Quantifying and Expanding the Theoretical Capacity of Late-Interaction Retrieval Models

Julian Killingback
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
jkillingback@cs.umass.edu

Varad Ingale
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
vingale@umass.edu

Hamed Zamani
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
zamani@cs.umass.edu

Cameron Musco
University of Massachusetts Amherst
cmusco@cs.umass.edu

Abstract

Late-interaction retrieval models that use the MaxSim similarity function have shown strong empirical performance, often outperforming single-vector dense and sparse retrieval models. Despite these empirical findings, little is known about the *theoretical* representation power of MaxSim and how it compares to other retrieval approaches. This paper shows by construction that MaxSim similarity can exactly replicate the inner product between any two non-negative k-sparse vectors with possibly infinite dimension, requiring only $O(k)$ representation space. Moreover, there exist similarities that MaxSim can express while standard vector inner products with the same representation space cannot. Leveraging our theoretical framework, we introduce *Signed MaxSim* which allows late-interaction models to exactly replicate any real-valued inner product, something we prove standard MaxSim is not capable of. We also show that MaxSim can act as an aggregation of soft-OR operations and as an evaluator of logical expressions in positive Conjunctive Normal Form. Our findings show that MaxSim is at least as capable as standard vector inner products for any non-negative vectors and our extension, Signed MaxSim, is as capable for any vectors. Both similarities possess additional capabilities that inner product cannot replicate, marking one of the first theoretical justifications and quantifications of late-interaction methods. Our theoretical findings are supported empirically: on a retrieval task featuring queries with negations, Signed MaxSim improves out-of-domain performance significantly over a standard ColBERT/MaxSim baseline with nDCG@10 increasing from 0.597 to 1.000 under a vocabulary shift and from 0.008 to 0.788 on negation-only queries.

1 Introduction

The landscape of neural information retrieval is currently dominated by two primary modeling paradigms. The first, comprising both dense retrievers (e.g., DPR [13]) and learned sparse retrievers (e.g., SNRM [33]), encodes queries and documents into single, fixed-dimensional vectors. Despite their differences in sparsity, both rely on a simple inner product to estimate relevance. The second paradigm, exemplified by late-interaction models like ColBERT [14], represents texts as sets of embeddings and estimates relevance with more complex similarity measures, most often MaxSim (Chamfer Similarity)–a sum of maximum similarities between query and document embeddings.

Preprint.

Definition 1.1 (MaxSim Similarity)**.** The MaxSim similarity $S : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$, where $\mathcal{U}, \mathcal{V} \subset \mathbb{R}^n$, is defined as:

$$S(\mathcal{U}, \mathcal{V}) = \sum_{\mathbf{m} \in \mathcal{U}} \max_{\mathbf{t} \in \mathcal{V}} \langle \mathbf{m}, \mathbf{t} \rangle.$$

Here, $\langle \cdot, \cdot \rangle$ is the inner product.

Generally speaking, late-interaction models have empirically demonstrated superior performance, particularly in out-of-domain settings [28], but the underlying mechanism for this gap has not been fully elucidated. Empirical work has shown that this gap is not explained by the additional representation space leveraged by late-interaction models [14, 17], as providing more embedding dimensions to single-vector approaches has diminishing returns and does not match the performance of late-interaction models. These results suggest that it is the MaxSim similarity that is the main differentiator. In this work, we theoretically prove this hypothesis, showing that MaxSim similarity can exactly replicate the inner product between any non-negative vectors and that there are similarities that MaxSim can produce with a given representation size which an inner product cannot reproduce. These findings show that MaxSim is more capable than inner product in some cases and at least as capable as inner product when both vectors are non-negative. With that said, to show complete parity with inner-product-based retrieval methods, it must be possible to replicate inner products between any real-valued vectors, not just non-negative ones. We prove that this is impossible for standard MaxSim with a fixed embedding dimension and the number of embeddings tied to the support of the original vectors. However, this is not a fundamental limit of late-interaction retrievers, as we demonstrate with our extension to MaxSim, made possible by our theoretical framework, that enables the exact replication of the inner product between any real-valued vectors.

These findings explain why late-interaction models can outperform inner-product-based retrieval methods: they can leverage a more capable similarity function. To understand how this more powerful similarity function can be leveraged, we show how MaxSim relates to both sparse inner products and Boolean logic evaluation, two of the oldest building blocks in Information Retrieval. In detail, our three main findings are:

1. MaxSim Subsumes Inner-Product Similarity. We prove by construction that the MaxSim operation over sets of vectors in $\mathbb{R}^3$ can exactly reconstruct the inner product of any two non-negative vectors from a high-dimensional space. This result holds for both sparse and dense vectors. Specifically, the exact inner product between two k-sparse vectors can be achieved using exactly k query embeddings and $k + 1$ document embeddings in $\mathbb{R}^3$ representing each of the sparse vectors. This guarantees that late-interaction models inherently possess the full representational capacity of any standard non-negative single-vector retriever. This shows that MaxSim can replicate many existing retrieval approaches, such as learned-sparse approaches (e.g. SNRM [33] and SPLADE [6]) and traditional term matching methods (e.g. BM25 [24]). However, the non-negative requirement makes it hard to compare with retrieval approaches, like dense retrieval, that use arbitrary real-valued vectors. Additionally, supporting negative values has clear utility for several query types. Negation queries, for example, naturally benefit from being able to directly reduce the score of documents that contain the negated concept. To address this limitation, we introduce an extension to the standard late-interaction representations and MaxSim similarity that allows for the exact reconstruction of inner products between arbitrary real-valued vectors (illustrated in Figure 1). We also prove that, under mild assumptions, standard MaxSim cannot replicate this ability. Our modification is thus at least as capable as any inner-product-based or MaxSim-based retrieval method and enables better performance for certain categories of queries such as those with negations. This finding additionally relates MaxSim to sparse inner product, which have been core building blocks of information retrieval systems for decades.

2. Separating MaxSim from Single-Vector Neural Retrieval. We prove separation between the two paradigms regarding high-dimensional sparse spaces. We prove that no single finite-dimensional vector embedding can preserve the inner products of k-sparse vectors drawn from an arbitrarily high-dimensional ambient space. In contrast, MaxSim can achieve this exact preservation using sets of vectors in $\mathbb{R}^3$ with size k by our previously discussed result. This mathematically formalizes why late-interaction models are uniquely suited for handling the "long tail" of vocabulary terms and rare entities that single-vector models inevitably compress or lose.

3. Logical Expressivity. Beyond comparing to inner product-based approaches, we analyze MaxSim as a way to evaluate logical expressions. We show that it can naturally function as an aggregation

of Soft-ORs and that this allows MaxSim to act as a rank-equivalent Conjunctive Normal Form (i.e. an AND of ORs) evaluator for expressions without negations. This makes it ideal for matching synonyms and multiple forms of a relevant concept without overly rewarding the presence of multiple surface forms in a document. It also connects MaxSim to the classic Boolean search approaches used in early information retrieval systems and suggests that MaxSim may enable a neural approach which can function in a similar way to prior systems that required manual creation of the logical expressions used to query retrieval systems.

Together, our results suggest that the success of late-interaction models is not due only to the additional representation space, but also to a similarity measure that is at least as expressive as the inner product, capable of infinite-dimensional sparse representation, and natively aligned with Boolean logic.

The rest of this paper is organized into five main sections. In Section 3, we prove that MaxSim can exactly reproduce inner products between sparse non-negative vectors with possibly infinite dimension and show that our extension to MaxSim can exactly replicate the inner product of any sparse real-valued vectors. In Section 4, we prove that, unlike MaxSim, the inner product between two finite vectors cannot replicate an inner product between sparse infinite-dimensional vectors, thus demonstrating a separation between MaxSim and single-vector retrievers. These results show that late-interaction retrieval models are uniquely capable and can be viewed as a generalization of both dense and sparse single-vector retrieval models. Section 5 shows that MaxSim with larger embedding dimensions can be seen as an aggregation of weighted fuzzy OR operations and that by selecting specific values it can be viewed as evaluating positive Conjunctive Normal Form (CNF) expressions so that a ranked set of documents is ordered in an identical way to an exact Boolean evaluation. Finally, in Sections 6 and 7, we cover our experimental procedure and results comparing our MaxSim extension against standard MaxSim on queries with negations. We find that there is a significant improvement when evaluated in-domain and even greater improvement on out-of-domain data, which illustrates the additional robustness that is provided by our extension.

2 Related Work

In this section, we explore prior work related to late-interaction retrieval models and approaches to theoretically understand the capabilities of retrieval models.

2.1 Late-Interaction Retrieval Models

Late-interaction models have become a prominent class of retrieval models thanks to their strong empirical performance. The category was created with the introduction of ColBERT [14], which uses the MaxSim similarity (Chamfer similarity) to find the similarity between queries and documents. Although the empirical results are strong, the necessity to store a vector for each document token resulted in large index sizes and slow retrieval. To address this, follow-up work has investigated several methods to make retrieval more tractable and reduce the total space consumed by the embeddings. Methods to reduce representation size include using clustering and residual encoding [27, 26], employing a learned per-token pruning mechanism [23, 8], clustering document tokens to decrease redundancy [29], and using a constant number of document embeddings [20, 16]. The most relevant to our work are methods that compress the set of query and document embeddings into a single embedding such that an inner product is an approximation of MaxSim. MUVERA achieves this by using Locality Sensitive Hashing (LSH) and provides a theoretical bound on the error between the inner product of the constructed vectors and the original MaxSim similarity [5]. LEMUR uses a data-aware approach which learns a mapping from the multi-vector representations to single-vector ones in a way that minimizes the difference between the MaxSim similarity and inner product [9].

2.2 Theoretical Properties of Retrieval Models

Understanding the capabilities and limitations of retrieval models is crucial to improving retrieval effectiveness and building reliable search systems. Prior work exploring these theoretical properties has largely investigated the capacity of dense retrieval models which use a single embedding (i.e. vector) to represent queries and documents, with the similarity generally computed using an inner product. Luan et al. [19] provide a bound on the number of pair-wise errors produced by compressing a vocabulary vector into a smaller dimension. They show that, for binary vectors, the size of the

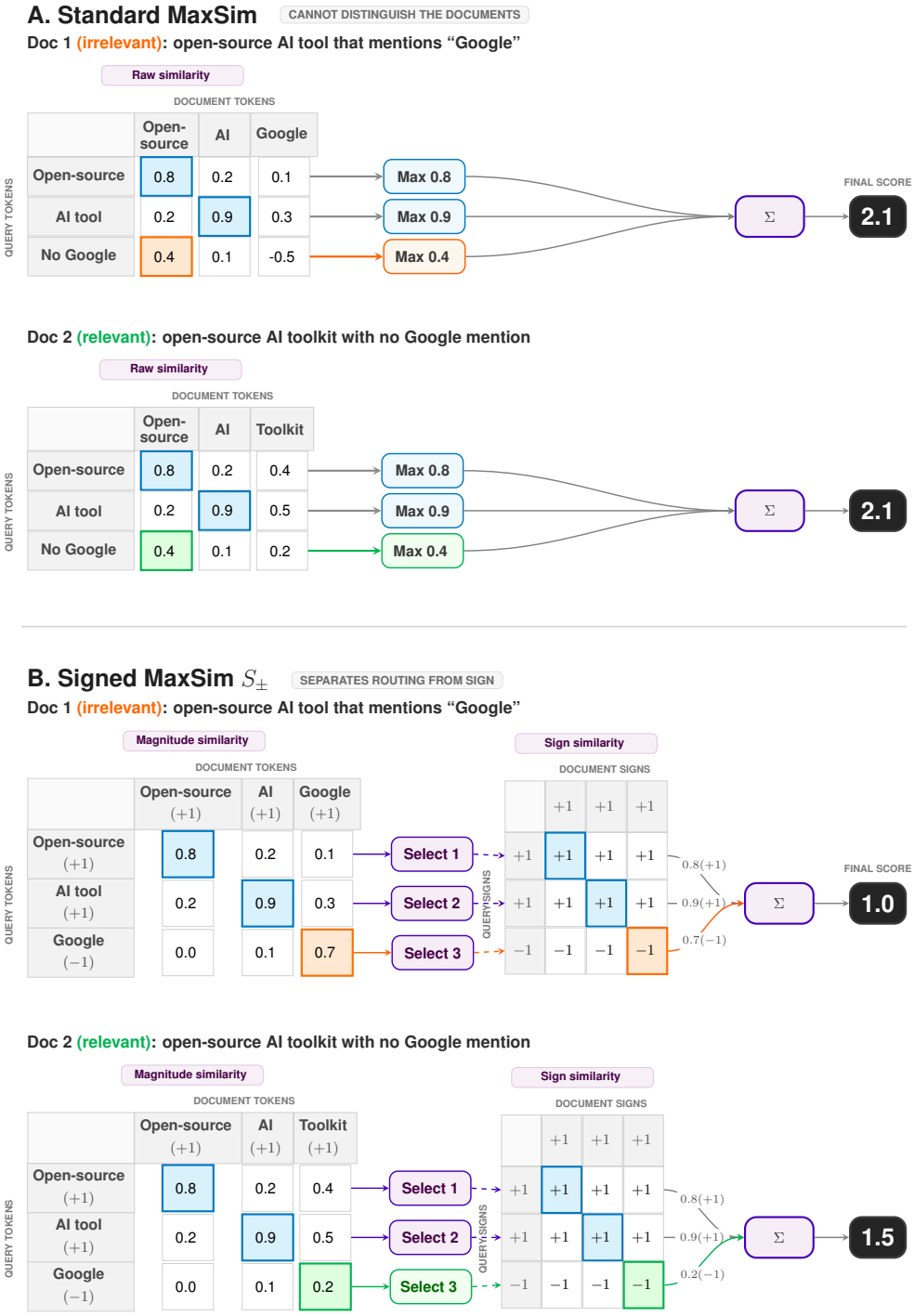

Figure 1: Comparison of scoring for the query *"Open-source AI tools that do not mention Google."* Both documents satisfy the positive request for an open-source AI tool, but only Doc 2 satisfies the exclusion constraint. **(A)** Standard MaxSim selects the largest raw similarity for each query token. Although the similarity to the explicit "Google" token can be negative, the maximum operation instead selects the incidental similarity of 0.4 to "Open-source" for both documents. The two documents therefore receive the same final score, so standard MaxSim cannot use the exclusion constraint to distinguish the relevant document. **(B)** Signed MaxSim separately uses magnitude similarity to identify the matching feature and sign similarity to determine its contribution. The explicit occurrence of "Google" therefore produces a large negative contribution, while the document with no Google mention receives only a small incidental penalty and is correctly ranked first.

embedding dimension needed to guarantee a specific error bound grows linearly with the largest number of unique terms in a document. They show that by using several vectors with smaller embedding dimension the error rate can remain the same while the embedding dimension decreases. Menon et al. [21] show that the inner product between two embeddings can approximate an arbitrary continuous scoring function given that the dimension of the embeddings is countably infinite. Ji et al. [12] show that when the output embedding dimension is smaller than the total size of the input to a transformer encoder then any approximation of a scoring function will have some irreducible error. They also show that their learned late-interaction similarity is a universal function approximation. Killingback et al. [15] shows that for any $d + 1$ document vectors with dimension d there is no query vector that can perfectly separate some relevance pattern using inner product. Weller et al. [31] proved that the number of orderings that a set of document and query embeddings can express such that there is a certain margin between relevant and irrelevant documents is dependent on the embedding dimension and number of documents. S et al. [25] observe that the results by Alon et al. [1] can be used to show that $2k + 1$ dimensions are necessary for any top-k ranking regardless of the number of documents when using single vectors and inner products to assess similarity. While the theoretical limits of single-vector dense retrieval have been increasingly scrutinized, the expressivity of multi-vector models remains largely unexplored.

Concurrent work by Jayaram [11] studies whether MaxSim similarities produced by sets of at most m unit vectors in $\mathbb{R}^d$ can be approximated by single-vector inner products in $\mathbb{R}^D$. They construct a finite family for which achieving an error ϵ requires $D = (\epsilon^2 m)^{\Omega(1/\epsilon)}$, even when the single-vector representations may be chosen after observing the complete dataset and each query contains only one vector. This establishes a strong separation in one direction: arbitrary multi-vector MaxSim similarities cannot generally be compressed into comparably sized single-vector representations. This finding aligns with our own finding that single-vector finite-dimensional inner products cannot simulate arbitrary-dimensional sparse inner products. Since we show that such sparse inner products can be simulated by MaxSim, this implies that single-vector embeddings cannot simulate multi-vector embeddings of comparable dimension – giving an analogous result to Jayaram [11], but in the exact rather than approximate setting. Unlike our work, Jayaram [11] does not address the reverse question: which single-vector similarities standard MaxSim can or cannot realize. They also do not propose extensions of MaxSim or demonstrate empirical retrieval gains resulting from their theoretical analysis.

3 Late-Interaction Similarity for Exact Inner Product Computations

This section contains the proof that MaxSim similarity can exactly replicate the inner product between non-negative vectors of arbitrary dimension (possibly infinite) with representation space tied to the number of non-zero elements. We additionally show that MaxSim can be extended to enable the exact inner product replication of any real-valued vectors.

3.1 Exact Non-Negative Inner Product Computations with MaxSim Similarity

As our results depend on the support and sparsity of vectors, we formally define these terms.

Definition 3.1 (Support and Sparsity)**.** The **support** of a vector $\mathbf{x}$ is the set of indices corresponding to its non-zero elements, defined as $\text{supp}(\mathbf{x}) := \{i \in \mathbb{N} \mid x_i \neq 0\}$. A vector $\mathbf{x}$ is said to be k**-sparse** if $|\text{supp}(\mathbf{x})| \leq k$, i.e., the cardinality of its support is at most k.

We now state the main result of this section: MaxSim similarity can exactly replicate the inner product between two non-negative vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n_{\geq 0}$, where n can be countably infinite, and the size of the vector sets $\mathcal{U}, \mathcal{V} \subset \mathbb{R}^3$ representing the original vectors depends only on the number of non-zero elements in $\mathbf{u}$ and $\mathbf{v}$. Importantly, the construction of $\mathcal{U}$ and $\mathcal{V}$ can be done independently (i.e. $\mathcal{U}$ has no knowledge of $\mathcal{V}$ and vice versa), making the finding directly applicable to realistic settings such as document retrieval.

This result indicates that MaxSim is as capable as an inner product between non-negative vectors and additionally can represent an infinite-dimension non-negative sparse vector with representation space only tied to the non-zero elements. We show in Section 4 that this is impossible for a standard vector inner product to accomplish, making MaxSim more capable in this regard.

Theorem 3.1. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n_{\geq 0}$ be non-negative vectors, where $\mathbf{u}$ is k_u-sparse and $\mathbf{v}$ is k_v-sparse. There exist sets of vectors $\mathcal{U}, \mathcal{V} \subset \mathbb{R}^3$, with $|\mathcal{U}| = k_u$ and $|\mathcal{V}| = k_v + 1$, such that:

$$\langle \mathbf{u}, \mathbf{v} \rangle = S(\mathcal{U}, \mathcal{V}), \tag{1}$$

Furthermore, the sets $\mathcal{U}$ and $\mathcal{V}$ can be constructed independently; the mapping of $\mathbf{u}$ to $\mathcal{U}$ depends strictly on $\mathbf{u}$, and the mapping of $\mathbf{v}$ to $\mathcal{V}$ depends strictly on $\mathbf{v}$, preserving the asymmetric dual-encoder architecture standard in retrieval.

We transform the sparse vectors $\mathbf{u}$ and $\mathbf{v}$ into sets of dense 3-dimensional vectors, denoted $\mathcal{U}$ and $\mathcal{V}$. The key intuition is that we can use the inner product between two vectors in $\mathbb{R}^3$ to represent evaluating a specific quadratic polynomial at a given point. By creating the polynomial so that when evaluated at a non-matching index the value is less than 0 and at a matching index is a predefined value, the maximum in the similarity only allows matching indices to contribute to the final similarity.

Lemma 3.1 (Polynomial Construction)**.** For any $d \in \mathbb{N}$ and $w \in \mathbb{R}_{\geq 0}$, there exists a coefficient vector $c(d, w) = (c_0, c_1, c_2)^T \in \mathbb{R}^3$ defining a quadratic polynomial $p(x) = \mathbf{c}^T \phi(x) = c_0 + c_1 x + c_2 x^2$ such that:

1. $p(d) = w$
2. $p(d') < 0$ for all $d' \in \mathbb{N}$ where $d' \neq d$.

Proof. Consider the polynomial $p(x) = w - C(x - d)^2$ for some constant $C > 0$. By construction, $p(d) = w - C(d - d)^2 = w$. For any other index $d' \in \mathbb{N}$ where $d' \neq d$, the term $(d' - d)^2 \geq 1$. Thus, $p(d') = w - C(d' - d)^2 \leq w - C$. To ensure $p(d') < 0$, we must choose C such that $w - C < 0$, which implies $C > w$. Any such choice of C satisfies the conditions. For example, let $C = w + 1$. The polynomial can be expanded to find the coefficients:

$$\begin{aligned} p(x) &= w - C(x^2 - 2dx + d^2) \\ &= (-C)x^2 + (2Cd)x + (w - Cd^2). \end{aligned}$$

The corresponding coefficient vector is $c(d, w) = (w - Cd^2, 2Cd, -C)^T$. □

The polynomial construction creates a function that evaluates to a given value at a certain point and negative elsewhere. We now construct the complementary embedding that represents a specific point to evaluate. Previous work has used a similar approach to exactly factorize sparse matrices [3, 4].

Definition 3.2 (Embedding Map)**.** Define the quadratic embedding map $\phi : \mathbb{N} \to \mathbb{R}^3$ as:

$$\phi(d) = \begin{pmatrix} 1 \\ d \\ d^2 \end{pmatrix}. \tag{2}$$

We now prove the main theorem of this section.

Proof of Theorem 3.1. We construct the sets $\mathcal{U}$ and $\mathcal{V}$ as follows. For the vector $\mathbf{u}$, we map each non-zero entry u_i to a vector in $\mathbb{R}^3$ using the embedding map ϕ:

$$\mathcal{U} = \{u_i \phi(i) \mid i \in \text{supp}(\mathbf{u})\}. \tag{3}$$

For the vector $\mathbf{v}$, we utilize the polynomial coefficients from Lemma 3.1. We define:

$$\mathcal{V} = \{c(i, v_i) \mid i \in \text{supp}(\mathbf{v})\} \cup \{\mathbf{0}\}, \tag{4}$$

where $\mathbf{0} \in \mathbb{R}^3$ is the zero vector.

We now show that $S(\mathcal{U}, \mathcal{V}) = \langle \mathbf{u}, \mathbf{v} \rangle$. By definition of the standard inner product, $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i \in \text{supp}(\mathbf{u})} u_i v_i$. Expanding the Max-Sim similarity for our constructed sets, we have:

$$S(\mathcal{U}, \mathcal{V}) = \sum_{\mathbf{m} \in \mathcal{U}} \max_{\mathbf{t} \in \mathcal{V}} \langle \mathbf{m}, \mathbf{t} \rangle = \sum_{i \in \text{supp}(\mathbf{u})} \max_{\mathbf{t} \in \mathcal{V}} \langle u_i \phi(i), \mathbf{t} \rangle. \tag{5}$$

To prove equivalence, it suffices to show that for every $i \in \text{supp}(\mathbf{u})$, the inner term evaluates to $u_i v_i$. We analyze two cases based on the value of v_i:

Case 1: $v_i = 0$ **(i.e.,** $i \notin$ **supp**$(\mathbf{v})$**).**
By Lemma 3.1, for any $c(j, v_j) \in \mathcal{V}$ (where $j \neq i$), the corresponding polynomial evaluates to a strictly negative value at i. Thus, $\langle u_i\phi(i), c(j, v_j)\rangle = u_i p(i) < 0$ (since $u_i > 0$). However, the zero vector $\mathbf{0} \in \mathcal{V}$ yields $\langle u_i\phi(i), \mathbf{0}\rangle = 0$. Therefore, the maximum over $\mathcal{V}$ is achieved by $\mathbf{0}$, yielding 0. This matches $u_i v_i = 0$.

Case 2: $v_i > 0$ **(i.e.,** $i \in$ **supp**$(\mathbf{v})$**).**
By construction, $\mathcal{V}$ contains the coefficient vector $c(i, v_i)$. By Lemma 3.1, the inner product with this specific vector is $\langle u_i\phi(i), c(i, v_i)\rangle = u_i p(i) = u_i v_i$. For all other vectors $c(j, v_j) \in \mathcal{V}$ $(j \neq i)$, Lemma 3.1 guarantees $p(i) < 0$, yielding a negative inner product. The zero vector yields 0. Since $u_i, v_i > 0$, the term $u_i v_i$ is strictly positive and is therefore the maximum over all $\mathbf{t} \in \mathcal{V}$.

In both cases, $\max_{\mathbf{t}\in\mathcal{V}}\langle u_i\phi(i), \mathbf{t}\rangle = u_i v_i$. Summing over all $i \in \text{supp}(\mathbf{u})$ yields the exact inner product, concluding the proof. □

3.2 Standard MaxSim Cannot Exactly Replicate Signed Inner Products

We now ask whether the restriction that both vectors must be non-negative is necessary or whether a similarly sparsity-preserving representation can recover arbitrary signed inner products without modifying the MaxSim operation.

We allow each query or document embedding complete information on the input vector and allow the document representation to include an additional set of embeddings that are shared for all document representations similar to the $\mathbf{0}$ vector in Section 3.1. The only limitation is that the number of query and document embeddings (not including those shared for all documents) must be equal to the cardinalities of the supports of the original query and document vectors. Even with these minimal constraints and a high degree of flexibility, standard MaxSim cannot recover all signed inner products in a fixed embedding dimension.

Definition 3.3 (Contextual Sparsity-Preserving Encoding)**.** Let $M \in \mathbb{N}$ be fixed. A contextual sparsity-preserving encoding consists of independently constructed mappings

$$\mathcal{U} : \mathbb{R}^n \to 2^{\mathbb{R}^M}, \qquad \mathcal{V}_{\text{var}} : \mathbb{R}^n \to 2^{\mathbb{R}^M},$$

together with a fixed finite set $\mathcal{F} \subset \mathbb{R}^M$ shared by every document representation. Here, $2^{\mathbb{R}^M}$ represents all possible subsets with elements from $\mathbb{R}^M$. The mappings satisfy

$$|\mathcal{U}(\mathbf{u})| = |\text{supp}(\mathbf{u})|, \qquad |\mathcal{V}_{\text{var}}(\mathbf{v})| = |\text{supp}(\mathbf{v})|,$$

and the complete document representation is

$$\mathcal{V}(\mathbf{v}) = \mathcal{V}_{\text{var}}(\mathbf{v}) \cup \mathcal{F}.$$

Each embedding in $\mathcal{U}(\mathbf{u})$ and $\mathcal{V}_{\text{var}}(\mathbf{v})$ may depend arbitrarily on the complete corresponding input vector. The only restriction is that the encoding contains exactly one input-dependent embedding per nonzero coordinate, while the vectors in $\mathcal{F}$ are fixed and identical across documents. This definition fits the prior construction for non-negative vectors when $\mathcal{F} = \{\mathbf{0}\}$. Additionally, in the non-negative construction the embeddings are constructed with only coordinate-wise information.

Theorem 3.2 (Sparsity-Preserving Limitation of Standard MaxSim)**.** Let $M, n \in \mathbb{N}$, let $\mathcal{F}, \mathcal{U}$, and $\mathcal{V}_{\text{var}}$ satisfy Definition 3.3. If

$$S\big(\mathcal{U}(\mathbf{u}), \mathcal{V}(\mathbf{v})\big) = \langle \mathbf{u}, \mathbf{v}\rangle$$

for every pair of one-sparse vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, then $n \leq M$. Consequently, when $n > M$, some pair of one-sparse vectors cannot have its inner product exactly recovered, regardless of the encoders or shared vectors used.

Proof. For each $i, j \in [n]$, let

$$\mathbf{u}_i = -\mathbf{e}_i, \qquad \mathbf{v}_j = \mathbf{e}_j, \qquad \langle \mathbf{u}_i, \mathbf{v}_j\rangle = -\delta_{ij},$$

where $\mathbf{e}_i$ denotes the ith standard basis vector and δ_{ij} is the Kronecker delta. Thus, query $\mathbf{u}_i$ must score document $\mathbf{v}_i$ as -1 and every document $\mathbf{v}_j$ with $j \neq i$ as 0.

Because these vectors are one-sparse, their input-dependent representations contain one embedding each. Write

$$\mathcal{U}(\mathbf{u}_i) = \{\mathbf{m}_i\}, \qquad \mathcal{V}_{\text{var}}(\mathbf{v}_j) = \{\mathbf{t}_j\},$$

for $\mathbf{m}_i, \mathbf{t}_j \in \mathbb{R}^M$. For each query, define the document-independent contribution of the shared vectors by

$$b_i = \max_{\mathbf{f} \in \mathcal{F}} \langle \mathbf{m}_i, \mathbf{f} \rangle,$$

using the convention $b_i = -\infty$ when $\mathcal{F} = \emptyset$. Exact recovery therefore requires

$$\max\{\langle \mathbf{m}_i, \mathbf{t}_j \rangle, b_i\} = -\delta_{ij} \qquad \text{for all } i, j \in [n].$$

Taking $j = i$, the maximum must equal -1, so both of its arguments are at most -1. In particular,

$$b_i \leq -1 \qquad \text{and} \qquad a_i := \langle \mathbf{m}_i, \mathbf{t}_i \rangle \leq -1.$$

For $j \neq i$, the required score is 0. Since $b_i \leq -1$, the shared vectors cannot produce this score, and exact recovery forces

$$\langle \mathbf{m}_i, \mathbf{t}_j \rangle = 0.$$

Hence the matrix $A \in \mathbb{R}^{n \times n}$ defined by

$$A_{ij} = \langle \mathbf{m}_i, \mathbf{t}_j \rangle$$

is diagonal with nonzero diagonal entries a_i, and therefore $\operatorname{rank}(A) = n$. On the other hand, if $U, V \in \mathbb{R}^{n \times M}$ have rows $\mathbf{m}_i^T$ and $\mathbf{t}_i^T$, respectively, then $A = UV^T$, implying

$$\operatorname{rank}(A) \leq M.$$

Thus $n \leq M$, proving the result. □

The proof shows that MaxSim is unable to exactly replicate even a simple inner product reconstruction between real-valued vectors. The shared document vectors $\mathcal{F}$ are of limited usefulness, since for a fixed query $\mathbf{u}_i$, their entire contribution is limited to the same value b_i for every document, forcing the input-dependent document embeddings $\mathbf{t}_j$ to realize a full-rank diagonal similarity matrix.

The zero vector used in Section 3.1 is an especially direct example. If $\mathbf{0} \in \mathcal{F}$, then every query has similarity 0 with a shared document vector, making every MaxSim score non-negative. Standard MaxSim then cannot recover even the one-dimensional inner product $\langle -\mathbf{e}_i, \mathbf{e}_i \rangle = -1$.

Theorem 3.2 does not mean that standard MaxSim can never produce negative scores. Rather, it shows that standard MaxSim cannot extend the earlier sparsity-preserving construction to arbitrary real-valued vectors while retaining a fixed embedding dimension. This motivates our extension, detailed in the following section, which overcomes this limitation with standard MaxSim by decomposing the original vector values into two parts, magnitude and sign, which remain disentangled until after the maximum step in the similarity.

3.3 Extension to MaxSim Similarity for Exact Real Number Inner Product Computations

In Section 3.1, we showed that MaxSim similarity can exactly replicate the inner product between two non-negative vectors with representation size corresponding to the number of non-zero indices. In Section 3.2, we proved that with a sparsity-preserving encoder standard MaxSim cannot replicate exact inner products between real-valued vectors. In this section, we address this limitation by extending MaxSim and multi-vector representations to enable the exact inner product to be replicated for any real-valued vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, where, as before, n can be countably infinite. The key modification is to handle negative values by decoupling each entry into its magnitude and sign. The embedding method used in Section 3.1 is applied strictly to the magnitudes, so the maximum in the similarity still properly selects matching indices, while the scalar signs are incorporated after the maximum to reconstruct the correct final product.

3.3.1 Modified Vector Representation and Similarity

Any non-zero real number x can be decomposed as $x = \operatorname{sgn}(x) \cdot |x|$. We adapt the sets $\mathcal{U}$ and $\mathcal{V}$ from Theorem 3.1 to store these signs alongside the embedded vectors. We start by modifying MaxSim to allow the incorporation of sign values after the maximum operation.

Definition 3.4 (Signed MaxSim Similarity)**.** Let $\mathcal{U}_s$ and $\mathcal{V}_s$ be sets of pairs, where each pair consists of a vector and a scalar sign. For each query pair $(\mathbf{m}, s_q) \in \mathcal{U}_s$, let $(\mathbf{t}^*(\mathbf{m}), s_d^*(\mathbf{m}))$ denote the specific pair in $\mathcal{V}_s$ that maximizes the inner product with $\mathbf{m}$:

$$(\mathbf{t}^*(\mathbf{m}), s_d^*(\mathbf{m})) = \underset{(\mathbf{t}, s_d) \in \mathcal{V}_s}{\arg\max} \langle \mathbf{m}, \mathbf{t} \rangle. \tag{6}$$

The Signed MaxSim similarity $S_\pm$ is defined as the sum over all query pairs of their maximum inner product, multiplied by the signs of both the query vector and the maximizing document vector:

$$S_\pm(\mathcal{U}_s, \mathcal{V}_s) = \sum_{(\mathbf{m}, s_q) \in \mathcal{U}_s} s_q \cdot s_d^*(\mathbf{m}) \cdot \langle \mathbf{m}, \mathbf{t}^*(\mathbf{m}) \rangle. \tag{7}$$

To utilize this new similarity function, we must also adapt our vector representations to store these scalar signs alongside the embedded magnitudes.

Definition 3.5 (Signed Vector Sets)**.** Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ be real-valued vectors. We represent them as sets of pairs, $\mathcal{U}_s, \mathcal{V}_s \subset \mathbb{R}^3 \times \{-1, 1\}$, where each pair consists of a vector and a scalar sign.

For the vector $\mathbf{u}$, we map each non-zero entry u_i using its magnitude $|u_i|$ and sign $\operatorname{sgn}(u_i)$:

$$\mathcal{U}_s = \left\{ \big(|u_i|\phi(i), \operatorname{sgn}(u_i)\big) \mid i \in \text{supp}(\mathbf{u}) \right\}. \tag{8}$$

For the vector $\mathbf{v}$, let $c(i, |v_i|)$ be the polynomial coefficient vector generated by Lemma 3.1. We define:

$$\mathcal{V}_s = \left\{ \big(c(i, |v_i|), \operatorname{sgn}(v_i)\big) \mid i \in \text{supp}(\mathbf{v}) \right\} \cup \{(\mathbf{0}, 1)\}, \tag{9}$$

where $(\mathbf{0}, 1)$ is the zero pair, included to serve as a baseline for non-matching indices.

3.3.2 Inner Product Recovery

With the signed vector representations and the modified similarity function defined, we now prove that this formulation exactly recovers the inner product for any arbitrary real-valued vectors.

Theorem 3.3. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ be arbitrary real-valued vectors, where $\mathbf{u}$ is k_u-sparse and $\mathbf{v}$ is k_v-sparse. There exist sets of pairs $\mathcal{U}_s, \mathcal{V}_s \subset \mathbb{R}^3 \times \{-1, 1\}$, with $|\mathcal{U}_s| = k_u$ and $|\mathcal{V}_s| = k_v + 1$, such that:

$$\langle \mathbf{u}, \mathbf{v} \rangle = S_\pm(\mathcal{U}_s, \mathcal{V}_s). \tag{10}$$

Importantly, like in Theorem 3.1, $\mathcal{U}_s$ and $\mathcal{V}_s$ can be produced with only access to $\mathbf{u}$ and $\mathbf{v}$, respectively, with no knowledge of the other vector until the similarity is calculated.

We extend the construction from Theorem 3.1 by operating exclusively on the absolute values of the vector entries. Because magnitudes are strictly positive, the inner maximization step correctly identifies matching indices exactly as it did in the positive-only case. Once the matching index is found (or the zero vector is selected in the case of a mismatch), the original signs are multiplied back into the resulting magnitude to recover the exact real-valued product.

Proof. We construct the sets $\mathcal{U}_s$ and $\mathcal{V}_s$ as defined above. We analyze the term contributed to the Signed Max-Sim sum by each pair $(\mathbf{m}_i, s_i) \in \mathcal{U}_s$, where $\mathbf{m}_i = |u_i|\phi(i)$ and $s_i = \operatorname{sgn}(u_i)$ for some $i \in \text{supp}(\mathbf{u})$.

The maximization step, $\arg\max_{(\mathbf{t}, s_d) \in \mathcal{V}_s} \langle \mathbf{m}_i, \mathbf{t} \rangle$, depends only on the vector components and operates on strictly positive magnitudes ($u_i > 0$ and $v_i > 0$). Therefore, the logic follows the proof of Theorem 3.1. We analyze two cases based on whether the index i is present in $\mathbf{v}$:

Case 1: $i \notin$ supp$(\mathbf{v})$ (Mismatched Index).
By Lemma 3.1, for all $(c(j, |v_j|), \operatorname{sgn}(v_j)) \in \mathcal{V}_s$, the inner product $\langle |u_i|\phi(i), c(j, |v_j|) \rangle = |u_i|p(i) < 0$. However, the zero pair yields $\langle |u_i|\phi(i), \mathbf{0} \rangle = 0$. Thus, the unique maximum is 0, achieved by the pair $(\mathbf{t}^*(\mathbf{m}_i), s_d^*(\mathbf{m}_i)) = (\mathbf{0}, 1)$. The contribution to the sum is:

$$s_i \cdot s_d^*(\mathbf{m}_i) \cdot \langle \mathbf{m}_i, \mathbf{t}^*(\mathbf{m}_i) \rangle = \operatorname{sgn}(u_i) \cdot 1 \cdot 0 = 0. \tag{11}$$

This correctly matches the product $u_i v_i = 0$.

Case 2: $i \in \textbf{supp}(\mathbf{v})$ **(Matching Index).**
By construction, $\mathcal{V}_s$ contains the pair $(c(i, |v_i|), \operatorname{sgn}(v_i))$. By Lemma 3.1, the inner product with this specific vector is $\langle |u_i|\phi(i), c(i, |v_i|)\rangle = |u_i||v_i| > 0$. For all other vectors $c(j, |v_j|) \in \mathcal{V}_s$ ($j \neq i$), the inner product is strictly negative, and the zero vector yields 0. Thus, the unique maximum is $|u_i||v_i|$, achieved by the pair $(\mathbf{t}^*(\mathbf{m}_i), s_d^*(\mathbf{m}_i)) = (c(i, |v_i|), \operatorname{sgn}(v_i))$. The contribution to the sum is:

$$\begin{aligned} s_i \cdot s_d^*(\mathbf{m}_i) \cdot \langle \mathbf{m}_i, \mathbf{t}^*(\mathbf{m}_i)\rangle &= \operatorname{sgn}(u_i) \cdot \operatorname{sgn}(v_i) \cdot \big(|u_i||v_i|\big) \\ &= \big(\operatorname{sgn}(u_i)|u_i|\big) \cdot \big(\operatorname{sgn}(v_i)|v_i|\big) \\ &= u_i v_i. \end{aligned}$$

In both cases, the $\arg\max$ is uniquely defined, and the term contributed by the index $i \in \operatorname{supp}(\mathbf{u})$ is exactly $u_i v_i$. Summing over all pairs in $\mathcal{U}_s$ yields:

$$S_{\pm}(\mathcal{U}_s, \mathcal{V}_s) = \sum_{i \in \operatorname{supp}(\mathbf{u})} u_i v_i = \langle \mathbf{u}, \mathbf{v}\rangle. \tag{12}$$

This completes the proof. □

4 The Dimensionality Bottleneck of Standard Inner Products

In Theorem 3.1, we demonstrated that MaxSim similarity can exactly reproduce the inner product of two non-negative k-sparse vectors of arbitrary dimension using finite sets of low-dimensional vectors. Given this capability, it is natural to ask whether single-vector representations can achieve a similar compression. In this section, we demonstrate a fundamental rank constraint: no finite-dimensional inner product space can exactly preserve the inner products of arbitrarily high-dimensional sparse vectors, establishing a strict theoretical separation between single-vector and multi-vector retrieval paradigms.

To prove that embedding high-dimensional sparse vectors into a lower finite dimension d is impossible, it suffices to show that we cannot even embed $d+1$ mutually orthogonal vectors into $\mathbb{R}^d$. We establish this lower bound below. To ensure our impossibility result is as general as possible, we allow for asymmetric embeddings, where the query and document vectors can be processed by entirely different mapping functions.

Theorem 4.1 (Impossibility of Dimensionality Compression)**.** Let $\mathcal{S} = \{\mathbf{e}_1, \ldots, \mathbf{e}_{d+1}\} \subset \mathbb{R}^{d+1}$ be the set of $d+1$ standard basis vectors. For any finite dimension $d \in \mathbb{N}$, there are no mappings $f, g : \mathcal{S} \to \mathbb{R}^d$ that exactly preserve pairwise inner products. That is, there are no f and g such that for all $\mathbf{x}, \mathbf{y} \in \mathcal{S}$:

$$\langle f(\mathbf{x}), g(\mathbf{y})\rangle = \langle \mathbf{x}, \mathbf{y}\rangle. \tag{13}$$

Proof. Assume such mappings f and g exist. Let $U, V \in \mathbb{R}^{d \times (d+1)}$ be the matrices whose i-th columns are the mapped vectors $f(\mathbf{e}_i)$ and $g(\mathbf{e}_i)$, respectively. The matrix of pairwise inner products between the mapped vectors is given by the product $U^T V \in \mathbb{R}^{(d+1)\times(d+1)}$.

By our assumption, $U^T V$ must exactly equal the Gram matrix of the original vectors in $\mathcal{S}$, which is the identity matrix I_{d+1}. However, the rank of a matrix product is bounded by its inner dimension, meaning $\operatorname{rank}(U^T V) \leq d$. This implies $d + 1 = \operatorname{rank}(I_{d+1}) \leq d$, which is a contradiction. □

This rank bottleneck dictates that standard inner products cannot exactly compress even $d + 1$ orthogonal concepts into d dimensions. Consequently, representing vectors from an arbitrarily large vocabulary in a fixed-dimensional space inherently requires approximation.

This highlights a core advantage of the MaxSim architecture. While a standard inner product is strictly constrained by the linear algebraic rank of its embedding dimension, MaxSim represents a k-sparse vector as a set of k vectors in $\mathbb{R}^3$. By requiring only $O(k)$ parameters, MaxSim bypasses this rank constraint entirely, perfectly preserving exact inner products regardless of the underlying vocabulary size.

5 Late-Interaction Similarity as Logical Expression Evaluation

In this section, we explore the logical expressivity of MaxSim. We demonstrate how it naturally acts as an aggregation of fuzzy logical operations and prove its capability to evaluate positive Conjunctive Normal Form expressions (i.e. Conjunctive Normal Form expressions without negations).

5.1 Motivation: MaxSim as a Fuzzy Logical OR

In the previous section, we demonstrated that MaxSim can replicate the exact inner product of two non-negative vectors by mapping each non-zero entry to a 3-dimensional embedding. While this establishes MaxSim's capacity to represent standard vector spaces, it does not fully explain the benefit of MaxSim.

Standard inner products compute a sum over all matching features. In information retrieval, this inherently acts as a "fuzzy AND" or an accumulation of evidence: a document with both "red" and "blue" features will score higher than a document with only "red", even if the user only wanted one or the other.

MaxSim, however, computes a sum of *maximums*. If we can encode a set of alternative concepts (e.g., "red", "crimson", "scarlet") into a *single* query vector, the inner maximization of MaxSim will score the document based solely on its single best matching feature. It will not over-reward documents that contain multiple synonyms. Similarly, an OR relationship between concepts like "red" and "blue" can be encoded into a single query vector, ensuring that documents containing both terms are not disproportionately rewarded when the user only desires one. In this way, MaxSim naturally executes a **fuzzy logical OR** over grouped query terms.

In this section, we formalize this intuition. We show that by increasing the embedding dimension proportionally to the size of the clause, a single query vector can encode an arbitrarily large OR-clause of terms. We then prove that this mechanism allows standard MaxSim to evaluate positive Conjunctive Normal Form (CNF) Boolean logic so that the ordering of documents matches the exact evaluation of the expression.

5.2 Formalism for Grouped Queries (Fuzzy OR)

To formalize the intuition of MaxSim as a fuzzy OR, we first define the mathematical structures representing documents and grouped queries. Let our universe of keys be the natural numbers $\mathbb{N}$ and our values be non-negative real numbers in $\mathbb{R}_{\geq 0}$.

Definition 5.1 (Document and Query Groups)**.**
Let $D = \{(d_1, w_1), \ldots, (d_{k_d}, w_{k_d})\}$ be a **document set** of key-value pairs, where $d_i \in \mathbb{N}$ are unique keys and $w_i \in \mathbb{R}_{\geq 0}$ are weights.

Let $S = \{(d_{q,1}, w_{q,1}), \ldots, (d_{q,m}, w_{q,m})\}$ be a **query group**. The group S is a set of key-value pairs representing a single OR-clause of terms, where $d_{q,i} \in \mathbb{N}$ are unique keys and $w_{q,i} \in \mathbb{R}_{\geq 0}$ are weights. Let $D_S \subset \mathbb{N}$ denote the set of keys present in query group S.

In standard fuzzy logic, the logical OR of a set of truth values $x_1, \ldots, x_n \in [0, 1]$ is defined by the maximum operator: $\bigvee_{i=1}^{n} x_i = \max(x_1, \ldots, x_n)$. To apply this to information retrieval, we must extend this concept in two natural ways.

First, rather than strict $[0, 1]$ truth values, a document's match to a specific term is typically represented by an unbounded, non-negative relevance weight $w_i \in \mathbb{R}_{\geq 0}$. It is natural that a document might be highly relevant to one term but only marginally relevant to another.

Second, not all terms within an OR-clause are equally valuable to the user. For example, an exact keyword match might be highly desired, while a broader synonym is acceptable but less preferred. We can represent this by assigning a query weight $w_{q,i} \in \mathbb{R}_{\geq 0}$ to each term $d_{q,i}$ in the query group S.

When a document contains a matching term, its overall score for that specific term is naturally the product of the document's relevance weight and the query's importance weight: $w_i w_{q,i}$. To evaluate the entire OR-clause, we apply the standard fuzzy logic principle of taking the maximum over all available options. This yields a weighted, unbounded generalization of the logical OR, which we refer to as a **Weighted Max-OR**.

Definition 5.2 (Weighted Max-OR)**.** Given a document set D and a query group S representing an OR-clause of terms, we define the **Weighted Max-OR** evaluation of S on D as the maximum combined weight across all matching terms:

$$\text{MaxOR}(S, D) = \max\left(\{0\} \cup \{w_i w_{q,i} \mid (d_i, w_i) \in D \text{ and } (d_i, w_{q,i}) \in S\}\right). \tag{14}$$

The set $\{0\}$ is included so that when there are no shared keys between D and S the maximum is still defined and evaluates to 0 like in standard fuzzy logic.

This definition provides a natural extension of Boolean logic to continuous retrieval scores. By taking the maximum over the matching terms in a query group, we select the single strongest weighted match for that concept. This perfectly captures the intent of an OR-clause: a document is rewarded for its best-matching synonym, but its score is not artificially inflated if it happens to contain multiple synonyms (which would happen if we used a sum, representing an accumulation of matches or a fuzzy AND).

With the Weighted Max-OR defined for a single clause, we now show that MaxSim can evaluate an entire set of these clauses simultaneously.

Theorem 5.1 (Weighted Max-OR Evaluation)**.** Let D be a document set and $Q = \{S_1, \ldots, S_{k_q}\}$ be a grouped query set consisting of multiple query groups. Let $M = 2\max_{S_j \in Q} |S_j| + 1$. There exist sets of vectors $\mathcal{Q}, \mathcal{D} \subset \mathbb{R}^M$, with $|\mathcal{Q}| = |Q|$ and $|\mathcal{D}| = |D| + 1$, such that the MaxSim similarity computes the sum of the Weighted Max-OR evaluations for each query group:

$$S(\mathcal{Q}, \mathcal{D}) = \sum_{S_j \in Q} \text{MaxOR}(S_j, D). \tag{15}$$

Importantly, the sets $\mathcal{Q}$ and $\mathcal{D}$ can be constructed independently; the mapping function applied to Q requires no knowledge of D, and the mapping applied to D requires no knowledge of Q.

To prove this theorem, we first generalize our embedding map to arbitrary dimensions, which will allow us to construct polynomials that encode entire OR-clauses.

Definition 5.3 (Generalized Embedding Map)**.** For a given dimension $M \in \mathbb{N}$ where $M \neq 0$, define the generalized embedding map $\phi_M : \mathbb{N} \to \mathbb{R}^M$ as:

$$\phi_M(d) = \begin{pmatrix} 1 & d & d^2 & \ldots & d^{M-1} \end{pmatrix}^T. \tag{16}$$

To encode an entire group S into a single vector, we construct a polynomial that interpolates the weights of the desired keys while evaluating to a negative number for all irrelevant keys.

Lemma 5.1 (OR-Clause Polynomial Construction)**.** For any query group S of size $m = |S|$, there exists a coefficient vector $c(S) = \mathbf{c}_S \in \mathbb{R}^{2m+1}$ defining a polynomial $p_S(t) = \mathbf{c}_S^T \phi_{2m+1}(t)$ such that:

1. $p_S(d) = w$ for all $(d, w) \in S$.
2. $p_S(d') < 0$ for all keys $d' \in \mathbb{N} \setminus D_S$.

Proof. Let $x_S(t)$ be the Lagrange interpolating polynomial of degree at most $m - 1$ that passes through all points in S. By definition, $x_S(d) = w$ for all $(d, w) \in S$.

Define the root polynomial $y_S(t) = \prod_{d \in D_S}(t - d)$. This polynomial has degree m and evaluates to zero exactly on the keys in D_S.

We define our target polynomial as $p_S(t) = x_S(t) - C \cdot (y_S(t))^2$ for some value $C > 0$ selected based on the pairs in S. For any target key $d \in D_S$, $y_S(d) = 0$, so $p_S(d) = x_S(d) = w$, satisfying the first condition.

For any non-target key $d' \in \mathbb{N} \setminus D_S$, we require $p_S(d') < 0$, which is equivalent to requiring $C > \frac{x_S(d')}{(y_S(d'))^2}$. Because d' and all $d \in D_S$ are distinct integers, the difference $(d' - d)$ is a non-zero integer, meaning $(y_S(d'))^2 \geq 1$. This ensures the denominator is never zero and does not shrink arbitrarily small. Furthermore, the degree of $(y_S(t))^2$ is $2m$, which is strictly greater than the degree of $x_S(t)$ (which is at most $m - 1$). Therefore, as $t \to \infty$, the ratio $\frac{x_S(t)}{(y_S(t))^2} \to 0$. Because the ratio

goes to zero asymptotically and is evaluated over discrete integers, it is bounded above by some finite maximum value B for all $d' \in \mathbb{N} \setminus D_S$.

By choosing $C > \max(0, B)$, we guarantee $p_S(d') < 0$ for all $d' \notin D_S$. The maximum degree of $p_S(t)$ is the degree of $(y_S(t))^2$, which is $2m$. Thus, it has $2m + 1$ coefficients, which can be represented by the vector $\mathbf{c}_S \in \mathbb{R}^{2m+1}$. □

Using this polynomial construction, we can now show that the inner maximization step of MaxSim perfectly computes the Weighted Max-OR for a single query group.

Lemma 5.2 (Inner Maximization for a Single Query Group)**.** Let S be a query group and D be a document set. Let $M \geq 2|S| + 1$. Let $c(S) = \mathbf{c}_S \in \mathbb{R}^M$ be the polynomial coefficient vector from Lemma 5.1 (padded with zeros if $2|S| + 1 < M$). Let $\mathcal{D} = \{w_i \phi_M(d_i) \mid (d_i, w_i) \in D\} \cup \{\mathbf{0}\}$ be the document embeddings. Then the maximum inner product evaluates to the Weighted Max-OR:

$$\max_{\mathbf{d} \in \mathcal{D}} \langle \mathbf{c}_S, \mathbf{d} \rangle = \text{MaxOR}(S, D). \tag{17}$$

Proof. For any document vector $\mathbf{d}_i = w_i \phi_M(d_i) \in \mathcal{D}$, the inner product evaluates the polynomial: $\langle \mathbf{c}_S, w_i \phi_M(d_i) \rangle = w_i p_S(d_i)$.

If $d_i \notin D_S$ (a non-matching key), Lemma 5.1 guarantees $p_S(d_i) < 0$. Since document weights $w_i \geq 0$, the inner product is ≤ 0. The zero vector $\mathbf{0} \in \mathcal{D}$ yields $\langle \mathbf{c}_S, \mathbf{0} \rangle = 0$, ensuring these non-matches result in a score of at most 0.

If $d_i \in D_S$ (a matching key), let $(d_i, w_{q,i})$ be the corresponding pair in S. Lemma 5.1 guarantees $p_S(d_i) = w_{q,i}$. The inner product is exactly $w_i w_{q,i} \geq 0$ as both $w_i \geq 0$ and $w_{q,i} \geq 0$.

Therefore, the set of all inner products evaluated by the $\max$ operation is exactly $\{0\} \cup \{w_i w_{q,i} \mid (d_i, w_i) \in D \text{ and } d_i \in D_S\}$. The maximum of this set is, by definition, MaxOR(S, D). □

We now prove the main theorem of this section.

Proof of Theorem 5.1. We construct $\mathcal{D}$ by mapping each document pair $(d_i, w_i) \in D$ using the generalized embedding map ϕ_M (Definition 5.3):

$$\mathcal{D} = \{w_i \phi_M(d_i) \mid (d_i, w_i) \in D\} \cup \{\mathbf{0}\}. \tag{18}$$

We construct $\mathcal{Q}$ by mapping each query group $S_j \in Q$ to its polynomial coefficient vector $\mathbf{c}_{S_j} = c(S_j)$ from Lemma 5.1 (padded with zeros to length M):

$$\mathcal{Q} = \left\{ \mathbf{c}_{S_j} \mid S_j \in Q \right\}. \tag{19}$$

Expanding the MaxSim similarity $S(\mathcal{Q}, \mathcal{D})$, we have:

$$S(\mathcal{Q}, \mathcal{D}) = \sum_{\mathbf{c}_{S_j} \in \mathcal{Q}} \max_{\mathbf{d} \in \mathcal{D}} \langle \mathbf{c}_{S_j}, \mathbf{d} \rangle. \tag{20}$$

By Lemma 5.2, the inner maximization for each query group $S_j \in Q$ exactly computes MaxOR(S_j, D). Substituting this into the sum yields:

$$S(\mathcal{Q}, \mathcal{D}) = \sum_{S_j \in Q} \text{MaxOR}(S_j, D). \tag{21}$$

This completes the proof. □

5.3 Exact Boolean Logic (Positive CNF)

Having established that MaxSim evaluates a Weighted Max-OR, we now show that if we restrict the weights to binary values, MaxSim exactly evaluates standard Conjunctive Normal Form (CNF) Boolean logic without negations, so that a set of scored documents is rank equivalent to the exact evaluation. We start by formally defining rank equivalence.

Definition 5.4 (Rank Equivalence)**.** Let X be a set of items. Given a target scoring function $s : X \to \mathbb{R}$ and an evaluated scoring function $s' : X \to \mathbb{R}$, we say s' is **rank equivalent** to s if for all $x, y \in X$:

$$s(x) > s(y) \implies s'(x) > s'(y). \tag{22}$$

This is useful for asserting equivalence for ranking tasks where the relative ordering matters, but the exact scores do not.

Next, we define the structure of a Boolean document and a positive CNF query.

Definition 5.5 (Boolean Document and Positive CNF Query)**.** Let a document $D \subset \mathbb{N}$ be a set of unique keys. Let $K = T_1 \wedge T_2 \wedge \cdots \wedge T_h$ be a logical query in Conjunctive Normal Form (CNF), where each clause $T_j = t_{j,1} \vee t_{j,2} \vee \cdots \vee t_{j,l_j}$ is a disjunction (OR) of positive keys.

The strict Boolean scoring function $s_K : 2^{\mathbb{N}} \to \{0, 1\}$ is defined as $s_K(D) = 1$ if D satisfies K (i.e., D contains at least one key from every clause T_j), and $s_K(D) = 0$ otherwise.

We now present the main result of this subsection: MaxSim's ability to evaluate positive CNF queries.

Theorem 5.2 (MaxSim Evaluates Positive CNF)**.** Let K be a positive CNF query with h clauses. There exists a mapping from K to a query set $\mathcal{Q} \subset \mathbb{R}^M$, and a mapping from any document D to a document set $\mathcal{D} \subset \mathbb{R}^M$, such that the MaxSim similarity $S(\mathcal{Q}, \mathcal{D})$ is rank equivalent to the strict Boolean scoring function $s_K(D)$, where $|\mathcal{Q}| = h$ and $|\mathcal{D}| = |D| + 1$. Additionally, the sets $\mathcal{Q}$ and $\mathcal{D}$ can be constructed independently; the mapping function applied to K requires no knowledge of D, and the mapping applied to D requires no knowledge of K.

Proof. We construct $\mathcal{D}$ by assigning a unit weight to each key in the document: $D_{set} = \{(d_i, 1) \mid d_i \in D\}$. We construct $\mathcal{Q}$ by creating a query group for each clause T_j, assigning unit weights to its keys: $S_j = \{(t, 1) \mid t \in T_j\}$. We then generate $\mathcal{D}$ and $\mathcal{Q}$ in $\mathbb{R}^M$ as defined in Theorem 5.1.

By Theorem 5.1, the MaxSim similarity is:

$$S(\mathcal{Q}, \mathcal{D}) = \sum_{S_j \in Q} \text{MaxOR}(S_j, D_{set}) \tag{23}$$

$$= \sum_{S_j \in Q} \max\left(\{0\} \cup \{w_i w_{j,i} \mid (d_i, w_i) \in D_{set} \text{ and } (d_i, w_{j,i}) \in S_j\}\right). \tag{24}$$

Because all weights are exactly 1, the product $w_i w_{j,i} = 1$ for all matching keys. The condition that a key exists in both D_{set} and S_j is equivalent to stating $D \cap T_j \neq \emptyset$.

Thus, for a single clause T_j, the Weighted Max-OR evaluates to 1 if $D \cap T_j \neq \emptyset$ (the clause is satisfied), and 0 if $D \cap T_j = \emptyset$ (the clause is unsatisfied). The total similarity $S(\mathcal{Q}, \mathcal{D})$ is exactly the number of clauses in K satisfied by D.

To prove rank equivalence, let D_a and D_b be two documents such that $s_K(D_a) > s_K(D_b)$. Because s_K outputs values in $\{0, 1\}$, this implies $s_K(D_a) = 1$ (satisfies all h clauses) and $s_K(D_b) = 0$ (fails at least one clause). Consequently, $S(\mathcal{Q}, \mathcal{D}_a) = h$, and $S(\mathcal{Q}, \mathcal{D}_b) \leq h - 1$. Therefore, $S(\mathcal{Q}, \mathcal{D}_a) > S(\mathcal{Q}, \mathcal{D}_b)$, satisfying the definition of rank equivalence. □

6 Experiments

In Section 3, we introduce a new late-interaction similarity function, Signed MaxSim $S_{\pm}$, which has theoretical capacities beyond standard MaxSim. In this section, we investigate whether these capacities translate to empirical gains. To better quantify the differences between these methods, we focus on a retrieval task that is more likely to benefit from the ability to replicate any real-valued inner product. Specifically, we focus on retrieval tasks that feature negations or exclusions. We refer to the model trained with the Signed MaxSim similarity $S_{\pm}$ as **Fallon** throughout our experiments.

6.1 Datasets

To isolate the architectural impact on retrieval performance and enable better introspection, we use synthetic retrieval tasks for both training and evaluation. Inspired by the LIMIT dataset [31],

documents are in the format *John Smith likes Alternative Rock, Angel Food Cakes, Apricots, Argentine Ants, Avocados, Blackberries, Cabbages, and Cantaloupes* while queries ask for a person with a specific set of attributes *People who like Alternative Rock and Blackberries but do not like Cherries*. Unlike the original LIMIT dataset, which only has single-attribute queries, our queries can include multiple features required for relevance and additionally can have attributes which should not be present, e.g. *but do not like Cherries*. By using synthetic data we can remove possible confounding factors like validation-test mismatch and limited training data, while evaluating the abilities we believe are important differentiators between standard MaxSim and Signed MaxSim. Additionally, it makes it easy to construct different variants to test robustness and simulate out-of-domain generalization.

6.1.1 Training Data

For training, we generate 100k queries with 200k documents using the original LIMIT attributes and method for name generation. During training, we used contrastive training with in-batch negatives, so we constructed queries to minimize the total number of positive documents to limit the risk of in-batch negatives being true positives. Each query can have one to four inclusion terms (e.g. non-negated terms) and all queries have one negated term which should not be in relevant documents. Each training query is bundled with 32 hard negatives: documents that are relevant except for the fact that they include the negated term. Additional random negatives were included if there were fewer than 32 hard negatives.

6.1.2 Evaluation Benchmarks

For evaluation we investigate three different datasets:

In-Domain This dataset uses the exact same setup as the training data in terms of query formatting and vocabulary. All queries in this dataset are distinct from those seen during training, but the documents may be the same, although the documents are generated separately.

Different Vocabulary This dataset uses the same setup as the in-domain data, except the vocabulary is different. We generate additional vocabulary terms using Gemini conditioned on the original vocabulary. This dataset provides insight into the generalization of the model to new features.

Negation Only This dataset uses the same vocabulary as the in-domain data, but changes the structure of queries to contain only negations; for example, *People who do not like Cherries*. We also allow one or two negations, such as *People who do not like Cherries or Watermelon*, 50% of queries have a single negation. This dataset tests the models' generalization to new query formats.

Each benchmark contains 2,000 queries evaluated against a corpus of 100k documents. Evaluation documents are generated independently from training documents, though incidental overlap is possible.

6.2 Training Setup

We train both models using a contrastive loss with in-batch negatives and a ModernBERT backbone. The main differences are the scoring function (MaxSim vs. Signed MaxSim) and the additional weight (i.e. sign) generation mechanism for Fallon. Below, we describe the loss function, model architecture, and optimization details.

All models are trained with a contrastive cross-entropy loss using in-batch negatives combined with hard negatives. Let $\mathbf{e}^q$ denote the query representation, $\mathbf{e}^{d^+}$ the positive document representation, and $\mathcal{N}$ the set of negatives. The loss uses a jointly learned temperature τ:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(S(\mathbf{e}^q, \mathbf{e}^{d^+})/\tau)}{\exp(S(\mathbf{e}^q, \mathbf{e}^{d^+})/\tau) + \sum_{d^- \in \mathcal{N}} \exp(S(\mathbf{e}^q, \mathbf{e}^{d^-})/\tau)}.$$

Both models use ModernBERT [30] as the backbone, which has a hidden dimension of $d = 768$ and 149 M parameters. We prepend every query with the prefix `"query: "` and truncate both queries and passages to a maximum of 196 tokens. No query-expansion tokens are added following contemporary late-interaction designs [2]. Before scoring, each token's hidden state is projected from $d = 768$ down to $m = 128$ by a five-layer MLP block. For the real-valued model, we use another five-layer

MLP to produce the final per-token weights. The embeddings of both models are ℓ_2-normalized after projection following [14, 27]. We tried using non-normalized embeddings for Fallon, but found it performed worse. We believe this is due to the unbounded document scores resulting in degraded training signal when using contrastive loss. We encourage future work to investigate alternative methods to normalize training.

The baseline ColBERT model otherwise follows the design of Santhanam et al. [27] and encodes queries and documents with a single shared encoder. Relevance is measured by MaxSim (Definition 1.1), and the contrastive loss temperature τ is learned jointly with all other parameters, starting from an initial value of $\tau_0 = 1$.

To implement the Fallon model (i.e. our model trained with Signed MaxSim), there are minimal changes needed when compared with a standard ColBERT model; the main change is the additional MLP used for weight generation. In the theoretical version, we use the sign to produce the weight, which would limit the weights to 1 or -1, presenting a problem for gradient descent. To mitigate this, we allow the weights to be any real value. This allows for gradients to flow without limitations. We tried using Tanh activation as well, but found that with ℓ_2-normalized embeddings training was unstable, thus we stayed with no activation.

The MLP blocks map a token hidden state $\mathbf{h} \in \mathbb{R}^{768}$ to an output $\mathbf{e} \in \mathbb{R}^{d_{\text{out}}}$, where d_{out} is either 128 (for embedding projections) or 1 (for the weight generator). The complete implementation is shown below:

$$
\begin{aligned}
\mathbf{z}_1 &= \mathrm{LN}\big(\mathrm{GELU}(W_1\,\mathbf{h} + \mathbf{b}_1)\big), \\
\mathbf{z}_l &= \mathbf{z}_{l-1} \;+\; \mathrm{LN}\big(\mathrm{GELU}(W_l\,\mathbf{z}_{l-1} + \mathbf{b}_l)\big), \qquad l = 2, 3, 4, \\
\mathbf{e} &= W_5\,\mathbf{z}_4 + \mathbf{b}_5.
\end{aligned}
$$

Here $W_1, \ldots, W_4 \in \mathbb{R}^{768\times768}$ and $W_5 \in \mathbb{R}^{d_{\text{out}}\times768}$, with LN denoting layer normalization and GELU representing the GELU activation function [7].

All models are implemented using Transformers [32] and PyTorch [22]. We use the AdamW optimizer [18] with default hyperparameters and no weight decay. All experiments use BF16 mixed-precision training with TF32 tensor operations. The learning rates were selected based on prior experiments and an LR sweep for ColBERT. We swept learning rates over $\{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$ and selected 1×10^{-5} based on validation nDCG@10. The complete training hyperparameter configurations are available in Table 1. During training, we evaluate each model on a validation dataset every 2,500 steps and select the final model based on validation nDCG@10. We stopped training after the validation metric showed no improvement for three consecutive checkpoints. The validation dataset uses the same setup as the In-Domain evaluation benchmark but with 1,000 queries and 40k documents. We ensure the queries are disjoint from both training and evaluation data.

Table 1: Training hyperparameters per model. *Eff. Batch* denotes effective batch size (per-device batch $\times$ gradient accumulation steps). *Hard Neg.* is the number of hard negatives per positive sampled in a training batch. *Attn. Drop* and *MLP Drop* are the dropout ratios set for the backbone ModernBERT model.

Model	LR	Eff. Batch	Total Steps	Hard Neg.	Warm-up	LR Decay	Attn. Drop	MLP Drop
ColBERT	1×10^{-5}	128	223,640	7	5%	Linear	0.1	0.1
Fallon	8×10^{-5}	128	223,640	7	5%	Linear	0.1	0.1

6.3 Evaluation Procedure

We perform retrieval using exhaustive search: for each query, we compute the exact similarity scores for all documents in the corpus and return the top 1,000 results. We use an exact approach over an approximate method, as our new late-interaction method is not directly compatible with standard approximate methods such as PLAID [26]. It is worth noting that our method does not preclude an approximate search technique, but developing one was outside the scope of this work.

We evaluate using three evaluation metrics. We use nDCG@10 [10] and P@10 to understand the precision of the approaches and how directly the results could be used in real-world applications where users may be unwilling to go beyond the first set of documents. To get a more holistic view

Table 2: Performance comparison between the ColBERT baseline and Fallon across in-domain, different-vocabulary, and negation-only evaluation settings. Here, † represents a statistically significant difference between the baseline ColBERT and our model. Statistical significance was calculated using a paired Student's t-test; we consider any difference with $p < 0.01$ as significant.

Model	In-Domain			Different Vocab.			Negation Only		
	nDCG@10	**P@10**	**AP**	**nDCG@10**	**P@10**	**AP**	**nDCG@10**	**P@10**	**AP**
ColBERT	0.982	0.875	0.943	0.597	0.587	0.511	0.008	0.009	0.002
Fallon	$\mathbf{0.997}^{\dagger}$	$\mathbf{0.893}^{\dagger}$	$\mathbf{0.997}^{\dagger}$	$\mathbf{1.000}^{\dagger}$	$\mathbf{0.955}^{\dagger}$	$\mathbf{1.000}^{\dagger}$	$\mathbf{0.788}^{\dagger}$	$\mathbf{0.789}^{\dagger}$	$\mathbf{0.039}^{\dagger}$

of the retrieval coverage we include Average Precision (AP) which considers both the recall and precision of the retrieved documents. As we retrieve only the top 1,000 documents from a corpus of 100k, AP may have a low upper bound for queries with numerous relevant documents.

7 Results and Discussion

The results of the three test datasets for the baseline ColBERT model and our Fallon model are shown in Table 2. The In-Domain results show that the standard MaxSim and ColBERT approach is capable of learning the synthetic task even though it does not naturally handle negative values, which are a natural way to handle negations. Although ColBERT does well, Fallon, which uses Signed MaxSim, is still significantly better across all the measured metrics.

Moving to the out-of-domain data, we see a much larger gap emerge. For the Different Vocabulary dataset, Fallon actually sees an improvement over the In-Domain dataset (likely due to changes in the number of positives per query), while ColBERT shows a substantial reduction. This result demonstrates that ColBERT's In-Domain performance relies heavily on the shared features between the training and evaluation datasets to work correctly. Fallon, which uses Signed MaxSim, on the other hand, learns a far more general strategy which can directly penalize the presence of unwanted features.

The Negation Only results are the most revealing and provide the strongest evidence for the limitations of standard MaxSim. For Negation Only queries, the majority of the 100k corpus documents are relevant (those lacking the negated attribute), so even random retrieval would do relatively well. Thus, ColBERT's near-zero performance in Table 2 indicates it is actively placing documents that contain the negated attribute at the top of the ranking.

This is a natural consequence of MaxSim. It is very difficult to have a low score for a query embedding, as this requires all document embeddings to have a low score, even those completely unrelated. Because during training ColBERT learns to always match some relevant feature, it struggles when there are no relevant aspects to match.

Fallon resolves this failure, in large part because Signed MaxSim allows the model to easily penalize features by supporting negative values natively. The relatively low AP, while improved dramatically over ColBERT, is largely an artifact of evaluating only the top 1,000 retrieved documents, since for the Negation Only queries where tens of thousands of documents are relevant, any relevant document outside the top 1,000 contributes zero precision regardless of model quality. The remaining gap from perfect performance suggests that there may still be limitations in generalization, though these limits are far less damaging than the limitations demonstrated by standard MaxSim.

Together, these results across the three evaluation settings demonstrate that Signed MaxSim addresses a fundamental limitation of standard MaxSim: the inability to encode semantic negations and complex relational patterns that require suppression of certain token matches. While ColBERT performs reasonably when training and evaluation distributions are closely aligned, its performance degrades dramatically when the data distribution changes. By leveraging more powerful representations and Signed MaxSim similarity, Fallon is able to generalize far better, making it the clear choice in challenging retrieval scenarios.

8 Conclusion

In this work, we established that MaxSim similarity is at least as expressive as inner-product-based retrieval for non-negative vectors, allowing late-interaction models to exactly replicate infinite-dimensional sparse representations using only $O(k)$ space. We proved that this exact replication is impossible for standard finite-dimensional single-vector models, formally separating the capacity of late-interaction models from single-vector retrievers. However, we also proved that standard MaxSim is incapable of exactly replicating the inner product between real-valued vectors, meaning MaxSim is not universally able to replicate inner product similarity. To address this shortcoming, we leverage our theoretical framework to extend standard MaxSim to enable the exact replication of the inner product between any real-valued vectors. This makes our extension, Signed MaxSim, provably as capable as any single-vector inner-product-based retrieval method. Our experimental results confirm the usefulness of Signed MaxSim and highlight that it is substantially more robust than standard MaxSim when evaluated out-of-domain. Furthermore, we showed that MaxSim naturally evaluates positive Conjunctive Normal Form (CNF) expressions by acting as an aggregation of fuzzy ORs. This finding provides another perspective to explain the empirical improvement seen in late-interaction models and connects late-interaction methods to the traditional IR methods which leverage structured Boolean queries.

Our theoretical results raise several open questions regarding the limits of retrieval similarities. While we show that standard MaxSim cannot replicate inner products between real-valued vectors when the number of embeddings is tied to the sparsity of the original vector, we do not investigate the more relaxed case where the number of embeddings is decoupled from sparsity. Whether such a relaxation would enable exact real-value replication remains an open question. In this work, we show that MaxSim can exactly replicate the inner product between sparse infinite-dimension vectors and that a finite-dimension standard inner product cannot replicate this; however, it is an open question whether all similarities expressible by MaxSim could be exactly expressed or closely approximated by a sparse infinite-dimension inner product. We show that MaxSim can act as a rank-equivalent positive CNF evaluator. Whether it is possible for standard MaxSim to act as a rank-equivalent evaluator of all CNFs remains an open question.

Finally, our findings suggest new directions for neural retrieval architectures. Although our polynomial embedding constructions are primarily theoretical tools, they demonstrate that exact logical evaluation is geometrically possible within late-interaction spaces. Future empirical work could investigate whether trained models implicitly learn these localized structures, or if explicitly regularizing embeddings toward these constructions improves out-of-domain generalization.

Acknowledgments and Disclosure of Funding

The authors would like to thank Vignesh Viswanathan for providing feedback on an early version of this work and Antoine Chaffin for answering questions about ColBERT training.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the NSF Graduate Research Fellowship Program Award #1938059, and in part by the Office of Naval Research contract number N000142412612. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

The authors used AI technology in various capacities while producing this work, including drafting proofs and other text, checking writing, generating code for experiments, and producing figures and tables. All AI outputs were checked by the authors for correctness, and the authors take full responsibility for the content of this paper.

References

[1] N. Alon, S. Moran, and A. Yehudayoff. Sign rank versus VC dimension. In V. Feldman, A. Rakhlin, and O. Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 47–80. JMLR.org, 2016. URL `http://proceedings.mlr.press/v49/alon16.html`.

[2] A. Chaffin, L. Arnaboldi, A. Chatelain, and F. Krzakala. Colbert-zero: To pre-train or not to pre-train colbert models, 2026. URL `https://arxiv.org/abs/2602.16609`.

[3] S. Chanpuriya, C. Musco, K. Sotiropoulos, and C. E. Tsourakakis. Node embeddings and exact low-rank representations of complex networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/99503bdd3c5a4c4671ada72d6fd81433-Abstract.html.

[4] S. Chanpuriya, R. A. Rossi, A. B. Rao, T. Mai, N. Lipka, Z. Song, and C. Musco. Exact representation of sparse networks with symmetric nonnegative embeddings. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/428ceef2cd8a53add7213e04d1746479-Abstract-Conference.html.

[5] L. Dhulipala, M. Hadian, R. Jayaram, J. Lee, and V. Mirrokni. MUVERA: multi-vector retrieval via fixed dimensional encodings. *CoRR*, abs/2405.19504, 2024. doi: 10.48550/ARXIV.2405.19504. URL https://doi.org/10.48550/arXiv.2405.19504.

[6] T. Formal, B. Piwowarski, and S. Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3463098. URL https://doi.org/10.1145/3404835.3463098.

[7] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[8] S. Hofstätter, O. Khattab, S. Althammer, M. Sertkan, and A. Hanbury. Introducing neural bag of wholewords with colberter: Contextualized late interactions using enhanced reduction. In M. A. Hasan and L. Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 737–747. ACM, 2022. doi: 10.1145/3511808.3557367. URL https://doi.org/10.1145/3511808.3557367.

[9] E. Jääsaari, V. Hyvönen, and T. Roos. LEMUR: learned multi-vector retrieval. *CoRR*, abs/2601.21853, 2026. doi: 10.48550/ARXIV.2601.21853. URL https://doi.org/10.48550/arXiv.2601.21853.

[10] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002. doi: 10.1145/582415.582418. URL http://doi.acm.org/10.1145/582415.582418.

[11] R. Jayaram. Multi-vector embeddings are provably more expressive than single vector embeddings, 2026. URL https://arxiv.org/abs/2606.23475.

[12] Z. Ji, H. Jain, A. Veit, S. J. Reddi, S. Jayasumana, A. S. Rawat, A. K. Menon, F. X. Yu, and S. Kumar. Efficient document ranking with learnable late interactions. *CoRR*, abs/2406.17968, 2024. doi: 10.48550/ARXIV.2406.17968. URL https://doi.org/10.48550/arXiv.2406.17968.

[13] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih. Dense passage retrieval for open-domain question answering. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

[14] O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In J. X. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM, 2020. doi: 10.1145/3397271.3401075. URL https://doi.org/10.1145/3397271.3401075.

[15] J. Killingback, H. Zeng, and H. Zamani. Hypencoder: Hypernetworks for information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pages 2372–2383, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3729983. URL https://doi.org/10.1145/3726302.3729983.

[16] J. Killingback, O. Meshi, H. Li, H. Zamani, and M. Karimzadehgan. A unified model and document representation for on-device retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2604.14403.

[17] J. Killingback, M. Rafiee, M. Manas, and H. Zamani. Scaling laws for embedding dimension in information retrieval, 2026. URL https://arxiv.org/abs/2602.05062.

[18] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL https://api.semanticscholar.org/CorpusID:53592270.

[19] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021. doi: 10.1162/TACL_A_00369. URL https://doi.org/10.1162/tacl_a_00369.

[20] S. MacAvaney, A. Mallia, and N. Tonellotto. Efficient constant-space multi-vector retrieval. In C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, and N. Tonellotto, editors, *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part III*, volume 15574 of *Lecture Notes in Computer Science*, pages 237–245. Springer, 2025. doi: 10.1007/978-3-031-88714-7_22. URL https://doi.org/10.1007/978-3-031-88714-7_22.

[21] A. Menon, S. Jayasumana, A. S. Rawat, S. Kim, S. Reddi, and S. Kumar. In defense of dual-encoders for neural ranking. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15376–15400. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/menon22a.html.

[22] A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[23] Y. Qian, J. Lee, S. M. K. Duddu, Z. Dai, S. Brahma, I. Naim, T. Lei, and V. Y. Zhao. Multi-vector retrieval as sparse alignment. *CoRR*, abs/2211.01267, 2022. doi: 10.48550/ARXIV.2211.01267. URL https://doi.org/10.48550/arXiv.2211.01267.

[24] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, page 232–241, Berlin, Heidelberg, 1994. Springer-Verlag. ISBN 038719889X.

[25] A. S, M. Agarwal, A. Garg, N. Kayal, and K. Shiragur. On strengths and limitations of single-vector embeddings, 2026. URL https://arxiv.org/abs/2603.29519.

[26] K. Santhanam, O. Khattab, C. Potts, and M. Zaharia. PLAID: an efficient engine for late interaction retrieval. In M. A. Hasan and L. Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1747–1756. ACM, 2022. doi: 10.1145/3511808.3557325. URL https://doi.org/10.1145/3511808.3557325.

[27] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In M. Carpuat, M. de Marneffe, and I. V. M. Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3715–3734. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.NAACL-MAIN.272. URL https://doi.org/10.18653/v1/2022.naacl-main.272.

[28] R. Sourty, A. Chaffin, O. Weller, P. R. Moura Junior, and A. Chatelain. Denseon with the lateon: Open state-of-the-art single and multi-vector models. https://huggingface.co/blog/lightonai/denseon-lateon, 2026.

[29] J. Veneroso, R. Jayaram, J. Rao, G. H. Ábrego, M. Hadian, and D. Cer. CRISP: clustering multi-vector representations for denoising and pruning. *CoRR*, abs/2505.11471, 2025. doi: 10.48550/ARXIV.2505.11471. URL https://doi.org/10.48550/arXiv.2505.11471.

[30] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, and I. Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL https://arxiv.org/abs/2412.13663.

[31] O. Weller, M. Boratko, I. Naim, and J. Lee. On the theoretical limitations of embedding-based retrieval. *CoRR*, abs/2508.21038, 2025. doi: 10.48550/arXiv.2508.21038. URL https://doi.org/10.48550/arXiv.2508.21038.

[32] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-art natural language processing. In Q. Liu and D. Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

[33] H. Zamani, M. Dehghani, W. B. Croft, E. Learned-Miller, and J. Kamps. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 497–506, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271800. URL https://doi.org/10.1145/3269206.3271800.