LEARNING SPARSE MULTIMODAL REPRESENTATIONS UNDER RETRIEVAL BUDGETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse multimodal retrievers are typically trained to preserve dense representations, ranking structure, or average retrieval cost, yet these objectives do not determine which semantic neighbors survive when inverted-index traversal is terminated under a finite budget. We formulate *Budgeted Neighborhood Preservation* (BNP), which learns sparse representations to retain the local retrieval neighborhood of a frozen dense teacher within the candidate set produced under bounded posting access. BNP combines neighborhood distillation with differentiable posting-work control and a budget-aware reachability objective that estimates whether teacher neighbors can be exposed before traversal stops. At inference, retrieval uses hard Top-k sparse codes and standard bounded inverted-index traversal, without access to the teacher neighborhood. Across five multimodal retrieval tasks with Qwen3-VL-Embedding-2B, BNP improves top-100 teacher-neighborhood coverage by 1.2–2.4 percentage points over the strongest cost-aware PUMA adaptation at a 20k-posting budget, while also improving ground-truth candidate inclusion on all five tasks. The advantage is largest under constrained budgets and narrows as traversal work increases. Final ranking gains are task-dependent, showing that preserving candidate accessibility and optimizing relevance are related but distinct objectives. These results suggest that sparse retrieval representations should be evaluated and learned according to what remains reachable under the execution budget used at inference.

1 INTRODUCTION

Large-scale multimodal retrieval increasingly relies on a two-stage pipeline: a lightweight first stage retrieves a small candidate set from a large collection, after which a stronger model reranks those candidates. Dense multimodal encoders provide strong semantic representations, but exact dense search becomes expensive as the collection grows. Sparse representations offer a complementary route because they can be served through inverted indexes, making retrieval cost depend on the posting lists activated by a query rather than on a full scan of the collection. This has motivated substantial work on learned sparse retrieval, sparse multimodal representations, and post-hoc sparsification of pretrained embedding models Paria et al. (2020); Formal et al. (2021); Chen et al. (2023); Nguyen et al. (2024); Attimonelli et al. (2026).

Most sparse retrieval objectives, however, are defined before the index is actually executed. Reconstruction losses preserve individual embeddings, ranking or distillation losses preserve dense similarity structure, and FLOPS-style objectives regulate expected retrieval cost through activation statistics Paria et al. (2020); Porco et al. (2025). These quantities are useful, but none directly specifies which semantic neighbors remain available after an inverted index has consumed a finite amount of work. This distinction matters because candidate generation is not determined by the sparse vector alone. It also depends on which posting lists are activated, how long those lists are, the order in which they are traversed, and when retrieval is terminated.

This creates a gap between preserving a dense retrieval space and preserving what is actually reachable under a serving budget. Two sparse representations may have similar support size, similar expected posting cost, or even similar unrestricted sparse rankings, yet expose different candidate sets when traversal stops after the same number of posting accesses. An item that would score highly under complete sparse evaluation can still be absent from the returned set if none of the postings that

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

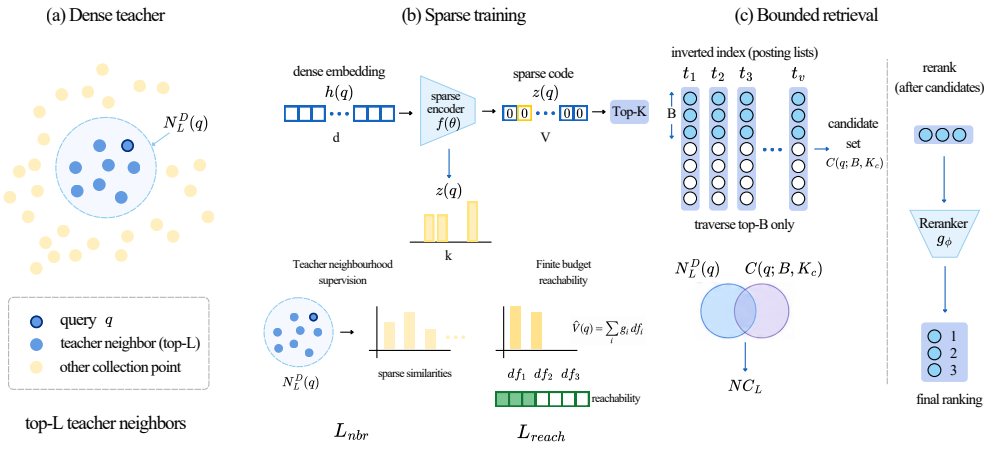


Figure 1: Overview of Budgeted Neighborhood Preservation. A frozen dense teacher defines a local retrieval neighborhood. The sparse representation is trained to preserve this structure while accounting for posting work and finite-budget reachability. At inference, a fixed traversal policy returns candidates under a hard posting budget, and neighborhood coverage measures how much of the teacher neighborhood survives before reranking.

expose it are reached before the budget is exhausted. We therefore study *Budgeted Neighborhood Preservation* (BNP): given a frozen dense teacher and a fixed inverted-index traversal budget, how should a sparse representation preserve the teacher's local retrieval neighborhood in the candidate set produced by that bounded execution?

Figure 1 summarizes this perspective. The dense teacher first defines a local neighborhood around each query. A sparse encoder then learns an indexable Top-k representation with two complementary forms of supervision. Neighborhood distillation preserves the teacher's local similarity structure, while budget-aware reachability models whether teacher neighbors can be exposed through the active posting lists before the available work is consumed. At inference, retrieval uses the hard sparse code and a fixed traversal policy; the teacher neighborhood is not available. The resulting candidate set is evaluated by how much of the teacher neighborhood survives the bounded retrieval stage, before any dense reranking is applied.

Our method implements this idea with a lightweight sparse autoencoder over frozen multimodal embeddings. In addition to reconstruction and neighborhood distillation, we estimate the posting volume induced by each query from dynamic document-frequency statistics. A straight-through support estimator provides gradients through hard Top-k selection, and a reachability objective favors sparse supports that give teacher neighbors multiple opportunities to be exposed within the training budget. This separates two quantities that are often conflated: controlling how much index work a representation tends to activate, and controlling which useful items can be reached by that work.

We evaluate BNP on five multimodal retrieval tasks using Qwen3-VL-Embedding-2B as the primary backbone, with additional experiments on CLIP and SigLIP 2. Under a common 20k-posting budget, BNP improves top-100 teacher-neighborhood coverage by 1.2–2.4 percentage points over PUMA-A+FLOPS, our strongest cost-aware adaptation of PUMA under the same sparse retrieval protocol. Ground-truth candidate inclusion also improves on all five tasks. The effect is most pronounced at constrained budgets and becomes smaller as traversal work increases. Final ranking quality does not improve uniformly: some tasks benefit from the additional candidate coverage, while others show nearly unchanged or slightly lower nDCG. This separation is useful because it shows that preserving candidate accessibility and optimizing final relevance are related but different objectives.

The paper makes three contributions. First, we formulate sparse multimodal candidate generation as neighborhood preservation under explicit posting budgets and introduce neighborhood coverage as a first-stage diagnostic distinct from sparsity, retrieval work, and final relevance. Second, we develop

a training objective that combines dense-neighborhood supervision with differentiable posting-work control and teacher-neighbor reachability under bounded traversal. Third, our experiments show that representations with similar retrieval cost can preserve different fractions of the dense neighborhood, identify reduced unvisited-neighbor loss as the main source of BNP's gains, and characterize where those gains persist across budgets, backbones, and domains.

2 Related Work

Learned sparse retrieval and retrieval cost. Learned sparse retrieval represents queries and documents with high-dimensional sparse vectors that can be served through inverted indexes. FLOPS regularization connects sparse activation patterns to expected retrieval computation, while SPLADE learns sparse lexical expansions for first-stage ranking Paria et al. (2020); Formal et al. (2021). Later work has examined the gap between vector sparsity and the work performed by an actual index. DF-FLOPS penalizes dimensions with high document frequency, which otherwise induce long posting lists, while systems such as Seismic accelerate sparse retrieval by changing how inverted lists are organized and traversed Porco et al. (2025); Bruch et al. (2024). Recent studies further revisit the role of sparsity regularization under improved retrieval engines and evaluate approximate sparse retrieval at substantially larger corpus scales Nardini et al. (2025); Bruch et al. (2025). Together, these works show that retrieval efficiency depends on both the learned representation and the execution procedure. They primarily target posting statistics, latency, or approximate-search efficiency. Our problem concerns a different consequence of bounded execution: which semantic neighbors are exposed before a fixed posting budget is exhausted.

Sparse multimodal representations. Sparse retrieval has also been extended beyond text. STAIR learns grounded sparse image and text representations in a shared token space, while multimodal learned sparse retrieval has studied the conversion of frozen dense representations into sparse vectors and explicitly models dimension co-activation and semantic deviation Chen et al. (2023); Nguyen et al. (2024). Sparse coding provides another route to efficient representations. CSR maps pretrained embeddings into a selectively activated feature space using lightweight autoencoding and task-aware contrastive training, with evaluations across image, text, and multimodal tasks Wen et al. (2025). Recent work has further explored sparsity during multimodal representation learning, alignment of sparse dictionaries across modalities, and inference-free sparse retrieval for visual documents Qin et al. (2026); Kaushik et al. (2026); Cho et al. (2026). These methods address semantic preservation, modality alignment, lexical grounding, and retrieval efficiency after sparsification. We use a Top-k sparse encoder only as the representation parameterization; our learning objective is defined by preserving dense-teacher neighborhoods under bounded candidate generation.

Post-hoc sparsification and budgeted candidate reachability. The closest line of work learns sparse retrieval representations from pretrained dense embeddings. SAE latents have been used directly as indexing units for learned sparse retrieval, showing that sparse latent spaces can support effective retrieval without vocabulary-constrained dimensions Formal et al. (2026). PUMA applies post-hoc sparsification to universal multimodal embeddings using Top-k sparse autoencoding, dense-geometry preservation, support alignment, and retrieval-aware fine-tuning Attimonelli et al. (2026). It also identifies insufficient pre-TopK support and retrieval-misaligned active support as distinct failure modes, making it a strong reference for whether dense retrieval structure survives sparsification. BNP studies the additional failure introduced by finite index execution. Given a fixed traversal policy and posting budget, we model whether teacher neighbors can be exposed before traversal terminates and optimize this reachability directly. This distinguishes budgeted neighborhood preservation from objectives that preserve unrestricted sparse rankings or regulate average retrieval cost, and lets candidate utility be evaluated before dense reranking.

3 Budgeted Neighborhood Preservation

Let f denote a frozen multimodal embedding model. For any query or collection item x, we use its normalized dense representation

$$h(x) = \frac{f(x)}{\|f(x)\|_2} \in \mathbb{R}^{d_h}, \tag{1}$$

where d_h is the embedding dimension. Given a query q and a collection $\mathcal{D}$, the dense teacher scores each item by $s_D(q, x) = h(q)^\top h(x)$. We denote its top-$L$ retrieval neighborhood as

$$\mathcal{N}_L^D(q) = \mathrm{TopL}_{x \in \mathcal{D}}\ s_D(q, x). \tag{2}$$

This neighborhood provides a local reference for the retrieval structure encoded by the dense model. It is used for training and evaluation, but is unavailable to the sparse retriever during candidate generation.

We seek a sparse representation $z_\theta(x) \in \mathbb{R}^m$ with at most k active dimensions. Candidate generation is performed by an inverted index under a fixed traversal policy π. A posting budget B limits the total number of posting entries that may be accessed across the active query dimensions, while K_c limits the number of returned candidates. We write the resulting set as $\mathcal{C}_{\theta,\pi}(q; B, K_c)$ and denote the realized number of posting accesses by $V_{\theta,\pi}(q) \leq B$. The distinction between the budget cap and realized work is important because some queries may terminate before exhausting the available budget. Since candidate generation depends jointly on the representation and the traversal procedure, we hold π fixed in the primary representation comparisons.

We measure how much of the dense neighborhood survives bounded retrieval using *neighborhood coverage*,

$$\mathrm{NC}_L(q; B) = \frac{\left|\mathcal{N}_L^D(q) \cap \mathcal{C}_{\theta,\pi}(q; B, K_c)\right|}{L}. \tag{3}$$

Dataset-level coverage averages this quantity over queries. NC is distinct from sparsity and retrieval work: representations with similar support sizes and similar posting costs may expose different candidate sets. It is also distinct from unrestricted sparse retrieval quality. An item can receive a high score when all sparse dimensions are evaluated yet remain absent from the finite-budget candidate set because the postings that expose it are not reached before traversal stops. Finally, teacher-neighborhood preservation is not itself a relevance metric. We therefore report NC together with ground-truth candidate inclusion and the task-specific retrieval metric of each benchmark.

The resulting learning problem is to preserve the teacher neighborhood under bounded candidate-generation work,

$$\max_\theta\ \mathbb{E}_q\left[\mathrm{NC}_L(q; B)\right]. \tag{4}$$

Direct optimization is difficult because sparse support selection, posting traversal, score accumulation, and hard budget termination jointly determine the returned set. We therefore optimize differentiable surrogates for neighborhood preservation and budgeted candidate reachability, while the hard posting cap remains part of the retrieval procedure at inference.

4 Learning Sparse Representations for Budgeted Neighborhood Preservation

We learn a sparse transformation of the frozen embedding space introduced in Sec. 3. The representation is trained to preserve the local retrieval structure of the dense teacher while favoring sparse supports that expose useful neighbors within limited index work. We separate these two requirements into neighborhood supervision and budget-aware reachability. The resulting representation is trained with differentiable surrogates, while inference uses the same hard sparse code and bounded inverted-index traversal used for evaluation.

4.1 Sparse Representation

Given a normalized embedding $h(x) \in \mathbb{R}^{d_h}$, a lightweight encoder produces non-negative activations

$$u_\theta(x) = \mathrm{ReLU}(W_e h(x) + b_e), \tag{5}$$

where $W_e \in \mathbb{R}^{m \times d_h}$ and $b_e \in \mathbb{R}^m$. We retain the k largest activations,

$$z_\theta(x) = \mathrm{TopK}(u_\theta(x), k), \qquad \|z_\theta(x)\|_0 \leq k, \tag{6}$$

and use the resulting vector directly for indexing and sparse scoring.

A linear decoder maps the sparse code back to the dense embedding space,

$$\hat{h}(x) = W_d z_\theta(x) + b_d, \tag{7}$$

with reconstruction loss

$$\mathcal{L}_{\text{rec}} = \|h(x) - \hat{h}(x)\|_2^2. \tag{8}$$

Reconstruction prevents the sparse transformation from discarding the dense representation indiscriminately, but does not determine which neighbors remain accessible after finite index traversal.

4.2 Neighborhood Preservation

Candidate generation depends on the relative geometry between a query and nearby collection items rather than on pointwise reconstruction alone. For each training query q, let $\mathcal{S}(q)$ contain dense-teacher neighbors together with sampled negatives. The teacher defines

$$p_D(x \mid q) = \frac{\exp(s_D(q,x)/\tau_D)}{\sum_{x' \in \mathcal{S}(q)} \exp(s_D(q,x')/\tau_D)}, \tag{9}$$

where τ_D is a temperature. Sparse similarities are computed by

$$s_S(q,x) = z_\theta(q)^\top z_\theta(x), \tag{10}$$

and converted into $p_S(x \mid q)$ with temperature τ_S. We minimize

$$\mathcal{L}_{\text{nbr}} = D_{\text{KL}}(p_D(\cdot \mid q) \| p_S(\cdot \mid q)). \tag{11}$$

This objective transfers the local ranking structure of the dense teacher to the sparse scoring space. It does not account for whether the corresponding sparse coordinates can be traversed within a finite posting budget. We address this separately through a differentiable approximation to retrieval work and teacher-neighbor exposure.

4.3 Budget-Aware Reachability

The hard Top-k support used by the sparse representation is discrete. During training, we associate it with a soft support signal

$$g_i(x) = \sigma\left(\frac{u_{\theta,i}(x) - t_k(x)}{\tau_g}\right), \tag{12}$$

where $t_k(x)$ is the kth-largest activation and τ_g controls the softness of the boundary. Let

$$g_i^{\text{hard}}(x) = \mathbb{1}[i \in \text{TopK}(u_\theta(x), k)]. \tag{13}$$

We use the straight-through gate

$$\bar{g}_i(x) = g_i^{\text{hard}}(x) + g_i(x) - \text{sg}(g_i(x)), \tag{14}$$

where sg denotes stop-gradient. Its forward value is the same hard support used at inference, while its backward derivative follows the soft gate.

Retrieval work also depends on how frequently each sparse dimension occurs in the collection. Let df_i denote the number of indexed items activating dimension i. Since the representation changes during training, we maintain an exponential moving estimate

$$\widehat{\text{df}}_i^{(t)} = \mu \widehat{\text{df}}_i^{(t-1)} + (1-\mu)\widetilde{\text{df}}_i^{(t)}, \tag{15}$$

where $\widetilde{\text{df}}_i^{(t)}$ is the document-frequency estimate from the current collection minibatch. These statistics are used as retrieval-cost estimates and are not differentiated through.

For a query q, the expected volume of its active posting lists is approximated by

$$\widehat{V}_\theta(q) = \sum_{i=1}^{m} \bar{g}_i(q)\, \widehat{\text{df}}_i. \tag{16}$$

We constrain its query average to a training target B_{train}. With a non-negative dual variable λ, this contributes

$$\mathcal{L}_{\text{cost}} = \lambda \left(\mathbb{E}_q[\widehat{V}_\theta(q)] - B_{\text{train}} \right), \tag{17}$$

with projected ascent

$$\lambda \leftarrow \left[\lambda + \eta_\lambda \left(\mathbb{E}_q[\widehat{V}_\theta(q)] - B_{\text{train}} \right) \right]_+ . \tag{18}$$

This term controls average posting volume, but does not specify which teacher neighbors should be exposed by that work.

To model finite-budget exposure, we estimate the fraction of the active posting volume that can be traversed as

$$\rho_\theta(q) = \min \left(1, \frac{B_{\text{train}}}{\widehat{V}_\theta(q) + \epsilon} \right). \tag{19}$$

For a teacher neighbor x, each shared sparse coordinate provides an opportunity for the item to be exposed during traversal. We combine these opportunities into the differentiable reachability score

$$r_\theta(q, x; B_{\text{train}}) = 1 - \prod_{i=1}^{m} \left(1 - \rho_\theta(q) \bar{g}_i(q) \bar{g}_i(x) \right). \tag{20}$$

The score increases when a teacher neighbor shares more active dimensions with the query and decreases when the query activates posting volume that substantially exceeds the available budget.

We weight teacher neighbors according to their dense similarity,

$$w_D(x \mid q) = \frac{\exp(s_D(q, x)/\tau_R)}{\sum_{x' \in \mathcal{N}_L^D(q)} \exp(s_D(q, x')/\tau_R)}, \tag{21}$$

and minimize

$$\mathcal{L}_{\text{reach}} = - \sum_{x \in \mathcal{N}_L^D(q)} w_D(x \mid q) \log \left(r_\theta(q, x; B_{\text{train}}) + \epsilon \right). \tag{22}$$

Unlike neighborhood distillation, which preserves similarities under complete sparse scoring, this term favors sparse supports through which teacher neighbors can be exposed before the traversal budget is exhausted.

The complete training objective is

$$\mathcal{L}_{\text{BNP}} = \mathcal{L}_{\text{nbr}} + \beta \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cost}} + \gamma \mathcal{L}_{\text{reach}}, \tag{23}$$

where β and γ control reconstruction and budget-aware reachability. The training budget is a surrogate for expected retrieval work; it does not replace the hard per-query budget used by the index.

4.4 Bounded Candidate Retrieval

Each collection item contributes one weighted entry to the posting list of every active sparse dimension,

$$\mathcal{P}(i) = \{ (x, z_{\theta,i}(x)) : z_{\theta,i}(x) > 0 \}. \tag{24}$$

For a query q, the retriever traverses the active posting lists according to the fixed policy π. Sparse scores are accumulated from shared active dimensions,

$$s_S(q, x) = \sum_{i \in \mathcal{K}_q \cap \mathcal{K}_x} z_{\theta,i}(q) z_{\theta,i}(x), \tag{25}$$

where $\mathcal{K}_x = \text{supp}(z_\theta(x))$.

Traversal terminates after at most B posting accesses, and the highest-scoring K_c encountered items form $\mathcal{C}_{\theta,\pi}(q; B, K_c)$. Some queries terminate earlier, so we record the realized posting count $V_{\theta,\pi}(q)$ together with the cap B.

For two-stage evaluation, dense similarity is recomputed only for the returned candidates. Reranking may change their final order, but cannot recover an item that was not exposed by the sparse stage. Neighborhood coverage is therefore measured on the candidate set before dense reranking.

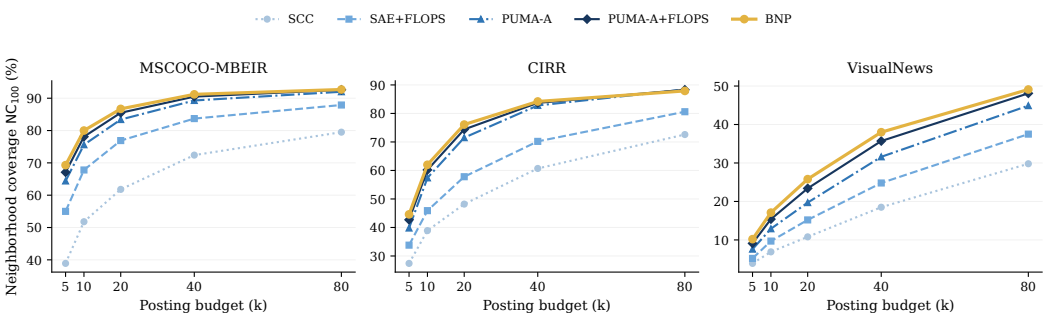


Figure 2: NC_{100} across posting budgets. BNP is strongest in the constrained-budget regime and generally converges toward the strongest baselines as more postings are available.

5 Experimental Setup

Datasets and backbones. We evaluate on five multimodal retrieval tasks: MSCOCO-MBEIR, CIRR, FashionIQ, VisualNews, and Fashion200K. They cover text-to-image retrieval as well as composed image retrieval with an image–text query. Our primary backbone is the frozen Qwen3-VL-Embedding-2B encoder, whose 2,048-dimensional embeddings define both the dense teacher and the input to the sparse transformation. We additionally test CLIP ViT-B/32, CLIP ViT-L/14, and SigLIP 2 B/16@256 to examine whether the behavior transfers across embedding families.

Baselines. We compare BNP with the original sparse autoencoding pipeline (SCC), an SAE trained with FLOPS regularization, and PUMA, the closest post-hoc multimodal sparsification method. Because our evaluation imposes a common sparse support and bounded traversal protocol, we reproduce PUMA under the same setting and denote this variant as PUMA-A. We also include PUMA-A+FLOPS as a stronger cost-aware baseline. Exact dense retrieval with the frozen backbone is reported as a reference but is not subject to the sparse posting budget.

Retrieval protocol. Unless stated otherwise, sparse representations use a dictionary size of $m = 32{,}768$ with at most $k = 128$ active dimensions. We set the teacher neighborhood size to $L = 100$, the candidate capacity to $K_c = 1{,}000$, and the posting-budget cap to $B = 20{,}000$. All sparse methods use the same inverted-index traversal policy π and candidate capacity. We additionally evaluate budgets of 5k, 10k, 40k, and 80k to characterize the coverage–work trade-off. Alongside the budget cap, we record the realized number of posting accesses for each method and query.

Evaluation. Our primary first-stage metric is NC_{100}, the fraction of the dense teacher's top-100 neighbors retained in the sparse candidate set. Candidate Hit@1k measures whether at least one labeled relevant item is present before reranking. Final retrieval quality is measured with the benchmark-specific ranking metrics, including nDCG@10, H@5, and R@10. For two-stage retrieval, we rerank at most 1,000 returned candidates using the same frozen dense similarity. Reranking cannot add candidates and therefore does not affect neighborhood coverage.

6 Results

6.1 Neighborhood Preservation under Bounded Retrieval

Table 1 shows that BNP achieves the highest NC_{100} on all five tasks. Relative to PUMA-A+FLOPS, coverage improves by 1.2–2.4 points and Hit@1k also increases on every dataset, while realized posting work remains similar or slightly lower. Final ranking behaves differently: BNP improves nDCG on MSCOCO-MBEIR and VisualNews but is 0.1–0.3 points below PUMA-A+FLOPS on the other three tasks. Thus, improved candidate accessibility does not necessarily translate into better final ranking.

Table 1: Matched-budget retrieval with Qwen3-VL-Embedding-2B at $B = 20$k. NC and Hit are measured before reranking; nDCG is measured after dense reranking. Post. denotes mean realized posting accesses (k).

Dataset	Method	NC_{100} ↑	Hit@1k ↑	nDCG@10 ↑	Post. ↓
MSCOCO-MBEIR	SCC	61.8	91.1	58.4	17.64
	SAE+FLOPS	76.9	95.2	65.5	16.98
	PUMA-A	83.4	97.0	69.0	18.42
	PUMA-A+FLOPS	85.5	97.2	69.6	17.21
	BNP	**86.7**	**97.4**	**69.8**	17.06
CIRR	SCC	48.2	72.9	31.6	19.76
	SAE+FLOPS	57.8	78.1	34.8	19.41
	PUMA-A	71.5	86.6	40.3	19.87
	PUMA-A+FLOPS	74.4	87.8	**40.9**	19.56
	BNP	**76.1**	**88.4**	40.7	19.48
FashionIQ	SCC	29.4	55.8	10.4	19.73
	SAE+FLOPS	36.9	61.5	11.9	19.35
	PUMA-A	51.7	69.1	15.3	19.82
	PUMA-A+FLOPS	54.2	70.4	**15.4**	19.47
	BNP	**55.4**	**70.7**	15.1	19.39
VisualNews	SCC	10.8	31.5	7.6	19.99
	SAE+FLOPS	15.2	35.7	9.8	19.73
	PUMA-A	19.7	43.9	12.9	20.00
	PUMA-A+FLOPS	23.4	46.7	14.0	19.82
	BNP	**25.8**	**48.1**	**14.7**	19.78
Fashion200K	SCC	14.8	25.6	4.5	19.97
	SAE+FLOPS	21.3	30.8	5.7	19.60
	PUMA-A	27.6	36.5	6.8	19.99
	PUMA-A+FLOPS	31.5	39.1	**7.3**	19.76
	BNP	**33.0**	**40.2**	7.2	19.70

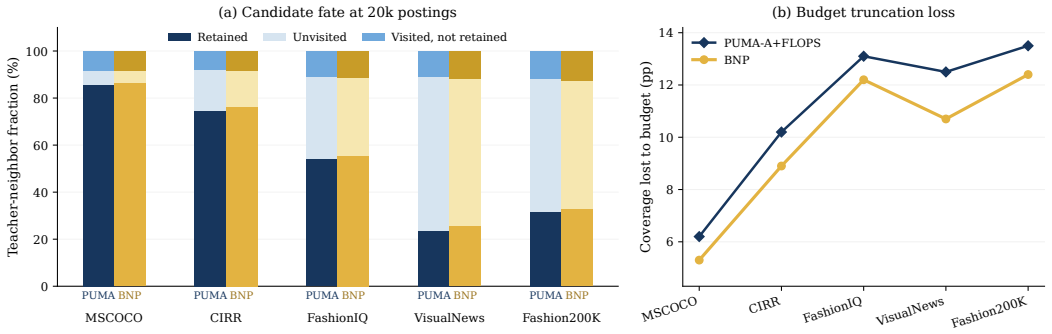


Figure 3: Teacher-neighbor loss under bounded retrieval. **Left:** Candidate fate at a 20k posting cap. **Right:** Coverage lost relative to unrestricted sparse scoring. BNP mainly reduces neighbors that are never visited.

Figure 2 confirms that the gain is budget-dependent. Against PUMA-A+FLOPS, BNP improves coverage by 2.2, 1.9, and 1.1 points at 5k postings on MSCOCO-MBEIR, CIRR, and VisualNews, respectively. The gaps narrow as traversal increases; on CIRR the two methods are effectively tied at 80k. VisualNews remains more sensitive to bounded execution, where BNP retains a measurable advantage even at larger budgets. These trends place the main benefit of BNP in the finite-traversal regime for which it is designed.

6.2 Where Does the Improvement Come From?

We separate teacher neighbors that are retained, never visited before the budget is exhausted, or visited but excluded from the final top-K_c candidate set.

Figure 3 localizes the improvement to finite traversal. On CIRR, BNP reduces the unvisited fraction from 17.8% to 15.7%, while visited-but-not-retained changes only from 7.8% to 8.2%. On Visu-

Table 2: Objective ablation at $B = 20$k. Post. denotes mean realized posting accesses (k).

	CIRR				VisualNews			
Method	NC ↑	Hit ↑	nDCG ↑	Post. ↓	NC ↑	Hit ↑	nDCG ↑	Post. ↓
Vanilla SAE	48.2	72.9	31.6	19.76	10.8	31.5	7.6	19.99
Neighborhood only	68.0	84.0	38.9	19.92	17.6	40.5	11.8	20.00
Budget only	54.6	77.1	33.1	18.68	13.7	35.1	8.8	19.41
Neighborhood + FLOPS	70.3	85.7	39.6	19.43	21.4	45.0	13.4	19.83
Mean-cost constraint	72.0	86.9	40.0	19.32	23.1	46.5	13.9	19.76
PUMA-A+FLOPS	74.4	87.8	**40.9**	19.56	23.4	46.7	14.0	19.82
BNP	**76.1**	**88.4**	40.7	19.48	**25.8**	**48.1**	**14.7**	19.78

alNews, unvisited neighbors fall from 65.7% to 62.5%. The unrestricted sparse coverage changes by only 0.4 points on CIRR and 0.6 on VisualNews, whereas the bounded-coverage gains are 1.7 and 2.4 points. BNP therefore changes which teacher neighbors become reachable under the budget more than it changes unrestricted sparse ranking.

Table 2 shows that neighborhood supervision provides the largest gain over the reconstruction-only SAE, whereas budget control alone is insufficient. FLOPS and mean-cost control improve coverage further, but adding $\mathcal{L}_{\text{reach}}$ raises NC_{100} from 72.0 to 76.1 on CIRR and from 23.1 to 25.8 on VisualNews at similar realized work. Together with Figure 3, this supports the intended role of reachability supervision: improving teacher-neighbor exposure under bounded traversal rather than directly optimizing final ranking.

6.3 Generalization across Backbones and Domains

The same behavior extends beyond Qwen3-VL-Embedding-2B. When retrained for CLIP ViT-B/32, CLIP ViT-L/14, and SigLIP 2, BNP improves NC_{100} over PUMA-A by 2.7–4.6 points in both retrieval directions, while R@10 changes by only 0.4–0.6 points. Frozen transfer is less consistent: a COCO-trained mapping improves coverage on CIRR (+3.6) and FashionIQ (+1.5), but decreases it on VisualNews (-0.7). Full results are provided in Appendix F.

7 Discussion and Conclusion

BNP treats sparse retrieval quality as an execution-dependent property. A representation that preserves dense similarities under complete scoring may still lose useful neighbors when only part of the inverted index can be traversed. Our results show that this distinction matters most under constrained budgets and becomes less pronounced as retrieval approaches exhaustive sparse evaluation. Neighborhood coverage should therefore be interpreted together with the traversal policy and posting budget under which candidates are produced, rather than as an intrinsic property of the sparse representation alone.

Higher teacher-neighborhood coverage does not imply uniformly better final relevance. BNP consistently improves candidate accessibility, while downstream ranking gains depend on how well the dense teacher neighborhood aligns with benchmark relevance and on the target distribution. Frozen cross-dataset transfer further shows that the learned sparse support is not distribution-independent. BNP is therefore best viewed as a candidate-generation objective for preserving useful semantic structure under bounded execution, rather than as a replacement for task-specific ranking objectives.

We introduced Budgeted Neighborhood Preservation as a formulation for learning sparse multimodal representations around the candidates that remain reachable under finite inverted-index work. BNP combines dense-neighborhood supervision, posting-work control, and differentiable reachability while retaining standard hard sparse retrieval at inference. Across multiple retrieval tasks and embedding backbones, its clearest gains occur in the constrained-budget regime. These results suggest that sparse retrieval should be evaluated not only by representation quality or average cost, but also by what semantic structure survives the execution budget used to generate candidates.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work as supporting research and writing tools. They assisted with language polishing and rewriting, literature search and organization, research discussion and early-stage experimental planning, and the drafting and restructuring of portions of the manuscript, including descriptions of the method, experiments, figures, and tables. AI tools were also used to explore possible experimental outcomes during planning, but were not used to generate or replace the experimental measurements reported in this paper. All methodological decisions, implementations, experiments, reported numerical results, citations, and scientific claims were independently checked and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We describe the sparse representation, neighborhood supervision, straight-through support estimator, document-frequency updates, budget-aware reachability objective, and bounded retrieval procedure in the main paper. The appendix provides dataset and backbone settings, baseline definitions, retrieval and reranking protocols, traversal policies, hyperparameter sensitivity, additional ablations, complete budget curves, candidate-loss analysis, and system-level measurements. These details specify the configurations required to reproduce the main experiments. Code, configuration files, and evaluation scripts will be released with the supplementary material where permitted by the submission policy.

REFERENCES

Matteo Attimonelli, Alessandro De Bellis, Franco Maria Nardini, Claudio Pomo, Cosimo Rulli, Rossano Venturini, and Tommaso Di Noia. Puma: Post-hoc sparsification of universal multimodal embeddings for efficient retrieval, 2026. URL https://arxiv.org/abs/2608.25780.

Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. Efficient inverted indexes for approximate retrieval over learned sparse representations. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14–18, 2024*, pp. 152–162. ACM, 2024. doi: 10.1145/3626772.3657769. URL https://doi.org/10.1145/3626772.3657769.

Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, Rossano Venturini, and Leonardo Venuta. Investigating the scalability of approximate sparse retrieval algorithms to massive datasets. In *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*, pp. 437–445. Springer, 2025. doi: 10.1007/978-3-031-88714-7_43. URL https://doi.org/10.1007/978-3-031-88714-7_43.

Chen Chen, Bowen Zhang, Liangliang Cao, Jiguang Shen, Tom Gunter, Albin Madappally Jose, Alexander Toshev, Yantao Zheng, Jonathon Shlens, Ruoming Pang, and Yinfei Yang. STAIR: Learning sparse text and image representation in grounded tokens. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15079–15094, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.932. URL https://aclanthology.org/2023.emnlp-main.932/.

Gyu-Hwung Cho, Youngjune Lee, Kiyoon Jeong, Siyoung Lee, Sanggyu Han, Hervé Dejean, Stéphane Clinchant, and Seung-won Hwang. Inference-free multimodal learned sparse retrieval for production-scale visual document search. *CoRR*, abs/2605.30917, 2026. doi: 10.48550/ARXIV.2605.30917. URL https://doi.org/10.48550/arXiv.2605.30917.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (eds.), *SIGIR '21: The 44th International ACM SIGIR*

Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11–15, 2021, pp. 2288–2292. ACM, 2021. doi: 10.1145/3404835.3463098. URL https://doi.org/10.1145/3404835.3463098.

Thibault Formal, Maxime Louis, Hervé Déjean, and Stéphane Clinchant. Learning retrieval models with sparse autoencoders. *CoRR*, abs/2603.13277, 2026. doi: 10.48550/ARXIV.2603.13277. URL https://doi.org/10.48550/arXiv.2603.13277.

Chiraag Kaushik, Davis Barch, and Andrea Fanelli. Decomposing multimodal embedding spaces with group-sparse autoencoders. *CoRR*, abs/2601.20028, 2026. doi: 10.48550/ARXIV.2601.20028. URL https://doi.org/10.48550/arXiv.2601.20028.

Franco Maria Nardini, Thong Nguyen, Cosimo Rulli, Rossano Venturini, and Andrew Yates. Effective inference-free retrieval for learned sparse representations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13–18, 2025*, pp. 2936–2940. ACM, 2025.

Thong Nguyen, Mariya Hendriksen, Andrew Yates, and Maarten de Rijke. Multimodal learned sparse retrieval with probabilistic expansion control. In Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (eds.), *Advances in Information Retrieval – 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*, volume 14609 of *Lecture Notes in Computer Science*, pp. 448–464. Springer, 2024. doi: 10.1007/978-3-031-56060-6_29. URL https://doi.org/10.1007/978-3-031-56060-6_29.

Biswajit Paria, Chih-Kuan Yeh, Ian En-Hsu Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. Minimizing flops to learn efficient sparse representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SygpC6Ntvr.

Aldo Porco, Dhruv Mehra, Igor Malioutov, Karthik Radhakrishnan, Moniba Keymanesh, Daniel Preotiuc-Pietro, Sean MacAvaney, and Pengxiang Cheng. An alternative to flops regularization to effectively productionize splade-doc. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13–18, 2025*, pp. 2789–2793. ACM, 2025. doi: 10.1145/3726302.3730163. URL https://doi.org/10.1145/3726302.3730163.

Chuan Qin, Constantin Venhoff, Sonia Joseph, Fanyi Xiao, and Stefan Scherer. Sparse clip: Co-optimizing interpretability and performance in contrastive learning. *CoRR*, abs/2601.20075, 2026. doi: 10.48550/ARXIV.2601.20075. URL https://doi.org/10.48550/arXiv.2601.20075.

Tiansheng Wen, Yifei Wang, Zequn Zeng, Zhong Peng, Yudi Su, Xinyang Liu, Bo Chen, Hongwei Liu, Stefanie Jegelka, and Chenyu You. Beyond matryoshka: Revisiting sparse coding for adaptive representation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66520–66538. PMLR, 2025. URL https://proceedings.mlr.press/v267/wen25e.html.

APPENDIX

A ADDITIONAL IMPLEMENTATION AND EVALUATION DETAILS

A.1 DATASETS AND BACKBONES

Table A1 summarizes the evaluation collections. The main experiments use Qwen3-VL-Embedding-2B with a 2,048-dimensional frozen embedding space. Cross-backbone experiments additionally use CLIP ViT-B/32, CLIP ViT-L/14, and SigLIP 2 B/16@256. Each backbone has its own sparse transformation; the dense encoder itself remains frozen. For composed-image retrieval datasets, the dense teacher follows the native multimodal query construction of the corresponding embedding model.

Table A1: Evaluation datasets used in the main experiments.

Dataset	Query type	Test queries	Collection size	Domain
CIRR	image+text → image	4,170	21,552	misc
FashionIQ	image+text → image	6,003	74,381	fashion
VisualNews	text → image	19,996	542,246	news
Fashion200K	text → image	1,746	201,824	fashion
MSCOCO-MBEIR	text → image	24,819	5,000	misc

The five tasks span markedly different collection sizes and query constructions. MSCOCO-MBEIR is the smallest collection in this group, with 5,000 candidates, whereas VisualNews contains more than 542k images and Fashion200K contains more than 200k. CIRR and FashionIQ additionally require composed image–text queries rather than a single text query. This variation is useful for the present study because a fixed posting budget covers very different fractions of the available collection across tasks. In particular, failures caused by bounded traversal are expected to be more visible on the larger collections, where long posting lists can consume the available work before all useful candidates are exposed.

A.2 SPARSE TRAINING CONFIGURATION

Unless otherwise stated, the sparse dictionary has $m = 32{,}768$ dimensions and the hard code keeps at most $k = 128$ non-zero entries. The dense-teacher neighborhood uses $L = 100$, the default posting cap is $B = 20{,}000$, the candidate capacity is $K_c = 1{,}000$, and dense reranking is restricted to the returned candidate set. The document-frequency statistics used by the reachability objective are updated with an exponential moving average with $\mu = 0.99$. The teacher neighborhood is used only for training and evaluation and is never queried during candidate generation.

Table A2: Default configuration for the primary experiments.

Setting	Value
Backbone	Qwen3-VL-Embedding-2B
Dense dimension d_h	2,048
Dictionary size m	32,768
Active dimensions k	128
Teacher neighborhood L	100
Posting cap B	20,000
Candidate capacity K_c	1,000
Rerank depth	1,000
DF EMA μ	0.99
Traversal policy	P_0

We keep this configuration fixed in the primary comparisons so that the candidate-generation budget is the principal changing factor. The choice $m = 32{,}768$, $k = 128$, and $L = 100$ is examined separately in Sec. G. The 1,000-candidate capacity is deliberately larger than the final evaluation cutoff, allowing the first stage to return a reasonably broad set without giving the reranker access

Table A3: Public Qwen3-VL-Embedding-2B reference scores reported by PUMA. These values use the original PUMA protocol and are not directly comparable to our bounded-traversal results.

Dataset	Method	H@5	R@10	nDCG@10
CIRR	Dense	52.21	62.81	42.35
CIRR	PUMA	53.29	64.49	43.56
FashionIQ	Dense	20.29	26.80	17.06
FashionIQ	PUMA	20.62	26.97	16.81
VisualNews	Dense	26.66	33.18	22.56
VisualNews	PUMA	25.90	32.43	21.68
Fashion200K	Dense	11.98	8.79	7.92
Fashion200K	PUMA	17.63	13.55	11.44
MSCOCO-MBEIR	Dense	80.49	87.21	71.81
MSCOCO-MBEIR	PUMA	80.98	88.05	71.91

to any item that was not found by sparse traversal. The distinction between $B = 20{,}000$ and $K_c = 1{,}000$ is important: B controls index work, whereas K_c limits the number of scored items retained after that work.

A.3 Baselines

SCC is the reconstruction-based sparse autoencoding pipeline without retrieval-aware supervision. SAE+FLOPS augments the same sparse parameterization with FLOPS regularization. PUMA-A is our adaptation of PUMA Attimonelli et al. (2026) to the common dictionary size, Top-k support, candidate capacity, and bounded traversal protocol used by BNP. PUMA-A+FLOPS adds a cost-aware FLOPS term to this adaptation. We use "-A" to distinguish these controlled reproductions from scores reported by the original PUMA paper, whose sparsity and retrieval protocol differ from ours. Exact dense retrieval is reported only as an unrestricted reference and is not assigned a posting budget.

The public PUMA values are included only to anchor the strength and behavior of the underlying Qwen3-VL-Embedding-2B representation. They also show why the controlled PUMA-A reproduction is necessary. Under the original protocol, PUMA can match or slightly exceed the dense reference on some tasks, such as CIRR and Fashion200K, while showing small degradations on others, such as FashionIQ and VisualNews. Our main experiments change the serving problem by imposing a common Top-k representation and hard posting cap, so directly mixing these public numbers with the bounded-traversal results would conflate representation quality with retrieval protocol.

A.4 Traversal and Metrics

The default policy P_0 performs balanced round-robin traversal over the posting lists activated by the query; postings within a list are ordered by their stored sparse weight. Each accessed item–weight pair counts as one posting visit, including repeated accesses to the same item through different dimensions. Scores are accumulated online and the top K_c encountered items are retained. The alternative policy P_1 prioritizes query dimensions by their query-side activation before traversing their posting lists. Representation comparisons in the main paper always hold the traversal policy fixed.

Neighborhood coverage NC_L is measured before reranking. Candidate Hit@1k records whether the candidate set contains at least one labeled relevant item. H@5, R@10, and nDCG@10 are computed after dense reranking. For mechanism analysis, every teacher neighbor is assigned to one of three mutually exclusive outcomes: retained, unvisited before the budget terminates, or visited but not retained among the final candidates. "Full sparse NC" evaluates the same sparse representation without the posting cap and therefore isolates the coverage loss introduced by bounded execution.

Table A4: Complete results at a 20k posting cap. Postings are reported in thousands.

Dataset	Method	NC_{100}	Hit@1k	H@5	R@10	nDCG@10	Mean Post.	P50 Post.	P90 Post.
MSCOCO-MBEIR	SCC	61.8	91.1	66.2	73.9	58.4	17.64	17.82	20.00
	SAE+FLOPS	76.9	95.2	74.0	81.6	65.5	16.98	17.06	19.85
	PUMA-A	83.4	97.0	77.9	84.9	69.0	18.42	18.71	20.00
	PUMA-A+FLOPS	85.5	97.2	78.6	85.6	69.6	17.21	17.33	19.91
	BNP	86.7	97.4	78.8	85.8	69.8	17.06	17.14	19.87
CIRR	SCC	48.2	72.9	39.6	48.0	31.6	19.76	20.00	20.00
	SAE+FLOPS	57.8	78.1	43.4	53.2	34.8	19.41	19.64	20.00
	PUMA-A	71.5	86.6	49.8	60.1	40.3	19.87	20.00	20.00
	PUMA-A+FLOPS	74.4	87.8	50.5	61.0	40.9	19.56	19.82	20.00
	BNP	76.1	88.4	50.6	60.6	40.7	19.48	19.74	20.00
FashionIQ	SCC	29.4	55.8	12.8	17.7	10.4	19.73	20.00	20.00
	SAE+FLOPS	36.9	61.5	14.7	20.2	11.9	19.35	19.61	20.00
	PUMA-A	51.7	69.1	18.5	24.9	15.3	19.82	20.00	20.00
	PUMA-A+FLOPS	54.2	70.4	18.6	25.0	15.4	19.47	19.80	20.00
	BNP	55.4	70.7	18.2	24.4	15.1	19.39	19.69	20.00
VisualNews	SCC	10.8	31.5	9.8	12.7	7.6	19.99	20.00	20.00
	SAE+FLOPS	15.2	35.7	12.6	16.7	9.8	19.73	20.00	20.00
	PUMA-A	19.7	43.9	16.1	21.8	12.9	20.00	20.00	20.00
	PUMA-A+FLOPS	23.4	46.7	17.5	23.3	14.0	19.82	20.00	20.00
	BNP	25.8	48.1	18.4	24.5	14.7	19.78	20.00	20.00
Fashion200K	SCC	14.8	25.6	7.7	5.6	4.5	19.97	20.00	20.00
	SAE+FLOPS	21.3	30.8	9.5	7.1	5.7	19.60	19.93	20.00
	PUMA-A	27.6	36.5	10.9	8.0	6.8	19.98	20.00	20.00
	PUMA-A+FLOPS	31.5	39.1	11.8	8.6	7.3	19.76	20.00	20.00
	BNP	33.0	40.2	11.7	8.5	7.2	19.70	19.98	20.00

B Complete Matched-Budget Results

Table A4 expands the main table with H@5, R@10, and the distribution of realized posting work. The primary conclusions are unchanged: BNP improves neighborhood coverage and candidate inclusion across all five tasks, while final ranking gains are task dependent.

The complete metrics reinforce the distinction between first-stage preservation and final ranking. BNP has the highest NC_{100} and Hit@1k on all five tasks. The gains over PUMA-A+FLOPS are 1.2, 1.7, 1.2, 2.4, and 1.5 percentage points in NC_{100} on MSCOCO-MBEIR, CIRR, FashionIQ, VisualNews, and Fashion200K, respectively. Realized posting work is never larger than PUMA-A+FLOPS in these comparisons. The ranking metrics are less uniform: BNP is strongest on MSCOCO-MBEIR and VisualNews, approximately tied on Fashion200K, and slightly below PUMA-A+FLOPS on CIRR and FashionIQ. This pattern is consistent with BNP optimizing which teacher-neighborhood items remain available to the second stage rather than directly optimizing the final benchmark ranking.

B.1 Training-Run Variation

Table A5 reports the variation across three training runs for the principal configurations. The differences between BNP and PUMA-A+FLOPS remain larger than the observed run-to-run variation in NC_{100} on the five primary datasets. Query-level evaluation is deterministic for a fixed checkpoint.

Run-to-run variability is small relative to the principal coverage differences. For example, on CIRR BNP reaches 76.1 ± 0.38 NC_{100} compared with 74.4 ± 0.36 for PUMA-A+FLOPS, while on VisualNews the corresponding values are 25.8 ± 0.37 and 23.4 ± 0.44. The final-ranking differences are much closer to the observed training variation, which is another reason to avoid interpreting small nDCG changes as the central contribution. The posting statistics are also stable across runs, indicating that the coverage differences are not explained by one configuration consistently using more realized work.

Table A5: Mean ± standard deviation across three training runs at $B = 20$k. Postings are in thousands.

Dataset	Method	NC_{100}	Hit@1k	nDCG@10	Mean Post.
MSCOCO-MBEIR	SCC	61.8±0.40	91.1±0.27	58.4±0.15	17.64±0.13
	SAE+FLOPS	76.9±0.29	95.2±0.24	65.5±0.23	16.98±0.13
	PUMA-A	83.4±0.42	97.0±0.20	69.0±0.21	18.42±0.08
	PUMA-A+FLOPS	85.5±0.30	97.2±0.30	69.6±0.20	17.21±0.10
	BNP	86.7±0.39	97.4±0.18	69.8±0.17	17.06±0.08
CIRR	SCC	48.2±0.34	72.9±0.21	31.6±0.14	19.76±0.08
	SAE+FLOPS	57.8±0.36	78.1±0.19	34.8±0.15	19.41±0.10
	PUMA-A	71.5±0.31	86.6±0.20	40.3±0.24	19.87±0.12
	PUMA-A+FLOPS	74.4±0.36	87.8±0.29	40.9±0.21	19.56±0.08
	BNP	76.1±0.38	88.4±0.30	40.7±0.24	19.48±0.10
FashionIQ	SCC	29.4±0.33	55.8±0.24	10.4±0.17	19.73±0.12
	SAE+FLOPS	36.9±0.35	61.5±0.17	11.9±0.15	19.35±0.08
	PUMA-A	51.7±0.30	69.1±0.18	15.3±0.21	19.82±0.10
	PUMA-A+FLOPS	54.2±0.48	70.4±0.24	15.4±0.22	19.47±0.10
	BNP	55.4±0.42	70.7±0.22	15.1±0.14	19.39±0.10
VisualNews	SCC	10.8±0.30	31.5±0.25	7.6±0.23	19.99±0.12
	SAE+FLOPS	15.2±0.43	35.7±0.29	9.8±0.24	19.73±0.08
	PUMA-A	19.7±0.37	43.9±0.27	12.9±0.23	20.00±0.10
	PUMA-A+FLOPS	23.4±0.44	46.7±0.30	14.0±0.21	19.82±0.11
	BNP	25.8±0.37	48.1±0.28	14.7±0.19	19.78±0.13
Fashion200K	SCC	14.8±0.40	25.6±0.27	4.5±0.18	19.97±0.10
	SAE+FLOPS	21.3±0.28	30.8±0.18	5.7±0.21	19.60±0.09
	PUMA-A	27.6±0.33	36.5±0.22	6.8±0.17	19.99±0.11
	PUMA-A+FLOPS	31.5±0.43	39.1±0.28	7.3±0.22	19.76±0.09
	BNP	33.0±0.35	40.2±0.19	7.2±0.17	19.70±0.09

Table A6: Neighborhood coverage NC_{100} across posting-budget caps.

Dataset	Method	5k	10k	20k	40k	80k
MSCOCO-MBEIR	SCC	38.9	51.8	61.8	72.4	79.5
	SAE+FLOPS	55.0	67.8	76.9	83.7	87.9
	PUMA-A	64.4	75.6	83.4	89.3	92.0
	PUMA-A+FLOPS	67.1	78.1	85.5	90.5	92.6
	BNP	69.3	80.0	86.7	91.2	92.7
CIRR	SCC	27.4	38.9	48.2	60.7	72.6
	SAE+FLOPS	33.8	45.9	57.8	70.2	80.6
	PUMA-A	39.8	57.4	71.5	82.8	88.4
	PUMA-A+FLOPS	42.7	60.2	74.4	83.7	88.3
	BNP	44.6	62.0	76.1	84.2	87.9
FashionIQ	SCC	14.2	21.9	29.4	40.1	49.0
	SAE+FLOPS	18.3	27.9	36.9	48.2	56.3
	PUMA-A	28.8	40.9	51.7	60.9	66.7
	PUMA-A+FLOPS	31.1	43.5	54.2	62.5	67.1
	BNP	32.4	44.8	55.4	63.0	67.0
VisualNews	SCC	3.9	6.9	10.8	18.5	29.8
	SAE+FLOPS	5.2	9.7	15.2	24.8	37.5
	PUMA-A	7.6	12.9	19.7	31.6	44.9
	PUMA-A+FLOPS	9.1	15.4	23.4	35.7	48.1
	BNP	10.2	17.1	25.8	38.0	49.1
Fashion200K	SCC	6.0	9.7	14.8	22.6	31.9
	SAE+FLOPS	8.4	14.3	21.3	31.1	40.8
	PUMA-A	12.1	19.8	27.6	37.4	45.9
	PUMA-A+FLOPS	14.5	22.9	31.5	40.9	48.2
	BNP	15.4	24.1	33.0	42.1	49.0

C BUDGET CURVES

The main paper visualizes representative budget curves. Tables A6–A8 provide the complete values for all five datasets. Coverage gains are largest in the constrained regime, while final ranking differences are smaller and may change sign as the budget increases.

The full coverage table shows a common shape across datasets. BNP has its largest or near-largest advantage at small and medium budgets, where the choice of sparse support determines which portions of the inverted index can be reached. On MSCOCO-MBEIR, the gap over PUMA-A+FLOPS is 2.2 points at 5k accesses, 1.9 at 10k, and 1.2 at 20k, but only 0.1 at 80k. CIRR follows the same pattern and slightly reverses at 80k, where PUMA-A+FLOPS reaches 88.3 versus 87.9 for BNP.

Table A7: Candidate Hit@1k across posting-budget caps.

Dataset	Method	5k	10k	20k	40k	80k
MSCOCO-MBEIR	SCC	71.1	82.0	91.1	94.3	95.8
	SAE+FLOPS	74.3	85.7	95.2	98.5	99.7
	PUMA-A	75.7	87.3	97.0	99.7	99.7
	PUMA-A+FLOPS	75.8	87.5	97.2	99.7	99.7
	BNP	76.0	87.7	97.4	99.7	99.7
CIRR	SCC	56.9	65.6	72.9	75.5	76.7
	SAE+FLOPS	60.9	70.3	78.1	80.8	82.2
	PUMA-A	67.5	77.9	86.6	89.6	91.1
	PUMA-A+FLOPS	68.5	79.0	87.8	90.9	92.4
	BNP	69.0	79.6	88.4	91.5	93.0
FashionIQ	SCC	43.5	50.2	55.8	57.8	58.7
	SAE+FLOPS	48.0	55.4	61.5	63.6	64.7
	PUMA-A	53.9	62.2	69.1	71.5	72.7
	PUMA-A+FLOPS	54.9	63.4	70.4	72.9	74.1
	BNP	55.1	63.6	70.7	73.2	74.4
VisualNews	SCC	24.6	28.4	31.5	32.6	33.1
	SAE+FLOPS	27.9	32.1	35.7	37.0	37.6
	PUMA-A	34.2	39.5	43.9	45.4	46.2
	PUMA-A+FLOPS	36.4	42.0	46.7	48.3	49.1
	BNP	37.5	43.3	48.1	49.8	50.6
Fashion200K	SCC	20.0	23.0	25.6	26.5	26.9
	SAE+FLOPS	24.0	27.7	30.8	31.9	32.4
	PUMA-A	28.5	32.9	36.5	37.8	38.4
	PUMA-A+FLOPS	30.5	35.2	39.1	40.5	41.1
	BNP	31.4	36.2	40.2	41.6	42.3

Table A8: Final nDCG@10 across posting-budget caps.

Dataset	Method	5k	10k	20k	40k	80k
MSCOCO-MBEIR	SCC	48.5	54.3	58.4	62.7	65.2
	SAE+FLOPS	57.4	62.5	65.5	68.1	69.6
	PUMA-A	62.1	66.2	69.0	70.5	70.8
	PUMA-A+FLOPS	64.7	67.5	69.6	70.8	71.1
	BNP	65.3	68.4	69.8	70.9	71.0
CIRR	SCC	22.0	27.3	31.6	35.8	39.0
	SAE+FLOPS	25.6	30.8	34.8	38.4	40.6
	PUMA-A	29.7	35.8	40.3	41.6	42.3
	PUMA-A+FLOPS	31.0	36.8	40.9	42.2	42.4
	BNP	31.5	37.0	40.7	41.9	42.1
FashionIQ	SCC	5.6	7.9	10.4	12.3	13.4
	SAE+FLOPS	6.9	9.4	11.9	13.8	14.9
	PUMA-A	8.7	11.6	15.3	16.0	16.4
	PUMA-A+FLOPS	9.0	12.0	15.4	16.2	16.5
	BNP	9.2	12.3	15.1	16.1	16.4
VisualNews	SCC	3.5	5.0	7.6	10.9	14.8
	SAE+FLOPS	4.2	6.8	9.8	13.4	16.8
	PUMA-A	5.3	8.9	12.9	16.3	19.2
	PUMA-A+FLOPS	5.9	9.7	14.0	17.1	19.5
	BNP	6.2	10.1	14.7	17.8	19.4
Fashion200K	SCC	2.1	3.1	4.5	5.4	6.2
	SAE+FLOPS	2.7	4.0	5.7	6.9	7.8
	PUMA-A	3.2	4.9	6.8	8.1	9.1
	PUMA-A+FLOPS	3.6	5.4	7.3	8.6	9.7
	BNP	3.8	5.6	7.2	8.5	9.6

VisualNews retains a larger advantage for longer because its much larger collection makes the 20k–40k regimes comparatively restrictive. These curves support a budget-specific interpretation of BNP rather than a claim that the representation dominates unrestricted sparse retrieval.

Candidate Hit@1k generally follows the coverage curves but saturates earlier, especially on MSCOCO-MBEIR where several methods approach the ceiling by 40k postings. On the larger collections, the differences remain visible for longer. At 20k, BNP reaches 88.4 on CIRR, 70.7 on FashionIQ, 48.1 on VisualNews, and 40.2 on Fashion200K, exceeding PUMA-A+FLOPS on each task. This agreement between teacher-neighborhood coverage and ground-truth candidate inclusion shows that the additional preserved teacher structure is not purely an artifact of the teacher metric, although the two quantities are not identical.

Table A9: Full teacher-neighbor fate decomposition at $B = 20$k. The first three columns sum to 100%.

Dataset	Method	Retained	Unvisited	Visited–dropped	Full sparse NC	Truncation loss
MSCOCO-MBEIR	PUMA-A+FLOPS	85.5	6.1	8.4	91.7	6.2
	BNP	86.7	5.2	8.1	92.0	5.3
CIRR	PUMA-A+FLOPS	74.4	17.8	7.8	84.6	10.2
	BNP	76.1	15.7	8.2	85.0	8.9
FashionIQ	PUMA-A+FLOPS	54.2	35.1	10.7	67.3	13.1
	BNP	55.4	33.2	11.4	67.6	12.2
VisualNews	PUMA-A+FLOPS	23.4	65.7	10.9	35.9	12.5
	BNP	25.8	62.5	11.7	36.5	10.7
Fashion200K	PUMA-A+FLOPS	31.5	56.9	11.6	45.0	13.5
	BNP	33.0	54.5	12.5	45.4	12.4

Final nDCG exhibits a weaker and less monotonic dependence on the representation. BNP is strongest at low budgets on several tasks, but the gap often shrinks or changes sign once the candidate set becomes large enough for dense reranking to recover most of the relevant items. For example, on CIRR BNP is ahead at 5k and 10k but is 0.2–0.3 points below PUMA-A+FLOPS from 20k onward. FashionIQ and Fashion200K show a similar separation between coverage and final relevance, whereas VisualNews benefits from BNP through 40k. These results motivate reporting first-stage candidate preservation separately from final ranking quality.

D Candidate-Loss Decomposition

Table A9 reports the full decomposition used in Figure 3. Across all five datasets, BNP reduces the fraction of teacher neighbors that are never visited before the budget terminates. The visited-but-not-retained fraction is nearly unchanged or slightly larger, which separates the reachability effect from a generic improvement in candidate pruning.

The decomposition identifies where the bounded-retrieval gain arises. Across all five tasks, BNP reduces the unvisited fraction relative to PUMA-A+FLOPS: from 6.1 to 5.2 on MSCOCO-MBEIR, 17.8 to 15.7 on CIRR, 35.1 to 33.2 on FashionIQ, 65.7 to 62.5 on VisualNews, and 56.9 to 54.5 on Fashion200K. By contrast, the visited-but-dropped component is approximately unchanged and is slightly larger for BNP on four of the five datasets. Unrestricted sparse coverage differs by only 0.3–0.6 points, while the finite-budget retained fraction differs by 1.2–2.4 points. Thus, most of the advantage appears after imposing the posting cap and is specifically associated with exposing teacher neighbors before traversal terminates.

E Additional Objective Ablations

The main paper reports the central objective comparison. Table A10 additionally removes reconstruction and freezes the document-frequency statistics. Removing reconstruction changes coverage only slightly, whereas stale posting statistics reduce both coverage and candidate inclusion. This sensitivity is consistent with the fact that the reachability surrogate depends on the current distribution of sparse supports.

The ablations separate three sources of improvement. Neighborhood supervision provides the largest single gain over the reconstruction-only SAE, raising NC_{100} from 48.2 to 68.0 on CIRR and from 10.8 to 17.6 on VisualNews. Cost control alone is much weaker, which shows that cheaper supports are not sufficient if they do not preserve the teacher neighborhood. Combining neighborhood supervision with FLOPS or the mean-cost constraint improves coverage further, but full BNP remains strongest at 76.1 on CIRR and 25.8 on VisualNews.

Removing reconstruction has only a small effect on coverage ($76.1 \rightarrow 75.4$ on CIRR and $25.8 \rightarrow 26.0$ on VisualNews), while candidate inclusion and nDCG decline slightly. The more consequential ablation is stale document frequency. Freezing the DF statistics reduces coverage to 72.5 on CIRR and 21.8 on VisualNews, showing that the reachability surrogate benefits from tracking how the

Table A10: Expanded ablation results at $B = 20$k. Post. denotes mean realized postings in thousands.

	CIRR				VisualNews			
Variant	NC	Hit	nDCG	Post.	NC	Hit	nDCG	Post.
Vanilla SAE	48.2	72.9	31.6	19.76	10.8	31.5	7.6	19.99
Neighborhood only	68.0	84.0	38.9	19.92	17.6	40.5	11.8	20.00
Budget only	54.6	77.1	33.1	18.68	13.7	35.1	8.8	19.41
Neighborhood + FLOPS	70.3	85.7	39.6	19.43	21.4	45.0	13.4	19.83
Mean-cost constraint	72.0	86.9	40.0	19.32	23.1	46.5	13.9	19.76
PUMA-A + FLOPS	74.4	87.8	40.9	19.56	23.4	46.7	14.0	19.82
BNP	76.1	88.4	40.7	19.48	25.8	48.1	14.7	19.78
BNP w/o reconstruction	75.4	88.1	40.4	19.44	26.0	47.4	14.3	19.72
BNP stale DF	72.5	86.7	39.6	19.63	21.8	44.5	13.1	19.86

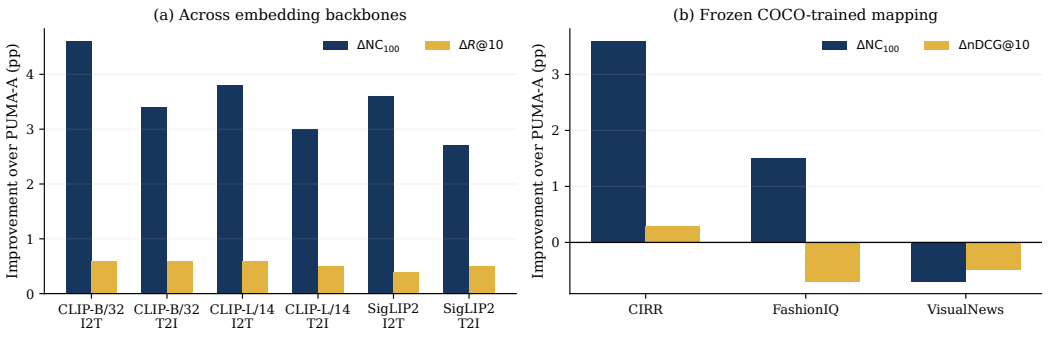


Figure A1: Generalization across embedding backbones and target domains. Left: changes relative to PUMA-A after retraining the sparse mapping for each backbone. Right: frozen transfer of an MSCOCO-MBEIR-trained mapping.

learned support distribution evolves during training. This result also distinguishes BNP from a fixed frequency penalty applied once before optimization.

F Generalization across Backbones and Domains

Figure A1 visualizes the relative changes summarized in the main paper. Tables A11 and A12 report the corresponding absolute values. Cross-backbone experiments retrain the sparse transformation for each encoder. Frozen transfer keeps the MSCOCO-MBEIR-trained sparse mapping fixed and rebuilds only the target collection index and document-frequency statistics.

Figure A1 makes the two generalization regimes visually distinct. Retraining BNP for a new embedding family produces a consistent positive coverage shift, whereas transferring one fixed sparse mapping across datasets can weaken or reverse the gain. This distinction is important for the scope of the method: the training objective is not specific to one dense geometry, but the resulting sparse coordinates are still adapted to the distribution on which they were learned.

The cross-backbone results show that the coverage effect is not tied to the Qwen representation family. Relative to PUMA-A, BNP improves NC_{100} by 4.6/3.4 points for CLIP ViT-B/32 in I2T/T2I retrieval, 3.8/3.0 points for CLIP ViT-L/14, and 3.6/2.7 points for SigLIP 2. PUMA-A+FLOPS closes part of this gap but remains below BNP in every backbone-direction pair. R@10 changes much less, by roughly 0.4–0.6 points over PUMA-A, reproducing the same separation between candidate preservation and final ranking observed in the main Qwen experiments. Because the sparse

Table A11: Absolute cross-backbone results on COCO-5K. Each sparse mapping is trained separately for its backbone.

Backbone	Direction	Method	NC_{100}	R@10
CLIP ViT-B/32	I2T	PUMA-A	78.6	81.4
CLIP ViT-B/32	I2T	PUMA-A+FLOPS	81.5	81.8
CLIP ViT-B/32	I2T	BNP	83.2	82.0
CLIP ViT-B/32	T2I	PUMA-A	82.7	65.0
CLIP ViT-B/32	T2I	PUMA-A+FLOPS	84.7	65.4
CLIP ViT-B/32	T2I	BNP	86.1	65.6
CLIP ViT-L/14	I2T	PUMA-A	80.1	84.6
CLIP ViT-L/14	I2T	PUMA-A+FLOPS	82.6	84.9
CLIP ViT-L/14	I2T	BNP	83.9	85.2
CLIP ViT-L/14	T2I	PUMA-A	84.0	68.8
CLIP ViT-L/14	T2I	PUMA-A+FLOPS	85.8	69.0
CLIP ViT-L/14	T2I	BNP	87.0	69.3
SigLIP 2 B/16@256	I2T	PUMA-A	82.4	91.2
SigLIP 2 B/16@256	I2T	PUMA-A+FLOPS	84.9	91.4
SigLIP 2 B/16@256	I2T	BNP	86.0	91.6
SigLIP 2 B/16@256	T2I	PUMA-A	86.3	82.7
SigLIP 2 B/16@256	T2I	PUMA-A+FLOPS	88.0	83.0
SigLIP 2 B/16@256	T2I	BNP	89.0	83.2

Table A12: Frozen transfer from an MSCOCO-MBEIR-trained sparse mapping.

Target	Method	NC_{100}	nDCG@10
CIRR	PUMA-A	52.8	34.7
CIRR	BNP	56.4	35.0
FashionIQ	PUMA-A	37.3	11.5
FashionIQ	BNP	38.8	10.8
VisualNews	PUMA-A	13.8	9.9
VisualNews	BNP	13.1	9.4

transformation is retrained for each encoder, these experiments test whether the objective transfers across embedding geometries rather than whether one learned mapping transfers unchanged.

Frozen transfer is less consistent than cross-backbone retraining. A mapping learned on MSCOCO-MBEIR improves CIRR coverage from 52.8 to 56.4 and slightly improves nDCG from 34.7 to 35.0. FashionIQ retains a smaller coverage advantage, 38.8 versus 37.3, but its nDCG decreases from 11.5 to 10.8. On VisualNews both metrics decline slightly. The contrast with Table A11 indicates that the BNP objective itself generalizes across embedding families, whereas a particular learned sparse mapping remains sensitive to the collection and query distribution on which it was trained.

G Traversal Policy and Hyperparameter Sensitivity

G.1 Traversal Policy

The representation and traversal policy jointly determine the bounded candidate set. Table A13 compares the default balanced round-robin policy P_0 with a query-term-priority policy P_1. Changing the policy alters absolute coverage and the size of the BNP advantage, supporting the use of a fixed policy for representation comparisons.

Changing the traversal policy affects both absolute coverage and the size of the representation gap. On CIRR, switching from P_0 to P_1 improves PUMA-A from 71.5 to 73.2 NC_{100}, while BNP changes from 76.1 to 75.6, shrinking the gap from 4.6 to 2.4 points. VisualNews shows an even larger policy effect on PUMA-A, which rises from 19.7 to 23.0, whereas BNP changes from 25.8 to 25.1. Final nDCG follows the same broad pattern. These results show that representation quality under a hard posting cap is conditional on the serving policy, and justify holding π fixed when comparing representation-learning objectives.

Table A13: Traversal-policy sensitivity at $B = 20$k. Post. is in thousands.

Dataset	Policy	Method	NC	nDCG	Post.
CIRR	P0 round-robin	PUMA-A	71.5	40.3	19.87
CIRR	P0 round-robin	BNP	76.1	40.7	19.48
CIRR	P1 term-priority	PUMA-A	73.2	40.6	19.74
CIRR	P1 term-priority	BNP	75.6	40.4	19.39
VisualNews	P0 round-robin	PUMA-A	19.7	12.9	20.00
VisualNews	P0 round-robin	BNP	25.8	14.7	19.78
VisualNews	P1 term-priority	PUMA-A	23.0	13.5	19.92
VisualNews	P1 term-priority	BNP	25.1	14.2	19.73

Table A14: Sensitivity to active support size k and dictionary size m on CIRR.

Support-size sweep ($m = 32{,}768$)					Dictionary-size sweep ($k = 128$)					
k	NC	Hit	nDCG	Post.	m	NC	Hit	nDCG	Post.	Active atoms
64	71.4	85.0	39.5	12.84	16,384	72.8	86.7	39.6	19.82	438
96	74.9	87.2	40.4	16.31	24,576	74.9	87.6	40.3	19.67	552
128	76.1	88.4	40.7	19.48	32,768	76.1	88.4	40.7	19.48	711
160	75.7	88.7	41.0	19.85	49,152	76.0	88.3	40.6	19.39	905
192	74.8	88.4	40.8	19.94	65,536	75.4	88.1	40.5	19.35	1080
256	73.6	87.9	40.4	19.98						

G.2 Sparse Support Size and Dictionary Size

The default $k = 128$ balances neighborhood coverage and posting work. Increasing k beyond this point consumes nearly the entire budget without further improving coverage. Increasing the dictionary from 16k to 32k helps distribute the sparse support, while larger dictionaries provide little additional benefit.

Active support size. Increasing k from 64 to 128 raises NC_{100} from 71.4 to 76.1 and Hit@1k from 85.0 to 88.4, while nDCG@10 improves from 39.5 to 40.7. This gain comes with more realized work, from 12.84k to 19.48k postings. Increasing the support further does not improve neighborhood preservation: at $k = 160$, NC_{100} falls slightly to 75.7 even though nDCG@10 reaches 41.0, and by $k = 256$ coverage drops to 73.6 while essentially the entire 20k budget is consumed. We therefore use $k = 128$ as the default operating point.

Dictionary size. Expanding the dictionary from 16,384 to 32,768 improves NC_{100} from 72.8 to 76.1 and nDCG@10 from 39.6 to 40.7, while realized posting work slightly decreases. The number of corpus-active atoms grows from 438 to 711, indicating that the additional capacity spreads retrieval evidence over more coordinates. Larger dictionaries continue to increase active atom count but do not improve retrieval: at $m = 49{,}152$ and 65,536, coverage is 76.0 and 75.4. The default $m = 32{,}768$ therefore captures most of the useful capacity increase without adding dimensions that do not improve bounded neighborhood preservation.

G.3 Teacher Neighborhood Size and Rerank Depth

Training with a very small teacher neighborhood emphasizes the closest neighbors and gives the highest NC_{10}, whereas larger neighborhoods improve broader coverage at a small cost in final nDCG. Rerank depth does not affect first-stage coverage and saturates around 500 candidates in this setting.

Teacher neighborhood size. A small L_{train} concentrates supervision on the closest teacher items. With $L_{\text{train}} = 10$, the model achieves the highest NC_{10} (88.9) and nDCG@10 (41.1), but its NC_{100} is only 71.7. As the training neighborhood grows, broader coverage improves: NC_{100} reaches 76.1 at $L_{\text{train}} = 100$ and 76.7 at 200, while NC_{10} and nDCG decline gradually. We use $L_{\text{train}} = 100$ as a balanced setting aligned with the primary NC_{100} evaluation.

Table A15: Teacher-neighborhood and reranking-depth sensitivity on CIRR.

Teacher neighborhood					Dense rerank depth				
L_{train}	NC_{10}	NC_{50}	NC_{100}	nDCG	RR	NC_{100}	Hit	nDCG	R@10
10	88.9	76.0	71.7	41.1	100	76.1	88.4	39.8	59.0
25	88.4	78.5	73.8	41.0	200	76.1	88.4	40.4	60.0
50	87.9	80.2	75.6	40.9	500	76.1	88.4	40.8	60.7
100	87.3	80.0	76.1	40.7	1000	76.1	88.4	40.7	60.6
200	86.5	80.4	76.7	40.2					

Table A16: High-DF interventions on COCO-5K image-to-text retrieval.

Method	Intervention	NC_{100}	R@10	Postings
SCC	None	58.4	76.7	20,000
SCC	Mask top-8 high-DF	59.1	77.0	18,620
SCC	Downweight high-DF x0.5	58.7	76.9	19,110
SCC	Random mask 8 matched	57.4	76.1	19,020
SCC	Query support 32-¿24	56.1	75.2	16,480
BNP	None	83.2	82.0	19,420
BNP	Mask top-8 high-DF	82.8	81.9	18,290
BNP	Downweight high-DF x0.5	83.3	82.1	18,870
BNP	Random mask 8 matched	81.9	81.4	18,810
BNP	Query support 32-¿24	80.8	80.8	16,220

Dense reranking depth. Changing rerank depth leaves NC_{100} and Hit unchanged at 76.1 and 88.4, confirming that these metrics are determined entirely by the sparse candidate stage. Increasing reranking from 100 to 500 candidates improves nDCG@10 from 39.8 to 40.8 and R@10 from 59.0 to 60.7. Expanding to 1,000 candidates gives no further gain, with the metrics changing slightly to 40.7 and 60.6. We retain a depth of 1,000 in the main experiments so that second-stage truncation does not become an additional bottleneck.

H High-Document-Frequency Dimensions

We further test whether the gains can be reproduced by simple interventions on high-document-frequency dimensions. Table A16 shows that masking a small number of high-DF dimensions can slightly improve the older SCC representation, but the same operation does not improve BNP. Random masking and reducing the query support hurt both methods. Downweighting alone should not be interpreted as a posting reduction unless the traversal rule also changes which entries are visited.

For SCC, explicitly suppressing the most frequent dimensions can be slightly beneficial. Masking the eight highest-DF dimensions raises NC_{100} from 58.4 to 59.1 and R@10 from 76.7 to 77.0 while reducing posting accesses from 20,000 to 18,620. Downweighting gives a smaller improvement. The matched random mask and the reduction from 32 to 24 query dimensions both lower coverage and R@10, despite also reducing posting work. Lower cost alone therefore does not explain the benefit of the high-DF intervention.

BNP shows a different pattern. Its unmodified representation reaches 83.2 NC_{100}, and masking the highest-DF dimensions slightly reduces coverage to 82.8. Downweighting leaves the result essentially unchanged, whereas random masking and support reduction lower coverage to 81.9 and 80.8. The smaller benefit from explicit high-DF suppression is consistent with BNP already learning to distribute useful support away from costly posting-list concentrations. It also shows why document frequency is used as a cost signal rather than as the retrieval objective itself.

I Sparse-Dimension Utilization

Table A17 summarizes the distribution of document frequency across active sparse dimensions. BNP uses a broader set of dimensions and places less posting mass in the highest-frequency tail

Table A17: Sparse-dimension utilization and document-frequency concentration.

Dataset	Method	Active atoms	DF Gini	Top-1% mass	Median DF	P90 DF	P99 DF
CIRR	PUMA-A	481	0.82	38.4	74	930	6,890
CIRR	PUMA-A+FLOPS	536	0.76	31.7	69	760	5,110
CIRR	BNP	711	0.68	24.9	61	610	3,980
FashionIQ	PUMA-A	424	0.80	36.2	119	1,230	8,200
FashionIQ	BNP	648	0.69	25.6	97	920	6,030
VisualNews	PUMA-A	497	0.86	42.7	1380	21,200	101,000
VisualNews	BNP	782	0.73	29.8	1020	14,600	72,800

Table A18: One-million-item stress test at a 20k posting cap.

Method	NC_{100}	Hit@1k	H@5	R@10	nDCG@10	Mean Post.
Dense exact FP16	–	98.4	77.1	83.4	68.0	–
SCC	34.8	76.5	48.7	60.5	46.0	20.00
SAE+FLOPS	49.7	86.3	59.8	71.8	55.9	19.81
PUMA-A	56.9	90.1	65.9	78.2	62.2	20.00
PUMA-A+FLOPS	59.6	91.0	67.1	79.3	63.0	19.87
BNP	62.1	92.0	68.0	79.8	63.5	19.82

than PUMA-A. This distribution is consistent with the retrieval-work objective, but it is not itself used as the primary measure of representation quality.

The learned dimension statistics provide a representation-level view of this redistribution. On CIRR, PUMA-A uses 481 active atoms with a DF Gini coefficient of 0.82, and the most frequent 1% of dimensions account for 38.4% of total posting mass. PUMA-A+FLOPS reduces these values to 0.76 and 31.7%, while BNP uses 711 active atoms with a Gini coefficient of 0.68 and a top-1% mass of 24.9%. The P99 document frequency also falls from 6,890 for PUMA-A to 3,980 for BNP.

The same pattern appears on FashionIQ and VisualNews. On FashionIQ the top-1% posting mass decreases from 36.2% to 25.6%; on VisualNews it falls from 42.7% to 29.8%, while P99 DF decreases from 101,000 to 72,800. BNP therefore uses a broader set of coordinates and relies less on the most expensive tail of the posting distribution. These statistics are descriptive rather than target metrics: a more uniform dictionary is only useful insofar as it helps preserve candidates under the serving budget.

J Million-Item Stress Test

To test whether the bounded-retrieval behavior persists when the collection is enlarged, we construct a one-million-item stress setting by adding external distractors while keeping the query protocol unchanged. The posting cap remains 20k. Table A18 shows that all sparse methods lose coverage as the collection grows, but BNP retains the strongest candidate coverage and candidate inclusion among the sparse systems.

Increasing the collection to one million items makes the fixed 20k posting cap substantially more restrictive. All sparse systems lose coverage relative to unrestricted dense search, but the ordering among them remains consistent. SCC retains 34.8% of the teacher top-100 neighborhood, SAE+FLOPS reaches 49.7, PUMA-A reaches 56.9, and PUMA-A+FLOPS reaches 59.6. BNP obtains the highest sparse coverage at 62.1 and the highest Hit@1k at 92.0 while using 19.82k mean postings.

The additional candidate preservation is reflected in the downstream metrics in this stress setting: relative to PUMA-A+FLOPS, BNP improves H@5 from 67.1 to 68.0, R@10 from 79.3 to 79.8, and nDCG@10 from 63.0 to 63.5. A clear gap to exact dense retrieval remains, however. Dense search reaches 98.4 Hit@1k and 68.0 nDCG@10. The experiment therefore supports persistence of the candidate-generation advantage at larger collection scale without implying that sparsification removes the quality–efficiency trade-off.

Table A19: Analytic storage accounting for a 1M-item, 2,048-dimensional collection. Values are MB.

Configuration	Dense store	Index/meta.	Sparse encoder	Total	Notes
Dense Flat FP32	8192.00	0.00	0.00	8192.00	1M $\times$ 2048 $\times$ 4 bytes
Dense Flat FP16	4096.00	0.00	0.00	4096.00	1M $\times$ 2048 $\times$ 2 bytes
Dense INT8 + scales	2048.00	4.00	0.00	2052.00	Approximate
HNSW Flat $M = 32$	8192.00	256.00	0.00	8448.00	Base graph estimate; excludes allocator overhead
IVF4096-PQ64x8	0.00	107.65	0.00	107.65	Codes, IDs, and centroids
Sparse-only $k = 128$	0.00	768.39	134.28	902.67	int32 doc ID + FP16 weight; query encoder only
Sparse + FP16 dense rerank	4096.00	768.39	134.28	4998.67	Hybrid total

K Storage and Runtime Analysis

K.1 Storage Accounting

We report storage separately for sparse-only candidate generation and for the two-stage setting that also keeps FP16 dense embeddings for reranking. The sparse posting format stores an int32 item identifier and an FP16 weight per posting. Frozen backbone parameters and transient runtime buffers are excluded from all entries. Under these assumptions, sparse-only retrieval is substantially smaller than a dense FP16 store, whereas the hybrid system is larger because it retains both representations.

The storage analysis separates sparse-only candidate generation from the hybrid system that retains dense embeddings for reranking. A one-million item FP32 dense store requires 8,192 MB and an FP16 store requires 4,096 MB. The sparse-only configuration uses 768.39 MB for the index and 134.28 MB for the sparse encoder, for a total of 902.67 MB, about 22% of the FP16 dense-vector storage under these assumptions.

The hybrid configuration has a different trade-off. Retaining FP16 dense embeddings together with the sparse index raises total storage to 4,998.67 MB, which is larger than the FP16 dense-only store. We therefore do not attribute an end-to-end storage reduction to the two-stage system; its purpose is to avoid comparing the query against every dense vector in the first stage. HNSW and IVF-PQ are shown as engineering references with different memory–quality trade-offs rather than as storage-matched versions of BNP.

K.2 Latency

System timing excludes dense query encoding so that the comparison isolates retrieval and optional reranking. ANN baselines are included as engineering references and are not matched by posting count. Sparse candidate generation has similar latency across the learned representations because they use the same index format and traversal cap.

Exact dense FP32 retrieval is the slowest reference in this profile, with 142.0 ms P50 and 188.0 ms P90 latency; INT8 quantization reduces these values to 49.0 and 72.0 ms. The ANN references are faster still: HNSW reaches 7.4/12.1 ms and IVF-PQ 2.2/3.8 ms P50/P90, although their retrieval quality and work are not matched to the sparse systems.

The learned sparse methods occupy a narrow latency range because they share the same index structure and posting cap. BNP has 6.0 ms P50 and 9.0 ms P90 latency, decomposed into roughly 4.1 ms of sparse retrieval and 1.6 ms of dense reranking. SCC, PUMA-A, and PUMA-A+FLOPS differ by less than half a millisecond at P50. The reachability objective therefore changes training and support allocation without adding a new inference module. BNP should not be read as the fastest ANN

Table A20: Retrieval latency in the 1M-item setting with cached query embeddings.

Configuration	P50 (ms)	P90 (ms)	Retrieve (ms)	Rerank (ms)	Notes
Dense Flat FP32	142.0	188.0	142.0	0.0	CPU exact dense
Dense INT8	49.0	72.0	49.0	0.0	quantized dense
HNSW M32	7.4	12.1	7.4	0.0	not matched postings
IVF4096-PQ64x8	2.2	3.8	2.2	0.0	not matched postings
SCC + RR	5.6	8.7	3.8	1.6	cached query embedding
PUMA-A + RR	5.8	9.2	4.0	1.6	cached query embedding
PUMA-A+FLOPS + RR	5.9	9.1	4.0	1.6	cached query embedding
BNP + RR	6.0	9.0	4.1	1.6	cached query embedding

Table A21: Offline training and index-construction cost.

Artifact	Time (min)	Hardware	Output (MB)	Online requirement
Sparse encoder training - SCC	186.00	1x A100 80GB	268.6	encoder only
Sparse encoder training - PUMA-A	318.00	1x A100 80GB	268.6	encoder only
Sparse encoder training - BNP	361.00	1x A100 80GB	268.6	encoder only
Encode 1M corpus - BNP	11.60	1x A100 80GB	768.4	yes
Build inverted index - BNP	6.40	32-core CPU	768.4	yes
Load / mmap sparse index	0.18	NVMe + RAM	768.4	yes

system in this table; its system cost is instead comparable to the other learned sparse representations under the controlled retrieval protocol.

K.3 Offline Cost and Serving Parallelism

Table A21 reports the one-time cost of learning and building the sparse representation. Table A22 reports serving behavior as thread count and query batch size are varied. The index itself is unchanged across these measurements.

BNP requires more one-time training than the simpler sparse objectives. SCC training takes 186 minutes on one A100 80GB, PUMA-A takes 318 minutes, and BNP takes 361 minutes. Relative to PUMA-A, the additional budget-aware training is 43 minutes, or roughly 14%, while the final encoder footprint remains the same at 268.6 MB.

After training, corpus preparation is comparatively small in the one million-item setting. Encoding the collection takes 11.6 minutes on one A100, inverted-index construction takes 6.4 minutes on a 32-core CPU, and loading the 768.4 MB index through memory mapping takes approximately 0.18 minutes. These operations are offline; online retrieval does not repeat the reachability calculations used during training.

With batch size one, increasing the thread count from 1 to 8 reduces P50 latency from 14.6 to 6.0 ms and raises throughput from 68.5 to 166.7 queries/s. Scaling to 16 threads provides little additional benefit: P50 changes only to 5.9 ms and P90 slightly increases to 9.4 ms. Eight threads therefore mark the practical saturation point for this setup.

Batching mainly improves throughput. With eight threads, batch size four reaches 851.1 queries/s with 4.7 ms P50 latency, and batch size eight reaches 1,860.5 queries/s with the lowest P50/P90 values of 4.3/6.6 ms. A batch of 16 increases throughput further to 3,555.6 queries/s while latency rises slightly to 4.5/6.9 ms. The serving implementation therefore benefits from moderate batching, with larger batches trading a small amount of latency for additional throughput.

L Scope of the Additional Results

The additional experiments clarify the scope of the main claim. First, neighborhood preservation is conditioned on the serving regime. Changing the traversal policy changes which posting lists receive the available work, and increasing the budget eventually reduces the importance of the learned

Table A22: Threading and batching for BNP in the 1M-item setting.

Threads	Batch	P50 (ms)	P90 (ms)	QPS
1	1	14.6	21.8	68.5
2	1	9.8	15.2	102.0
4	1	7.1	10.8	140.8
8	1	6.0	9.0	166.7
16	1	5.9	9.4	169.5
8	4	4.7	7.0	851.1
8	8	4.3	6.6	1860.5
8	16	4.5	6.9	3555.6

support allocation. BNP should therefore be compared under a specified policy and budget rather than treated as having a single execution-independent retrieval quality.

Second, the reachability objective primarily affects candidate exposure. The candidate-loss decomposition shows that the largest change is in teacher neighbors that are never visited; the visited-but-dropped component changes little, and the final ranking metrics can remain flat or slightly decline even when coverage improves. Candidate accessibility and task relevance are consequently related but distinct quantities.

Third, the training principle transfers more reliably than a particular learned mapping. Retraining BNP on CLIP and SigLIP 2 reproduces the coverage gains, whereas frozen cross-dataset transfer weakens on FashionIQ and reverses on VisualNews. Finally, the system analysis shows that BNP does not introduce a special online computation beyond the sparse retrieval pipeline, but the hybrid sparse+dense system retains the storage cost of dense reranking. Taken together, these results define BNP as a candidate-generation criterion conditioned on the representation, workload, and execution policy, rather than as a universal replacement for ranking or ANN design.