|Re(S(\boldsymbol{\omega}))|}\big[\cos(\boldsymbol{\omega}^\top\boldsymbol{q}), \sin(\boldsymbol{\omega}^\top\boldsymbol{q})\big], \sqrt{|Im(S(\boldsymbol{\omega})|}\big[\cos(\boldsymbol{\omega}^\top\boldsymbol{q}), \sin(\boldsymbol{\omega}^\top\boldsymbol{q})\big]\Big] \tag{50}$$

Note that the above expressions for $\boldsymbol{S}_q$ and $\boldsymbol{S}_x$ are similar to Eq 7 in our paper, where they were defined for each component frequency ω_k thanks to the decomposability of the score functions as a sum of independent scores across dimensions ($s(\boldsymbol{q}-\boldsymbol{x}) = \sum_{k=1}^K s(q[k]-x[k])$). In contrast, here, we show that the setup can be extended to generic (shift invariant) scoring functions which need not be decomposable as a sum across dimensions.

However, we can define a similar distribution $p(\boldsymbol{\omega})$ over the vector $\boldsymbol{\omega}$ and obtain $s(\boldsymbol{q}-\boldsymbol{x}) = \mathbb{E}_{p(\boldsymbol{\omega})}[\boldsymbol{S}_q(\boldsymbol{\omega})^\top \boldsymbol{S}_x(\boldsymbol{\omega})/p(\boldsymbol{\omega})]$

Note that the above mechanism applies for any shift invariant function with bounded query, corpus embeddings.

E.g., Facility location function used in colbert can be represented in colbert. Given a query $q = (q_1, .., q_m)$ and one corpus $x = (x_1, ..., x_n)$ colbert compute the similarity scores between these two sets as

$$sim(q,x) = \sum_{i=1}^m \max_{j\in[n]} a(q_i, x_j) \tag{51}$$

If a is a shift invariant score $a(\boldsymbol{q}_i - \boldsymbol{x}_j)$, say inverse to Euclidean distance between two items, then we can write its soft surrogate

$$sim(q,x) = \frac{1}{\lambda}\sum_{i=1}^m \log\left(\sum_{j\in[n]} \exp(\lambda a(\boldsymbol{q}_i - \boldsymbol{x}_j))\right) \tag{52}$$

Note that the function $sim(q,x)$ is a shift invariant function of the form $s(\boldsymbol{q}-\boldsymbol{x})$

> *Can this algorithm handle box embeddings (perhaps with some modifications to the algorithm / the embeddings)? Search over box embeddings is a known bottleneck and is one reason why, despite modeling advantages, they have not replaced angular-similarity embeddings in major industrial recommendation pipelines.*

Yes they can be used for box embeddings. As we discussed above, any shift invariant score can be used in our algorithm. In the following, we will show that box embedding based volume score can be expressed as a shift invariant score $a(\boldsymbol{q}-\boldsymbol{x})$. In box embedding setup, query and corpus are expressed as the boxes represented by $(\boldsymbol{z}_q, \boldsymbol{Z}_q)$ and $(\boldsymbol{z}_x, \boldsymbol{Z}_x)$ respectively. Then we represent the hard intersection between the q and x as the box $(\boldsymbol{z}, \boldsymbol{Z})$ where $\boldsymbol{z}_{q,x} = \max(\boldsymbol{z}_q, \boldsymbol{z}_x,)$ and $\boldsymbol{Z}_{q,x} = \min(\boldsymbol{Z}_q, \boldsymbol{Z}_x,)$. Then the score between q,x is measured as:

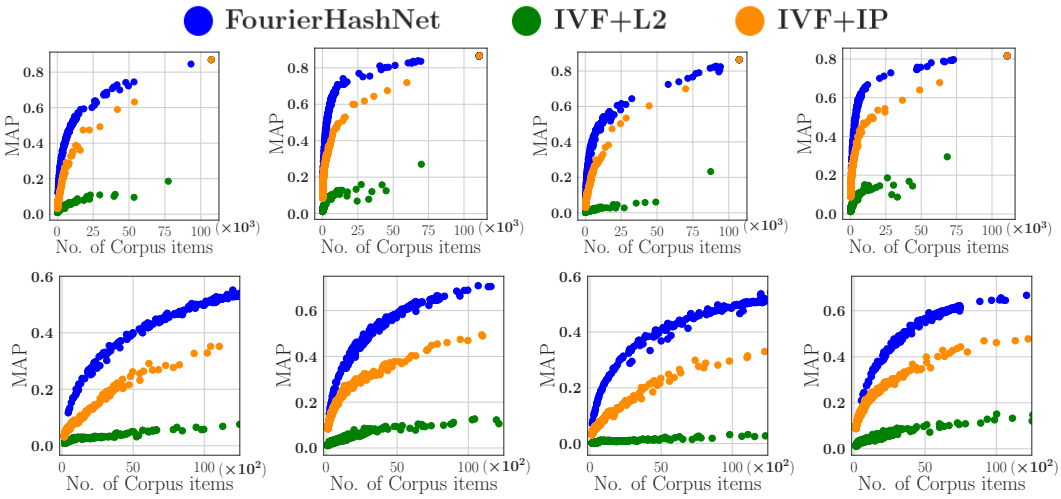
$$sim(q,x) = \prod_{k=1}^K [\boldsymbol{Z}_{q,x}[k] - \boldsymbol{z}_{q,x}[k]]_+ \qquad \text{eqn (S)} \tag{53}$$

We will show that there exists embedding $\boldsymbol{q}$ and $\boldsymbol{x}$ for which $sim(q,x) = a(\boldsymbol{q} - \boldsymbol{x})$. We first note that $\max(x,y) = x + (y-x)_+$ and $min(x,y) = y - (x-y)_+$. Using them, we have $\boldsymbol{z}_{q,x} = \boldsymbol{z}_q + (\boldsymbol{z}_x - \boldsymbol{z}_q)_+$ and $\boldsymbol{Z}_{q,x} = \boldsymbol{Z}_x - (\boldsymbol{Z}_x - \boldsymbol{Z}_q)_+$. Thus, Eq. (S) is written as

$$sim(q,x) = \prod_{k=1}^{K} [\boldsymbol{Z}_x - \boldsymbol{z}_q - (\boldsymbol{z}_x - \boldsymbol{z}_q)_+ - (\boldsymbol{Z}_x - \boldsymbol{Z}_q)_+]_+[k] \tag{54}$$

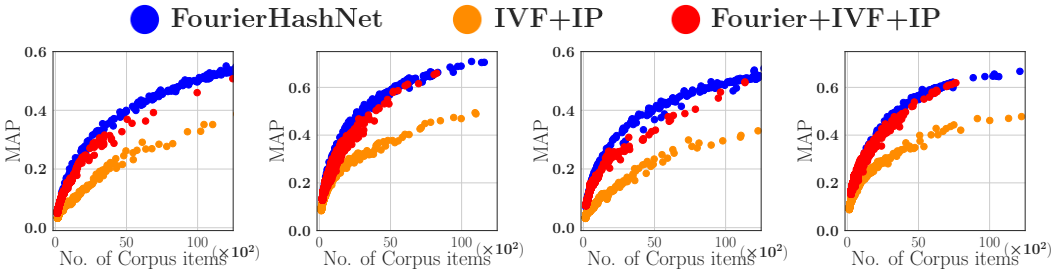
If we represent $\boldsymbol{q} = [\boldsymbol{z}_q, \boldsymbol{z}_q, \boldsymbol{Z}_q]$ and $\boldsymbol{x} = [\boldsymbol{Z}_x, \boldsymbol{z}_x, \boldsymbol{Z}_x]$, then we have $sim(q,x) = \prod_{k=1}^{K} [A_1(\boldsymbol{q} - \boldsymbol{x}) - [A_2(\boldsymbol{q} - \boldsymbol{x})]_+ - [A_3(\boldsymbol{q} - \boldsymbol{x})]_+]_+$ where $A_1 = [\mathbb{I}, 0, 0]$, $A_2 = [0, \mathbb{I}, 0]$ and $A_3 = [0, 0, \mathbb{I}]$. Chheda et al. Box Embeddings: An open-source library for representation learning using geometric structures

Thus sim(q,x) is shift invariant with respect to $\boldsymbol{q}$ and $\boldsymbol{x}$. Thus, we can extend our algorithm to box embedding setup.



(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

Figure 14: Trade-off between number of retrieved corpus items and accuracy (MAP) for MSWEB and MSNBC datasets (First row: complete view across full time axis, second row: Zoomed version of first row where there is $\geq$ 10X speedup compared to exhaustive search). We compare FOURIERHASHNET against the FAISS-IVF indexing based on L2 distance (IVF-L2) and Inner Product similarity (IVF-IP). We provide embeddings $\boldsymbol{q}, \boldsymbol{x}$ trained using hinge distance to all the methods, and use hinge distance to rank retrieved items. We observe that FOURIERHASHNET outperforms both IVF-L2 and IVF-IP across all datasets. IVF quantizers, that assign vectors to the Voronoi cells rely on a metric like L2 or IP, which is unsuitable for asymmetric hinge distance.



(a) MSWEB-1 (b) MSNBC-1 (c) MSWEB-2 (d) MSNBC-2

Figure 15: Trade-off between number of retrieved corpus items and accuracy (MAP) for MSWEB and MSNBC datasets, where there is $\geq$ 10X speedup compared to exhaustive search). We propose an alternative Fourier+IVF+IP, where we first apply our Fourier transformation on the input embeddings, before using FAISS-IVF index. We compare it against the (better performing) FAISS-IVF indexing based on Inner Product similarity (IVF-IP), and FOURIERHASHNET. We provide embeddings $\boldsymbol{q}, \boldsymbol{x}$ trained using hinge distance to all the methods, and use hinge distance to rank retrieved items. We observe that Fourier transformation provides a significant boost in performance across all datasets, as seen while comparing Fourier+IVF+IP against IVF+IP. However, we further observe that FOURIERHASHNET still outperforms Fourier+IVF+IP, most noticeably in MSWEB-2.

MAP	PTC-MM	PTC-FM	AIDS
Hinge Score	0.51	0.46	0.49
MaxSim (ColBERT) Score	0.195	0.22	0.23

Table 16: Retrieval performance comparison in terms of Mean Average Precision (MAP) for subgraph matching based graph retrieval. Each graph is encoded as a set of contextual node embeddings. The sets are scored for subgraph match using ColBERT's MaxSim score. Alternatively, the sum aggregate of the sets are scored using our Hinge Score. We observe that the Hinge score provides significantly higher MAP than MaxSim, across all three graph datasets.

References

[1] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3287–3318. World Scientific, 2018.

[2] Alex Auvolat, Sarath Chandar, Pascal Vincent, Hugo Larochelle, and Yoshua Bengio. Clustering is efficient for approximate maximum inner product search. *arXiv preprint arXiv:1507.05910*, 2015.

[3] Haim Avron, Huy Nguyen, and David Woodruff. Subspace embeddings for the polynomial kernel. *Advances in neural information processing systems*, 27, 2014.

[4] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[5] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*, 2013.

[6] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

[7] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 327–336, 1998.

[8] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–284, 2000.

[9] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

[10] Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. Box embeddings: An open-source library for representation learning using geometric structures. *arXiv preprint arXiv:2109.04997*, 2021. URL `https://www.iesl.cs.umass.edu/box-embeddings/main/index.html`.

[11] Flavio Chierichetti and Ravi Kumar. Lsh-preserving functions and their applications. *Journal of the ACM (JACM)*, 62(5):1–25, 2015.

[12] Ondrej Chum, James Philbin, Andrew Zisserman, et al. Near duplicate image detection: Min-hash and tf-idf weighting. In *Bmvc*, volume 810, pages 812–815, 2008.

[13] Shib Sankar Dasgupta, Michael Boratko, Siddhartha Mishra, Shriya Atmakuri, Dhruvesh Patel, Xiang Lorraine Li, and Andrew McCallum. Word2box: Capturing set-theoretic semantics of words using box embeddings. *arXiv preprint arXiv:2106.14361*, 2021.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL Conference*, 2019. URL `https://www.aclweb.org/anthology/N19-1423.pdf`.

[15] Khoa Doan, Shulong Tan, Weijie Zhao, and Ping Li. Asymmetric hashing for fast ranking via neural network measures. *arXiv preprint arXiv:2211.00619*, 2022.

[16] Sreenivas Gollapudi and Rina Panigrahy. Exploiting asymmetry in hierarchical topic extraction. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 475–482, 2006.

[17] E. Grant, C. Hegde, and P. Indyk. Nearly optimal linear embeddings into very low dimensions. In *Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[19] Piotr Indyk, Ilya Razenshteyn, and Tal Wagner. Practical data-dependent metric compression with provable guarantees. *Advances in Neural Information Processing Systems*, 30, 2017.

[20] Sergey Ioffe. Improved consistent sampling, weighted minhash and l1 sketching. In *2010 IEEE international conference on data mining*, pages 246–255. IEEE, 2010.

[21] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[22] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Artificial intelligence and statistics*, pages 583–591. PMLR, 2012.

[23] Vladimir Karpukhin, Barlas Oguz, 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)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://www.aclweb.org/anthology/2020.emnlp-main.550.

[24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[25] Alice Lai and Julia Hockenmaier. Learning to predict denotational probabilities for modeling entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 721–730, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-1068.

[26] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[27] Ping Li. 0-bit consistent weighted sampling. In *Proceedings of the 21th ACM SIGKDD International conference on knowledge discovery and data mining*, pages 665–674, 2015.

[28] Jimmy J. Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *WSDM Conference*, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3437963.3441667.

[29] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. Random features for kernel approximation: A survey on algorithms, theory, and beyond. *arXiv preprint arXiv:2004.11154*, 2020.

[30] Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, Jure Leskovec, et al. Neural subgraph matching. *arXiv preprint arXiv:2007.03092*, 2020.

[31] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, pages 950–961, 2007.

[32] Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, 13:1–126, 2018. URL https://www.nowpublishers.com/article/Details/INR-061.

[33] Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *International Conference on Machine Learning*, pages 1926–1934. PMLR, 2015.

[34] Anna Pagh, Rasmus Pagh, and S Srinivasa Rao. An optimal bloom filter replacement. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 823–829, 2005.

[35] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.

[36] Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–247, 2013.

[37] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[38] 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*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`.

[39] Hongyu Ren, Weihua Hu, and Jure Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. *arXiv preprint arXiv:2002.05969*, 2020.

[40] Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 2017.

[41] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen. Learning binary codes for maximum inner product search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4148–4156, 2015.

[42] Anshumali Shrivastava and Ping Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). *NeurIPS*, abs/1405.5869, 2014. URL `https://arxiv.org/pdf/1405.5869.pdf`.

[43] Anshumali Shrivastava and Ping Li. In defense of minhash over simhash. In *Artificial Intelligence and Statistics*, pages 886–894. PMLR, 2014.

[44] Konstantinos Skianis, Giannis Nikolentzos, Stratis Limnios, and Michalis Vazirgiannis. Rep the set: Neural networks for learning set representations. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1410–1420. PMLR, 26–28 Aug 2020. URL `https://proceedings.mlr.press/v108/skianis20a.html`.

[45] Zhao Song, David Woodruff, Zheng Yu, and Lichen Zhang. Fast sketching of polynomial kernels of polynomial degree. In *International Conference on Machine Learning*, pages 9812–9823. PMLR, 2021.

[46] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[47] Manolis Terrovitis, Panagiotis Bouros, Panos Vassiliadis, Timos Sellis, and Nikos Mamoulis. Efficient answering of set containment queries for skewed item distributions. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 225–236, 2011.

[48] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015. URL `https://arxiv.org/pdf/1511.06361`.

[49] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120, 2009. URL `https://arxiv.org/pdf/0902.2206.pdf`.

[50] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *Advances in neural information processing systems*, 21, 2008.

[51] Wei Wu, Bin Li, Ling Chen, and Chengqi Zhang. Canonical consistent weighted sampling for real-value weighted min-hash. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1287–1292. IEEE, 2016.

[52] Wei Wu, Bin Li, Ling Chen, and Chengqi Zhang. Consistent weighted sampling made more practical. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1035–1043, 2017.

[53] Wei Wu, Bin Li, Ling Chen, Chengqi Zhang, and S Yu Philip. Improved consistent weighted sampling revisited. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2332–2345, 2018.

[54] Wei Wu, Bin Li, Ling Chen, Junbin Gao, and Chengqi Zhang. A review for weighted minhash algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2553–2573, 2022.

[55] Xiao Yan, Jinfeng Li, Xinyan Dai, Hongzhi Chen, and James Cheng. Norm-ranging lsh for maximum inner product search. 31, 2018.

[56] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.