Adicionando suporte à diversificação de resultados em índices HNSW considerando espaços de baixa e alta dimensionalidade

Mauro Weber[1], João Silva-Leite[1], Lúcio F. D. Santos[2], Daniel de Oliveira[1], Marcos Bedo[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Gal. Milton Tavares de Souza, S/N – Niterói/RJ, Brasil

[2]Instituto Federal do Norte de Minas Gerais (IFNMG)
Prof. Monteiro Fonseca, 216 – Montes Claros/MG, Brasil

{mauro_weber, joaovitorleite}@id.uff.br

{danielcmo, marcosbedo}@ic.uff.br, lucio.santos@ifnmg.edu.br

Abstract. *Hierarchical Navigable Small World (HNSW) indexes provide state-of-the-art performances for approximate k-nearest neighbor (kNN) queries. However, binding the approximate search quality with the characterization of the HNSW construction heuristic remains an open issue. This article investigates if result diversification can bridge this gap by discussing a new HNSW construction strategy based on a local, diversification-driven partitioning principle. Accordingly, we extend the HNSW kNN search algorithm to support result diversification. Experimental evaluations on* `ANN-Benchmark` *showed while diversification-based partitioning improves the search Recall, standard construction still yields higher throughput. We examined this trade-off through the lens of Local Intrinsic Dimensionality (LID), stratifying the datasets into quartiles. This evaluation indicated the throughput difference shrinks with the LID, with the diversification-based construction yielding a higher Recall in every LID-based manifold. Such outcomes suggest that HNSW long edges' tuning may depend on the LID manifold.*

Resumo. *Índices do tipo Hierarchical Navigable Small World (HNSW) apresentam desempenhos estado-da-arte em consultas aproximadas aos k-vizinhos mais próximos (kNN). Não obstante, caracterizar a estratégia de construção destes índices e seu impacto na qualidade da busca aproximada ainda é um desafio em aberto. Este artigo investiga como a diversificação de resultados pode contribuir para esta caracterização ao discutir uma nova construção para o HNSW que utiliza a perspectiva dos objetos de consulta para gerar regiões diversificadas. Nesse sentido, o algoritmo de busca kNN do HNSW também é estendido para dar suporte à diversificação de resultados. Avaliações experimentais no* `ANN-Benchmark` *mostram que, embora o particionamento com diversidade melhore substancialmente a qualidade da busca, a estratégia HNSW atinge uma maior taxa de vazão. Para entender melhor esse balanço, foi utilizado o conceito da Dimensionalidade Intrínseca Local (LID) para estratificar os dados em quartis de dificuldade. Essa avaliação mostrou que a diferença de vazão entre as duas construções diminui com a LID, enquanto que a qualidade das consultas permanece maior no particionamento por diversidade. Esses resultados sugerem que o ajuste do HNSW depende da distribuição de distâncias.*

1. Introdução

O crescimento de aplicações baseadas em aprendizado profundo (*deep learning*) deu início à produção massiva de conjuntos de dados com representações vetoriais de textos e imagens representados em espaços de alta dimensionalidade. Uma consulta típica sobre essas representações é a busca aos k-vizinhos mais próximos (kNN). Não obstante, soluções exatas baseadas em índices cujo particionamento obedece às propriedades dos Espaços Métricos são ineficientes na execução dessas buscas em espaços de alta-dimensionalidade devido ao fenômeno de *concentração de distâncias* [Volnyansky and Pestov 2009, He et al. 2012, Houle 2013].

Índices com heurísticas para obtenção de soluções aproximadas são mais eficientes nestes casos. Em particular, o índice *Hierarchical Navigable Small World* (HNSW) oferece desempenho estado-da-arte para consultas aproximadas em conjuntos de dados de alta dimensionalidade [Aumüller et al. 2020, Malkov and Yashunin 2016, Santana and Ribeiro 2023]. Este índice é baseado em uma estrutura hierárquica onde cada camada é um grafo conectado em que: *(i)* o número de arestas (distância para vizinhos conhecidos) de cada nó é limitado por um parâmetro de construção e *(ii)* os nós são alcançáveis com poucos saltos [Malkov and Yashunin 2016, Aumüller and Ceccarello 2021, Wang et al. 2021].

O HNSW emprega uma heurística gulosa para selecionar e reorganizar arestas, adotando um critério baseado em hiperplanos para evitar construir arestas com elementos que estão mais próximos de vizinhos previamente inseridos, o que permite a construção de arestas longas [Malkov and Yashunin 2016, Peng et al. 2022]. O algoritmo de busca kNN do HNSW é baseado em uma busca em profundidade, que ordena parcialmente os nós visitados em duas filas de prioridade [Li et al. 2021, Shimomura et al. 2021, Peng et al. 2022]. Portanto, dois parâmetros são importantes para o ajuste fino do HNSW: *(i)* M, o grau de nó; e *(ii)* ef, o tamanho máximo das filas de prioridade [Malkov and Yashunin 2016, Santana and Ribeiro 2023].

Este ajuste de parâmetros pode ofuscar questões relevantes envolvendo a caracterização do comportamento do HNSW, a saber: (**Q1**) *como outros princípios de particionamento afetam o comportamento do HNSW?* (**Q2**) *como o HNSW pode ser estendido para resolver buscas mais complexas, por exemplo, buscas kNN com diversificação de resultados (kN$_d$N)?.* Este artigo avalia estas questões por meio do princípio de particionamento em bola, conhecido como *Influência* [Santos et al. 2013, Jasbick et al. 2020, Jasbick et al. 2023]. Esse princípio não requer parâmetros do usuário e particiona o espaço de busca da perspectiva de um objeto de consulta, separando elementos *Influenciados* por vizinhos mais próximos em bolas cuja cobertura aumenta monotonicamente com a quantidade de objetos indexados.

A premissa aqui investigada é a utilização deste critério de particionamento para construir a última camada (grafo) do HNSW, bem como estender o algoritmo de busca kNN do HNSW para dar suporte à diversificação por *Influência*, mensurando o quanto este novo critério pode melhorar a qualidade (*Recall*) e a vazão (*Queries per second* – QPS) deste tipo de índice. A implementação foi realizada na biblioteca `nsmlib`[1]

[1]Disponível em `https://github.com/nmslib/hnswlib`

que foi acoplada ao benchmark `ANN-Benchmark`[2] para fins de comparação com o HNSW [Shimomura et al. 2021, Aumüller et al. 2020].

Avaliações experimentais no `ANN-Benchmark` sobre quatro conjuntos de dados mostram que o particionamento do espaço de busca por *Influência* pode melhorar a qualidade das buscas $k\text{N}_d\text{N}$. No entanto, a estratégia de construção padrão do HNSW pode alcançar maior vazão média. Para compreender melhor essa troca qualidade/tempo, foi empregado o conceito de *Dimensionalidade Intrínseca Local* (LID) para estratificar os conjuntos de dados em *quartis* de acordo com o grau de dificuldade da busca [Amsaleg et al. 2018, Amsaleg et al. 2019, Aumüller and Ceccarello 2021].

Nessa avaliação detalhada foi observado que a diferença de vazão entre os particionamentos diminui com a LID, enquanto que o particionamento por *Influência* obtém valores médios de Recall maiores em todos os estratos. De acordo com estes achados, foram examinadas as distribuições de arestas produzidas pelos dois particionamentos, de onde se observou que a construção por *Influência* produz distribuições menos concentradas, sugerindo que o ajuste fino do HNSW pode estar ligado à distribuição de LIDs.

O restante deste artigo é como se segue. A Seção 2 apresenta conceitos e trabalhos relacionados. A Seção 3 discute a implementação da proposta. A Seção 4 apresenta a avaliação experimental, enquanto a Seção 5 fornece as conclusões e trabalhos futuros.

2. Conceitos e Trabalhos Relacionados

Consultas por Similaridade. Funções de distância (δ) medem a proximidade entre objetos e são o principal componente de buscas por similaridade. As distâncias de Minkowski, incluindo $\delta = L_2$, quantificam a dissimilaridade entre vetores. Já a distância Angular, enfatiza direção em vez de magnitude. A organização das distâncias de um conjunto de dados para um objeto de consulta define um critério de busca, como o kNN.

Consultas kNN. Uma consulta kNN recupera o conjunto dos k elementos mais próximos em um conjunto de dados $\mathcal{O} \subset \mathbb{R}^d$ para um objeto de referência $o_q \in \mathbb{R}^d$. Incrementalmente, um conjunto kNN $(o_q, \delta, k, \mathcal{O}) = o_1, o_2, ..., o_k$ é formalizado como:

$$o_1 = o_i \in \mathcal{O},\ \forall\ o_j \in \mathcal{O},\ \delta(o_i, o_q) \leq \delta(o_j, o_q),$$
$$o_{m=2,\ldots,k} = o_i \in \mathcal{O} \setminus \cup_{h=1}^{m-1} o_h,\ \forall\ o_j \in \mathcal{O} \setminus \cup_{h=1}^{m-1} o_h, \delta(o_i, o_q) \leq \delta(o_j, o_q)$$

HNSW. O HNSW é um índice em camadas, onde cada camada é construída incrementalmente como um grafo conexo [Santana and Ribeiro 2023]. A camada mais profunda é um "mundo pequeno" onde o número de arestas é limitado por um parâmetro definido pelo usuário, de modo que *(i)* cada nó esteja a poucos saltos um do outro e *(ii)* a travessia da estrutura seja limitada pelo grau dos nós [Malkov and Yashunin 2016].

O HNSW permite aproximar consultas kNN sem alterar o critério de busca. A diversificação de resultados complementa esse critério, permitindo recuperar objetos diferentes entre si [Drosou et al. 2017]. A medida de *Influência* estabelece intervalos de distância para podar vizinhos, garantindo a diversificação [Jasbick et al. 2023].

[2]Disponível em `https://ann-benchmarks.com/`

Medida de *Influência*. Sejam três objetos $o_i, o_j, o_q \in \mathbb{R}^d, o_i \neq o_j \neq o_q$ e uma função de distância δ, a *Influência* entre o_i, o_j é dada por $I(o_i, o_j) = 1/\delta(o_i, o_j)$. Além disso, se o_q é a referência de consulta e o_i é um vizinho diversificado (não *Influenciado*), então suas *Influências* para o_j definem uma relação ternária onde o_j é mais *Influenciado* por o_i do que por o_q *se e somente se* $I(o_i, o_j) > I(o_j, o_q)$.

Conjunto de *Influência*. O *Conjunto de Influência* de um vizinho diversificado o_i para um objeto de referência o_q ($\ddot{I}_{o_i,o_q}$) cobre toda entrada o_j de um conjunto de dados $\mathcal{O} \setminus \{o_i, o_q\} \subseteq \mathbb{R}^d$ que é *(i)* mais distante de o_q do que o_i e *(ii)* mais *Influenciado* por o_i do que por o_q, *ou seja*, $\ddot{I}_{o_i,o_q} = \{o_j \mid o_j \in \mathcal{O} \setminus \{o_i, o_q\}, I(o_i, o_j) > I(o_i, o_q) \wedge I(o_i, o_j) > I(o_j, o_q) \wedge I(o_i, o_q) \neq I(o_j, o_q)\}$.

Consultas kN$_d$N. Uma busca kNN com diversificação de resultados (kN$_d$N) recupera os k elementos mais próximos e não-*Influenciados* de $\mathcal{O} \subseteq \mathbb{R}^d$ para um objeto de consulta $o_q \in \mathbb{R}^d$, de modo que kN$_d$N $(o_q, \delta, k, \mathcal{O}) = \mathcal{R} = \{o_1, o_2, ..., o_k\}$:

$$
\begin{gathered}
o_1 = o_i \in \mathcal{O},\ \forall\, o_j \in \mathcal{O}, \delta(o_i, o_q) \leq \delta(o_j, o_q),\\
o_{m=2,\ldots,k} = o_i \in \mathcal{O},\ (\forall\, o_j \in \cup_{h=1}^{m-1} o_h \Rightarrow o_i \notin \ddot{I}_{o_j,o_q}) \wedge\ (\forall\, o_g \in \mathcal{O} \setminus \cup_{h=1}^{m-1} o_h \Rightarrow\\
(\delta(o_i, o_q) \leq \delta(o_g, o_q) \vee \exists\, o_j \in \cup_{h=1}^{m-1} o_h \Rightarrow o_g \in \ddot{I}_{o_j,o_q})).
\end{gathered}
$$

Concentração de distâncias. O fenômeno de *concentração de distância* afeta o desempenho de busca kNN, pois as distâncias produzidas por certas funções (*e.g.*, L_2) convergem com o aumento da dimensionalidade dos dados em torno de um valor médio com pequena variância, aumentando expressivamente a probabilidade de dois elementos serem indistinguíveis em termos de distância [Volnyansky and Pestov 2009].

Medidas de Concentração. O nível de concentração pode ser quantificado por diversas medidas, como a *Variância Relativa* (RV), que representa a razão entre a variância (σ) e a média das distâncias, *i.e.*, $RV(\mathcal{O} \subset \mathbb{R}^d) = \sigma(\delta(o_i, o_j))/\mu(\delta(o_i, o_j)),\ \forall\, o_i, o_j \in \mathcal{O}, o_i \neq o_j$; ou a *Dimensionalidade Intrínseca* (ID), que estima a estrutura de distâncias dentro de conjunto de dados $ID(\mathcal{O} \subset \mathbb{R}^d) = \mu(\delta(o_i, o_j))^2/2\sigma(\delta(o_i, o_j))^2,\ \forall\, o_i, o_j \in \mathcal{O}, o_i \neq o_j$.

Dimensionalidade Intrínseca Local (LID). Embora as medidas de RV e ID quantifiquem a concentração de forma global, consultas kNN também dependem de um aspecto de *localidade*, *i.e.*, o objeto o_q. Seja F a distribuição cumulativa sobre as distâncias em $\mathcal{O}$ para qualquer objeto o_1, então a função contínua LID (LID_F) para um limiar de distância $r \in \mathbb{R}_+$ é $LID_F(r) := \lim_{h \to 0^+} (\ln(F((1+h) \cdot r)) - \ln(F(r))/\ln(1+h))$, sempre que o limite existir [Amsaleg et al. 2019].

A função LID_F pode ser numericamente aproximada pela medida de Máxima Verossimilhança (MLE) de Amsaleg et *al.* (2019), ao se utilizar altos valores de vizinhança, *e.g.*, $k = 100$. Sejam as distâncias dos elementos $o_i \in \mathcal{O}$ ordenadas para um objeto de consulta o_q, *i.e.*, $\delta(o_q, o_1) \leq \ldots \leq \delta(o_q, o_k)$, então a LID pode ser aproximada pelo MLE como $LID(o_q, k, \delta, \mathcal{O}) = -\left(\frac{1}{k} \cdot \sum_{i=1}^{k} \ln \frac{\delta(o_q, o_i)}{\delta(o_q, o_k)}\right)^{-1}$

3. Materiais e Métodos

3.1. Particionamento e busca *k*NN em índices HNSW

Índices HNSW apresentam desempenho estado-da-arte para consultas por similaridade, estando incluídos em *engines* escaláveis para nuvem como *Amazon OpenSearch*, *Azure Elastic* ou *Apache Lucene* [Aumüller et al. 2020, Wang et al. 2021, Xian et al. 2024]. Este índice pode ser construído de acordo com uma de duas estratégias: *(i)* conectando cada objeto aos seus M vizinhos mais próximos, ou *(ii)* utilizando uma heurística para conectar incrementalmente cada objeto ao seu vizinho mais próximo que não tenha sido descartado por hiperplanos definidos por vizinhos anteriores [Malkov and Yashunin 2016].

Essa heurística permite a construção de "arestas longas" que facilitam a travessia do grafo e é comumente usada como a construção padrão do HNSW. O HNSW utiliza uma hierarquia de *skip lists*, onde as camadas superiores contêm exponencialmente menos objetos. Cada elemento inserido é emparelhado com seu vizinho mais próximo em cada camada até alcançar a camada mais profunda. Nesse ponto, o elemento é conectado

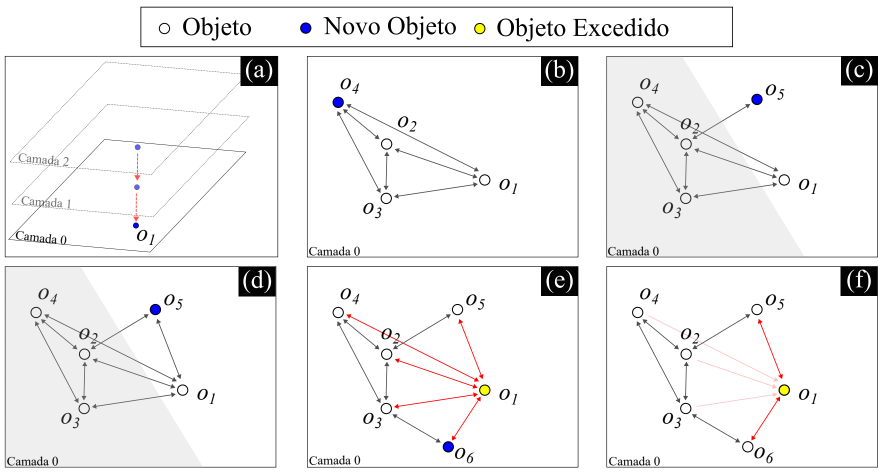

Figura 1. Construção clássica HNSW baseada em hiperplanos em $\mathbb{R}^2$ ($M = 4$).

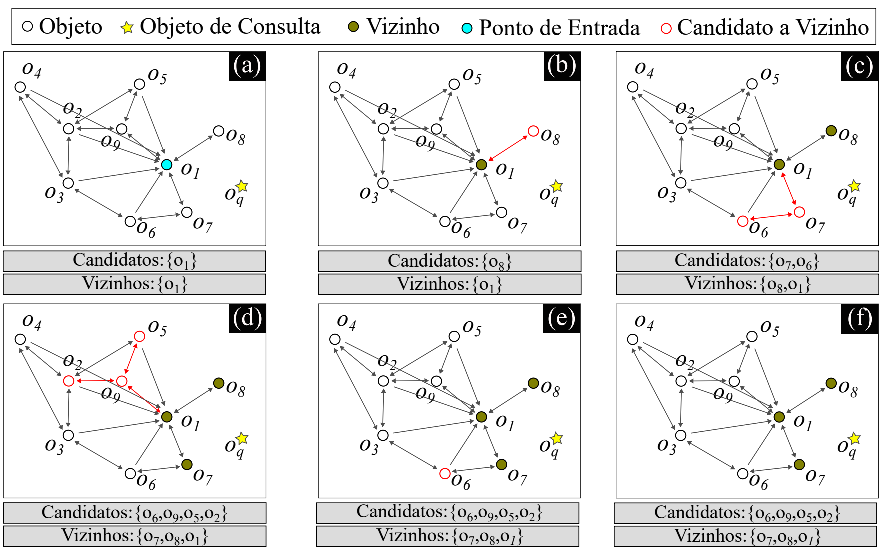

Figura 2. Exemplo de uma consulta kNN com $k = 3$ em $\mathbb{R}^2$.

com até M vizinhos de acordo com o particionamento por hiperplano. A Figura 1 ilustra o início de uma inserção incremental dos objetos $o_1, \ldots, o_9$ para $M = 4$. Inicialmente, os objetos atravessam a estrutura até a camada inferior - Figura 1(a). Os objetos $o_1, \ldots, o_4$ formam um grafo conexo já que a última camada possui M elementos - Figura 1(b). Na sequência, o elemento o_5, ao alcançar a camada inferior, é ligado ao seu vizinho mais próximo o_2, gerando um hiperplano que descarta todos os demais objetos como candidatos, exceto o objeto o_1, que é tomado como vizinho de o_5. Se o grau de um nó ultrapassa M conexões após uma inserção, então suas arestas são reestruturadas - Figura 1(e-f).

Para resolver consultas kNN, o algoritmo atravessa o HNSW usando o elemento mais próximo do objeto de consulta como ponto de partida. Em seguida, ele utiliza duas filas de prioridade (uma para a lista de candidatos e outra para o conjunto resposta) para encontrar os demais vizinhos. A Figura 2 mostra o passo a passo da consulta kNN ($k = 3$) para a última camada do HNSW com objeto mais próximo o_1. Após, a inspeção de o_1 seus vizinhos (o_2, o_7, o_9) serão inseridos na fila de candidatos e, à medida que esses vizinhos são inspecionados, seus vizinhos também serão inseridos na fila de candidatos. Os candidatos são postos na fila de vizinhos caso a fila não tenha k elementos, ou caso a distância ao objeto de consulta seja menor do que um dos objetos na fila de vizinhos.

3.2. Uma nova estratégia de particionamento (em bola) para o HNSW

O particionamento em bola permite pré-definir regiões que não apenas são adequadas para a recuperação kNN, mas que também auxiliam na implementação de determinados critérios de diversificação de resultados, *e.g.*, algoritmos Motley e r-disc [Drosou et al. 2017]. Em particular, o particionamento baseado em *Influência* é um candidato natural para estender o particionamento por hiperplano do HNSW já que *(i)* é baseado em limiares dinâmicos (o raio de cobertura da *bola*) que são induzidos pela localidade do objeto inserido e *(ii)* as partições geradas podem ser eficientemente varridas durante uma busca com diversificação de resultados [Jasbick et al. 2023].

Assim, propõe-se uma extensão da construção HNSW (aqui denominada *d*HNSW) que usa o critério de *Influência* para particionar a última camada HNSW. A Figura 3 ilustra a abordagem proposta com um exemplo para $M = 4$. A lógica *skip list* continua mantida, bem como a construção do grafo totalmente conexo para os primeiros

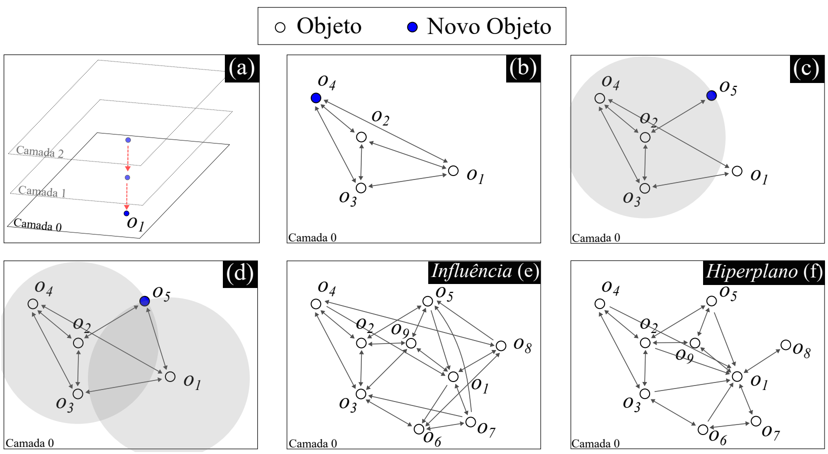

Figura 3. A proposta de particionamento por *Influência* para o HNSW ($M = 4$).

M objetos. A partir daí, cada inserção de elemento irá criar uma aresta entre o objeto inserido e seu primeiro vizinho. Essa vinculação define uma região de *Influência* centrada no primeiro vizinho cuja cobertura é igual a aresta – Figura 3(c). Elementos cobertos por essa bola aberta são descartados como *Influenciados* pelo primeiro vizinho e não são considerados para criar outras arestas (durante esta inserção). Essas etapas se repetem para o próximo vizinho (criando uma nova bola de cobertura) até que M arestas sejam criadas ou não restem candidatos válidos – Fig. 3(d). Caso o grau de um nó exceda M, suas arestas são reestruturadas. As Figuras 3(e–f) destacam as diferenças entre o HNSW e o *d*HNSW. O HNSW cria arestas longas, evitando conexões com objetos dentro do espaço limitado pelo hiperplano. O *d*HNSW também cria arestas longas, porém mantém as conexões mais próximas juntas, pois a cobertura por *Influência* aumenta suavemente com a vizinhança.

3.3. Um algoritmo $k\mathrm{N}_d\mathrm{N}$ aproximado com HNSW

O particionamento por *Influência* também permite estender a rotina de busca kNN do HNSW para realizar consultas $k\mathrm{N}_d\mathrm{N}$ aproximadas. O Algoritmo 1 apresenta o passo-a-passo da consulta na última camada do *d*HNSW. A ideia é se valer de duas filas de prioridade (uma para candidatos $\mathcal{C}$ e outra para o conjunto resposta $\mathcal{K}$), além de um mapa de *bits* que indica objetos já comparados ($\mathcal{V}$). O conjunto reposta sempre inclui o elemento de entrada da camada (pois ele é o primeiro vizinho mais próximo) e define a primeira região de *Influência* excluída. Assim, a lista de candidatos é construída percorrendo-se ordenadamente o conjunto de arestas de cada elemento incluído no conjunto resposta. O Algoritmo 1 percorre no máximo k caminhos (Linha 5), cada um exigindo no máximo M cálculos de distância (Linha 8), realizando no máximo $M \cdot \sum_{r=1}^{k} r$ comparações por *Influência* (Linha 10), o que limita os cálculos de distância em $O\left(kM((3+k)/2)\right)$.

A Figura 4 mostra um exemplo de busca para um objeto de consulta o_q e $k = 3$. O ponto de entrada é o_1 e seu conjunto de arestas leva aos objetos o_6, o_7, o_8 e o_9. Os elementos o_6, o_7, o_8 e o_9 são marcados como visitados, porém o elemento o_7 não é incluído no conjunto de candidatos pois se encontra *Influenciado* por o_1. A próxima aresta livre leva a o_8 que é topo da lista de candidatos e, na iteração seguinte, recuperado como o próximo vizinho diversificado – Figura 3(b–d). Então, o objeto o_6 passa a ser o topo da lista de candidatos e, como não é *Influenciado* por o_1 nem por o_8, é recuperado como

Busca $k\mathrm{N}_d\mathrm{N}$ (Objeto de consulta o_q, #vizinhos k, primeiro vizinho o_p);

1 $\mathcal{C} \leftarrow \{o_p\}$; `/* Fila de prioridade para candidados */`
2 $\mathcal{K} \leftarrow \emptyset$; `/* Fila de prioridade para vizinhos */`
3 $\mathcal{V} \leftarrow \{o_p\}$; `/* Mapa de bits de elementos examinados */`
4 $\mathcal{L} \leftarrow \emptyset$; `/* Fila de prioridade auxiliar para varrer arestas */`
5 **while** $\mathcal{C} \neq \emptyset \wedge |\mathcal{K}| < k - 1$ **do**
6     $o_i \leftarrow \mathcal{C}$.removerTopo();
7     $\mathcal{K} \leftarrow \mathcal{K} \cup \{o_i\}$;
8     $\mathcal{L} \leftarrow$ `vizinhosConectadosPorArestas`$(\langle o_i, \delta(o_q, o_i)\rangle)$;
9   **for** $o_j \in \mathcal{L}$ **do**
10   **if** $o_j \notin \mathcal{V} \wedge o_j \notin \ddot{I}_{o_r,o_q}, \forall\, o_r \in \mathcal{K}$ **then** $\mathcal{C} \leftarrow \mathcal{C} \cup \{o_j\}$;
11         $\mathcal{V} \leftarrow \mathcal{V} \cup \{o_j\}$;
12 return $\mathcal{K} \cup \{o_i, o_i \leftarrow \mathcal{C}$.removerTopo()$\}$;

Algoritmo 1: O algoritmo de busca $k\mathrm{N}_d\mathrm{N}$ para HNSW.

o terceiro vizinho - Figura 3(e–f). Embora o algoritmo pare após preencher a fila de vizinhos, é fácil ver que um parâmetro $ef > k$ pode prolongar a avaliação como na busca kNN. Não obstante, a premissa da proposta é que essa abordagem não é adequada para consultas kN$_d$N, já que a diversificação naturalmente garante que mais regiões do espaço de busca estejam cobertas antes do algoritmo finalizar, como nas Figuras 3(b–f) e 4(b–f).

3.4. Medindo a qualidade de buscas aproximadas kN$_d$N

A qualidade de consultas aproximadas é medida em termos de revocação (Recall), que capturam a proporção entre as distâncias dos vizinhos exatos e aproximados. Em [Kucuktunc and Ferhatosmanoglu 2013], a medida de Recall é generalizada como a escala das distâncias médias dos objetos nos conjuntos resposta $\mathcal{A}$ (kNN) e $\mathcal{B}$ (kNN aproximado), onde Recall $= \sum_{o_i \in \mathcal{A}} \delta(o_q, o_i) / \sum_{o_j \in \mathcal{B}} \delta(o_q, o_j)$, Recall $\in [0, 1]$.

Entretanto, não é possível estender diretamente essa medida para busca kN$_d$N já que as distâncias dentro dos conjuntos exatos não são limitadas pelo resultado aproximado. Por exemplo, a Figura 5(a–b) apresenta um conjunto de dados $\mathcal{O} = \{o_1, \ldots, o_5\}$ cujas distâncias para o objeto de consulta são 2.38, 2.54, 2.58, 2.97 e 3.06, respectiva-

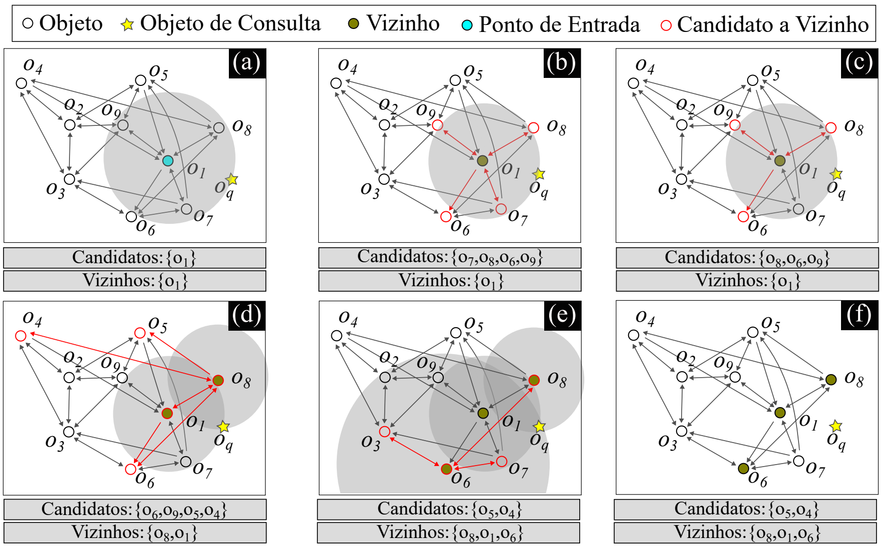

Figura 4. Exemplo de uma consulta kN$_d$N em um índice *d*HNSW para $k = 3$.

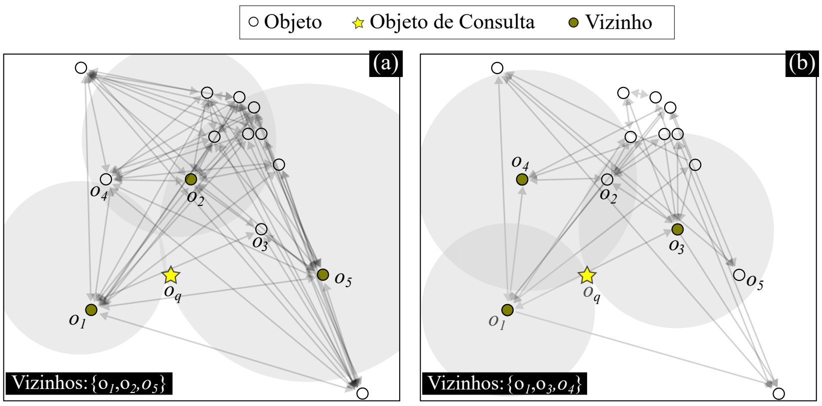

Figura 5. Conjuntos-resposta (a) exato e (b) approximado para uma busca kN$_d$N.

Tabela 1. Conjuntos de dados avaliados

Conjunto	$\|\mathcal{O}\|$	$\|d\|$	$\lceil ID \rceil$	1QT	4QT	Distância
MNIST	70K	784	26	[0, 10.91)	(15.98, 89.92]	L_2
FASHION-MNIST	70K	784	26	[0, 10.59)	(18.31, 101.48]	L_2
SIFT	1.01M	128	15	[0, 15.90)	(23.67, 703.19]	L_2
GLOVE	1.19M	100	16	[0, 12.39)	(23.24, 50.30]	Angular

mente. O resultado exato é $\{o_1, o_2, o_5\}$ com soma das distâncias 7.98, enquanto a resposta aproximada inclui $\{o_1, o_3, o_4\}$ com soma das distâncias 7.93. Portanto, substituir o kNN por kN$_d$N na expressão acima resulta em Recall $= 7.98/7.93 > 1$, *i.e.*, fora do intervalo $[0, 1]$. Isso ocorre porque uma região de *Influência* exata foi ignorada devido à ausência de uma aresta no HNSW. Para garantir que esse efeito seja considerado, propõe-se estender a expressão adotando a diferença absoluta entre as distâncias (em escala) para ambos os conjuntos-resposta kN$_d$N exato ($\mathcal{R}$) e kN$_d$N aproximado ($\mathcal{K}$), conforme a Eq 1.

$$\text{Recall} = \left(k - \textstyle\sum_{o_j \in \mathcal{K}, o_i \in \mathcal{R}} |\delta(o_q, o_j) - \delta(o_q, o_i)| / \max\{\delta(o_q, o_j), \delta(o_q, o_i)\}\right) / k \quad (1)$$

3.5. Implementando a construção *d*HNSW e busca kN$_d$N no **ANN-Benchmark**

A construção *d*HNSW e o Algoritmo 1 foram implementados sobre a biblioteca nsmlib para facilitar a integração com o ANN-Benchmark. Nesse ambiente, os arquivos de referência para buscas exatas precisaram ser redefinidos para a avaliação de consultas kN$_d$N ao invés de kNN. Foi necessário acoplar uma busca sequencial exata da para consultas kN$_d$N e armazenar os resultados produzidos por essa rotina em arquivos HDF5 na forma de um algoritmo adicional ao ANN-Benchmark. Além disso, as implementações *d*HNSW e kN$_d$N também foram acopladas, o que permitiu instanciar a configuração do ANN-Benchmark para todas as comparações reportadas na sequência de referência.

4. Avaliação Experimental

Os experimentos foram realizados em um cluster Linux QLustar de 11 nós (01 *master* e 10 *workers*) com processadores AMD Opteron de 2.2GHz, 94GB de RAM e disco SATA de 1TB rodando o ANN-Benchmark. Quatro conjuntos de dados (FASHION-MNIST, GLOVE, MNIST, SIFT) foram escolhidos com diferentes cardinalidade ($|\mathcal{O}|$), dimensionalidade de imersão (d), dimensionalidade intrínseca (ID) e intervalos dos quartis da distribuição LID (1QT, 2QT, 3QT, 4QT), todos detalhados na Tabela 1. O HNSW e *d*HNSW foram comparados em termos de Recall e vazão (*Queries per second* – QPS), esta última calculada como a média de cinco execuções do *benchmark*.

4.1. Consultas kN$_d$N no HNSW *vs.* *d*HNSW

A Figura 6 detalha a comparação entre o HNSW e o *d*HNSW em relação à consultas kN$_d$N. Cada ponto no gráfico é comparado com o resultado exato produzido pela busca sequencial, seguindo o parâmetro M utilizado na construção do grafo. Nos experimentos, foi utilizado $M < k$, com vizinhança $k = \{10, 15, 20, 25\}$ e M variando entre $\{5, 10, 15, 20\}$. Devido a restrições de espaço, a Figura 6 mostra apenas as saídas para a configuração representativa $k = 25, M \in \{5, \ldots, 20\}$. Os resultados mostram que

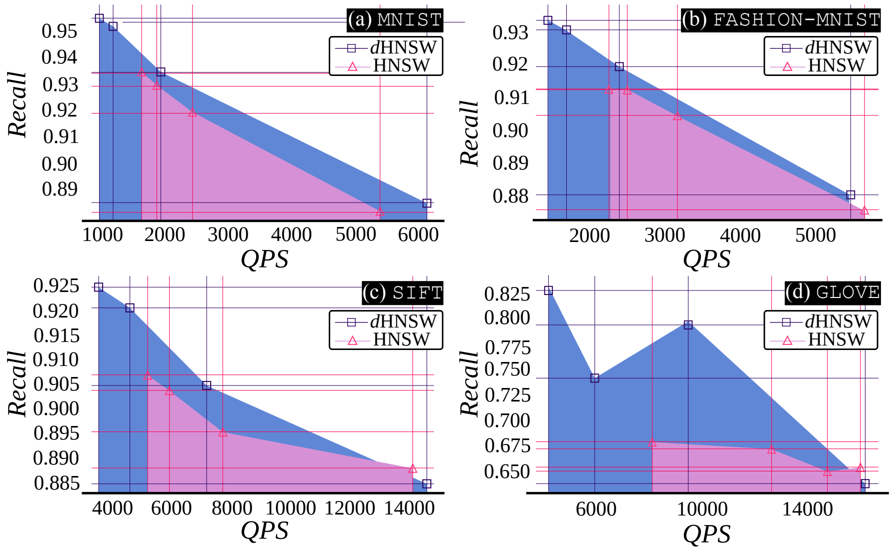

Figura 6. Resultados da comparação entre *d*HNSW e HNSW para $k = 25$.

o *d*HNSW superou consistentemente o HNSW (em até 3% de Recall) e também executou mais consultas por segundo para valores de Recall abaixo de 0.9 ($M = 5$) no conjunto de dados MNIST. As comparações no conjunto de dados SIFT mostraram resultados semelhantes, com o *d*HNSW superando o HNSW em termos de Recall (até 2%) e vazão ($M = 5$). O conjunto de dados GLOVE também apresentou essa tendência, com o *d*HNSW superando o HNSW em Recall (até 27%) e em uma vazão ($M = 5$). No caso do FASHION-MNIST, o *d*HNSW também alcançou um Recall maior do que o HNSW, sendo que a estratégia baseada em *Influência* foi mais lenta que o HNSW ($M = 5$).

Esses indícios experimentais mostram que o Recall do *d*HNSW supera consistentemente o do HNSW, embora a vazão apresente desempenhos variáveis dependendo do Recall. Para entender esse balanço, foi realizada uma segunda avaliação valendo-se da estratificação dos conjuntos por LID para identificar potenciais limitações do *d*HNSW.

4.2. Avaliação baseada em LID

Para essa avaliação cada conjunto de dados foi dividido em quatro *manifolds* de acordo com a distribuição de LID – Intervalos na Tabela 1. As Figuras 7(a–h) mostram os resultados para o primeiro e quarto *manifold* (*colunas*) de cada conjunto de dados (*linhas*). Em relação ao conjunto de dados MNIST, o HNSW foi mais rápido que o *d*HNSW no 1QT, mas essa diferença diminuiu no 4QT, com o *d*HNSW superando o concorrente quando $M = 5$. Em termos de Recall, o *d*HNSW obteve ganhos em ambos os *manifolds*, porém Recall mais altos foram observadas no 4QT. Um comportamento semelhante foi observado para o SIFT, onde o HNSW superou o *d*HNSW no 1QT e foi superado no 4QT. Em termos de Recall, o *d*HNSW obteve maiores ganhos no 4QT.

Os resultados no 1QT do FASHION-MNIST mostraram um desempenho de Recall mais próximo entre o *d*HNSW e o HNSW, com o *d*HNSW alcançando valores mais altos. O mesmo comportamento foi observado no 4QT, com uma redução na vazão tanto no *d*HNSW quanto do HNSW. Embora o HNSW tenha produzido mais consultas por segundo no 1QT, a diferença diminuiu com a LID, com ambos os métodos atingindo uma vazão semelhante para o valor de Recall mais baixo. Os resultados para o GLOVE se-

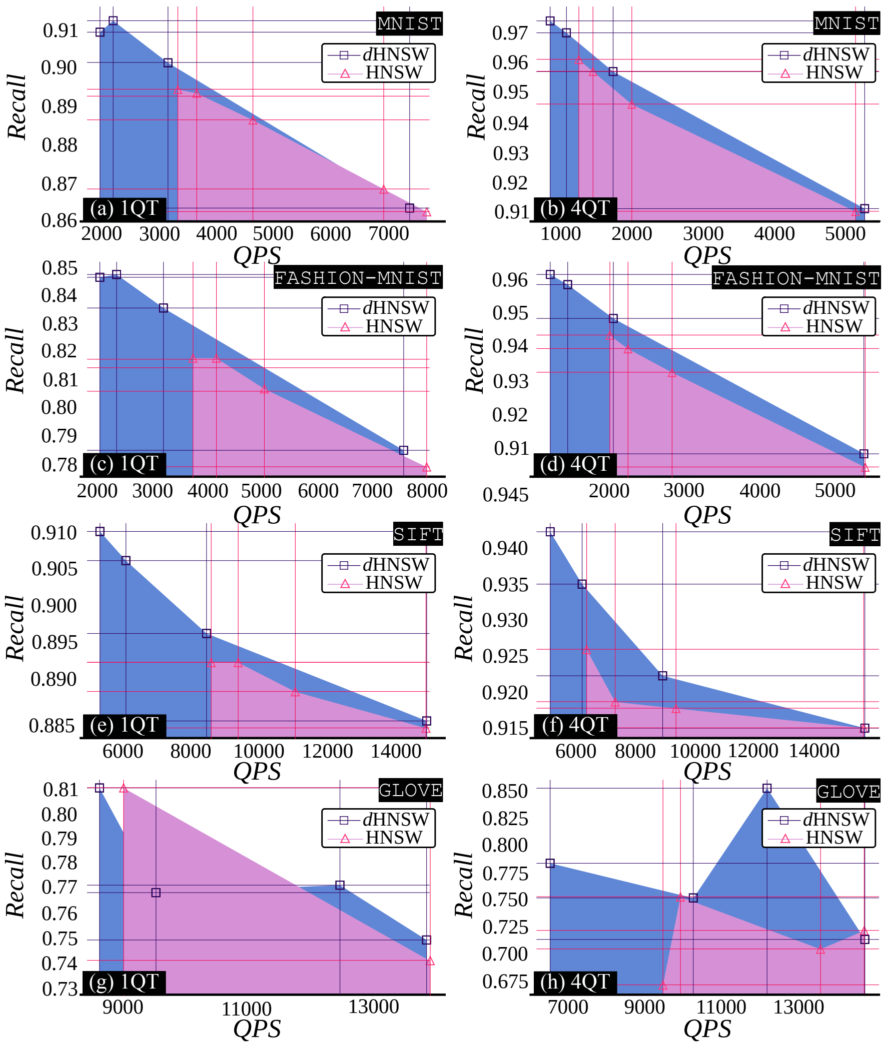

Figura 7. Comparação entre o *d*HNSW e HNSW para consultas kN$_d$N ($k = 20$) considerando *manifolds* de baixa (1QT) e alta dimensionalidade intrínseca (4QT).

guiram um padrão semelhante, porém com uma curva irregular. As diferenças de vazão também diminuíram com o LID, com o *d*HNSW superando ligeiramente o HNSW no 4QT. No geral, os resultados mostraram *(i)* que o *d*HNSW entrega um Recall mais alto que o HNSW no 1QT e no 4QT, e *(ii)* a diferença de vazão diminui com o LID, com o *d*HNSW apresentando um desempenho comparável ao HNSW.

4.3. Concentração de distâncias no *d*HNSW e HNSW

As diferenças de desempenho de qualidade e vazão guardam relação direta com a organização do espaço devido ao particionamento (arestas construídas) A Figura 8 apresenta a distribuição de distâncias, *i.e.*, arestas, para as construções *d*HNSW e HNSW em relação aos *manifolds* 1QT e 4QT dos conjuntos representativos `MNIST` e `FASHION-MNIST`, de onde se calculam as medidas de Variância Relativa (VR) e Dimensionalidade Intrínseca (ID).

Os resultados mostram que o particionamento por *Influência* produz distribuições de distâncias menos concentradas, com maior média (linhas tracejadas) e desvios padrão

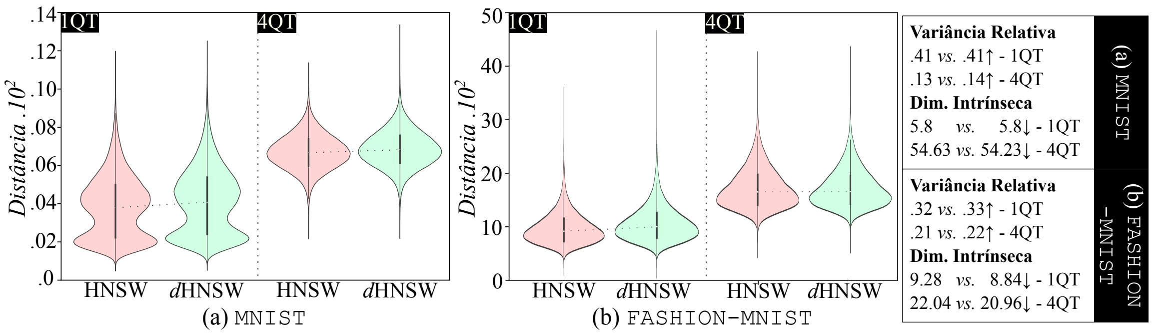

Figura 8. Distribuição de distâncias (arestas) por LID dentro do *d*HNSW e HNSW.

mais altos. Em particular, observou-se que o *d*HNSW apresentou uma VR mais alta do que o HNSW e também induziu um ID menor por *manifold*. Esses resultados fornecem indícios experimentais que sugerem que o ajuste do melhor particionamento do HNSW está relacionado à distribuição de LID dos dados, o que impacta tanto o Recall quanto a vazão de consultas por vizinhança.

5. Conclusões

Este estudo estendeu o índice HNSW como *d*HNSW por meio de uma estratégia de particionamento em bola com o conceito de *Influência*. Foi proposto um algoritmo $k\mathrm{N}_d\mathrm{N}$ e foram comparados os desempenho do *d*HNSW e HNSW sobre o `ANN-Benchmark` com e sem estratificação por LID. A abordagem proposta superou o HNSW com relação ao Recall, mostrando também uma vazão comparável em *manifolds* de alta dimensionalidade intrínseca. Além disso, a distribuição de arestas foi examinada, o que mostrou que o particionamento baseado em bola gerou estruturas menos concentradas, sugerindo um possível vínculo entre a construção do HNSW e a LID. Como trabalhos futuros, será explorada a definição de consultas por abrangência com e sem diversidade no HNSW.

Referências

Amsaleg, L., Chelly, O., Furon, T., Girard, S., Houle, M., Kawarabayashi, K.-i., and Nett, M. (2018). Extreme-value-theoretic estimation of local intrinsic dimensionality. *DMKD*, 32(6):1768–1805.

Amsaleg, L., Chelly, O., Houle, M., Kawarabayashi, K., Radovanović, M., and Treratanajaru, W. (2019). Intrinsic dimensionality estimation within tight localities. In *ICDM*.

Aumüller, M., Bernhardsson, E., and Faithfull, A. (2020). Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Info. Sys.*, 87:101374.

Aumüller, M. and Ceccarello, M. (2021). The role of local dimensionality measures in benchmarking nearest neighbor search. *Info. Sys.*, 101:101807.

Drosou, M., Jagadish, H., Pitoura, E., and Stoyanovich, J. (2017). Diversity in big data: A review. *Big Data*, 5:73–84.

He, J., Kumar, S., and Chang, S.-F. (2012). On the difficulty of nearest neighbor search. In *ICML*, pages 41–48.

Houle, M. (2013). Dimensionality, discriminability, density and distance distributions. In *ICDM*, pages 468–473. IEEE.

Jasbick, D., Dutra Santos, L., de Oliveira, D., and Bedo, M. (2020). Some branches may bear rotten fruits: Diversity browsing vp-trees. In *SISAP*, pages 140–154. Springer.

Jasbick, D., Santos, L., Azevedo-Marques, P., Traina, A., de Oliveira, D., and Bedo, M. (2023). Pushing diversity into higher dimensions: The LID effect on diversified similarity searching. *Info. Sys.*, 114:102166.

Kucuktunc, O. and Ferhatosmanoglu, H. (2013). λ-diverse nearest neighbors browsing for multidimensional data. *TKDE*, 25(3):481–493.

Li, L., Xu, J., Li, Y., and Cai, J. (2021). Hctree+: A workload-guided index for approximate knn search. *Info. Sc.*, 581:876–890.

Malkov, Y. and Yashunin, D. (2016). Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *TPAMI*, PP.

Peng, Z., Zhang, M., Li, K., Jin, R., and Ren, B. (2022). Speed-ann: Low-latency and high-accuracy nearest neighbor search via intra-query parallelism.

Santana, D. and Ribeiro, L. (2023). Approximate similarity joins over dense vector embeddings. In *SBBD*, pages 51–62. SBC.

Santos, L., Oliveira, W., Ferreira, M., Traina, A., and Traina Jr, C. (2013). Parameter-free and domain-independent similarity search with diversity. In *SSDBM*, pages 1–12.

Shimomura, L. C., Oyamada, R. S., Vieira, M. R., and Kaster, D. S. (2021). A survey on graph-based methods for similarity searches in metric spaces. *Info. Sys.*, 95:101507.

Volnyansky, I. and Pestov, V. (2009). Curse of dimensionality in pivot based indexes. In *SISAP*, pages 39–46. IEEE.

Wang, M., Xu, X., Yue, Q., and Wang, Y. (2021). A comprehensive survey and experimental comparison of graph-based approximate nn search. *PVLDB*, 14(11):1964–1978.

Xian, J., Teofili, T., Pradeep, R., and Lin, J. (2024). Vector search with OpenAI embeddings: Lucene is all you need. In *ICWSDM*, pages 1090–1093.