Adaptive Representations for Semantic Search

Aniket Rege [*1] **Aditya Kusupati** [*1 2] **Sharan Ranjit** [1] **Sham Kakade** [3] **Prateek Jain** [+2] **Ali Farhadi** [+1]

Abstract

Web-scale search systems use a large neural network to embed the query which is then hooked into a separate approximate nearest neighbour search (ANNS) pipeline to retrieve similar data points. Such approaches use a rigid – potentially high-dimensional – representation out of encoder to perform the entire search. This can be far from optimal accuracy-compute trade-off. In this paper, we argue that in different stages of ANNS, we can use representations of different capacities, *adaptive representations*, to ensure that the accuracy-compute tradeoff can be met nearly optimally. In particular, we introduce AdANNS, a novel ANNS design paradigm that explicitly leverages the flexibility and adaptive capabilities of the recently introduced Matryoshka Representations (Kusupati et al., 2022). We demonstrate that using AdANNS to construct the search data structure (AdANNS-C) provides state-of-the-art accuracy-compute tradeoff; AdANNS powered inverted file index (IVF) is up to $\mathbf{1.5}\%$ more accurate or up to $\mathbf{100}\times$ faster ImageNet-1K retrieval. We also show that matryoshka representations can power compute-aware adaptive search during inference (AdANNS-D) on a fixed ANNS (IVF) structure and be up to $\mathbf{16}\times$ faster for similar accuracy. Finally, we explore the applicability of adaptive representations across ANNS building blocks and further analyze the choice of matryoshka representations for semantic search. Code is open-sourced at `https://github.com/RAIVNLab/AdANNS`.

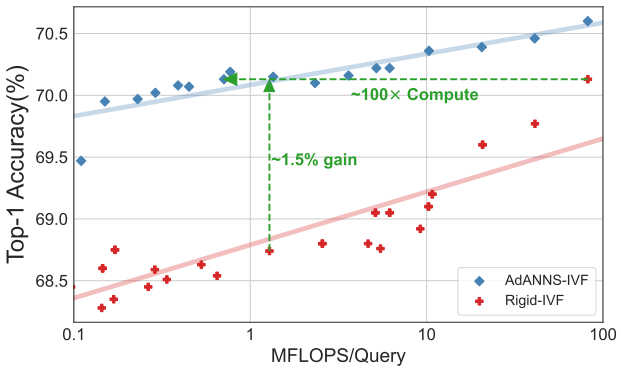


Figure 1. AdANNS powered by the built-in adaptivity of matryoshka representations (Kusupati et al., 2022) significantly outperforms the status quo ANNS using Rigid-IVF (IVF-RR) (Sivic & Zisserman, 2003). AdANNS-IVF can be either $100\times$ more efficient or 1.5% more accurate than Rigid-IVF .

1. Introduction

Semantic similarity search (Johnson et al., 2019) on learned representations (Nayak, 2019; Waldburger, 2019; Neelakantan et al., 2022) is a major component in retrieval pipelines (Dean, 2009; Brin & Page, 1998). In its simplest form, semantic search methods learn to embed queries as well as a large number (N) of data points in a d-dimensional vector space and retrieve the nearest (in embedding space) examples for a given query. That is, per query, retrieval computation scales as $\mathcal{O}(dN)$, where N is routinely higher than 1B. Owing to the infeasibility of exhaustive search, approximate nearest neighbor search (ANNS) (Indyk & Motwani, 1998) is now a standard approach to reduce retrieval costs.

ANNS pipelines are built on the default Rigid Representations (RRs), typically high-dimensional outputs from a neural network (Beaumont, 2021). Let us consider inverted file index (IVF) (Sivic & Zisserman, 2003), a simple yet accurate ANNS technique at web-scale (Guo et al., 2020). IVF has two stages (see Figure 2) during inference – mapping the query to a cluster of data points (Lloyd, 1982) and then a linear scan over all data points in the cluster to find the NN. Currently, IVF utilizes the same high-dimensional RR for both phases, however, this design choice can be far from optimal accuracy-compute trade-off.

We argue that utilizing different capacity representations with similar semantic information for clustering and linear scan can lead to a better ANNS index. But how do we find such *adaptive representations*? These desired adaptive representations should be cheap to get and provide

[*]Equal contribution, [+]Equal advising [1]University of Washington [2]Google Research [3]Harvard University. Correspondence to: Aditya Kusupati <kusupati@cs.washington.edu>.

Preprint.

accurate low-dimensional representations that have some semblance to the high-dimensional representation. The low-dimensional embeddings help in better clustering while being aware of the high-dimensional embeddings used for the linear scan. Post-hoc dimensionality reduction techniques like SVD (Golub & Kahan, 1965) and random projections (Johnson, 1984) on high-dimensional RRs are potential candidates but fail to preserve the overall geometry of the embedding space and are highly inaccurate (Figure 3).

In contrast, we identify that the recently proposed Matryoshka Representations (MRs) (Kusupati et al., 2022) align well with the desired properties of adaptive representations. Matryoshka representations pack information in a hierarchical nested manner ie., the first m-dimensions of the d-dimensional MR form an accurate low-dimensional representation while being aware of the information in the higher dimensions. This allows us to deploy MRs in two ways as part of IVF: (a) construction with approximate clustering and (b) inference with approximate distance computation using representations of varying capacities.

In this paper, we propose the use of *adaptive representations* across different stages of ANNS to ensure near optimal accuracy-compute trade-off. As an effort in this direction, we introduce AdANNS[1], a generalizable paradigm for ANNS that explicitly leverages the adaptive capabilities of matryoshka representations. AdANNS is applicable to both construction (AdANNS-C), through design adaptivity, and inference (AdANNS-D), through distance adaptivity, of ANNS data structures. We show that AdANNS-C achieves near-optimal accuracy-compute trade-off while AdANNS-D enables compute-aware elastic search on pre-built indices.

Specifically, AdANNS construction of IVF (AdANNS-IVF-C) results in a significant improvement in accuracy-compute trade-off over existing IVF indices on rigid representations (see Figure 1) – up to 1.5% more accurate for same compute and $100\times$ lower compute for same accuracy on ImageNet-1K 1-NN retrieval (Section 4.1). At the same time, AdANNS-IVF-D can enable compute-aware search during inference on ANNS structures pre-built on high-dimensional MR. We observe that adaptive inference results in up to $16\times$ faster yet similarly accurate search on an IVF structure built on 2048-d MR (Section 4.2).

We also explore the applicability of adaptive representation across ANNS building blocks like product quantization (PQ) (Jegou et al., 2010), HNSW (Malkov & Yashunin, 2020). Our preliminary results show that AdANNS is complementary to popular ANNS techniques (Section 5), thus indicating a potential combination of AdANNS with such methods. Lastly, we further analyze and justify the choice of matryoshka representations for semantic search (Section 6).

[1]Pronounced "A Dance"

2. Related Work

Approximate nearest neighbour search (ANNS) is a paradigm to come as close (Clarkson, 1994) to retrieving the "true" NN without the exorbitant search costs (Indyk & Motwani, 1998; Weber et al., 1998). The "approximate" nature of the search comes from data pruning as well as the cheaper distance computation that make real-time web-scale retrieval possible. In its naive form, NN-search has a complexity of $\mathcal{O}(dN)$; d is the data dimensionality used for distance computation and N is the number of data points in the database. ANNS employs each of these approximations to reduce the linear dependence on the data dimensionality (cheaper distance computation) and data points visited during the search (data pruning).

Cheaper distance computation. From a bird's eye view, cheaper distance computation is always obtained through dimensionality reduction (quantization included). PCA/SVD (Golub & Kahan, 1965; Jolliffe & Cadima, 2016) can reduce dimensionality and preserve distances only to a limited extent without sacrificing accuracy. On the other hand, quantization-based techniques (Gray, 1984; Jegou et al., 2010; Ge et al., 2013; Chen et al., 2020) have proved extremely crucial for relatively accurate yet cheap distance computation and simultaneously reduce the memory overhead significantly. Another naive solution is to independently train the representation function with varying low-dimensional information bottlenecks (Kusupati et al., 2022). Despite being extremely accurate, such representations are rarely used owing to the costs associated with maintaining multiple models and databases for each dimensionality.

Data pruning. Enabled by various data structures, data pruning reduces the number of data points visited as part of the search. This is often achieved through hashing (Datar et al., 2004; Salakhutdinov & Hinton, 2009), trees (Friedman et al., 1977; Sivic & Zisserman, 2003; Bernhardsson, 2018; Guo et al., 2020) and graphs (Malkov & Yashunin, 2020; Jayaram Subramanya et al., 2019). More recently there have been efforts towards end-to-end learning of the search data structures (Kraska et al., 2018; Kusupati et al., 2021; Gupta et al., 2022). However, web-scale ANNS indices are often constructed on rigid d-dimensional real vectors using the aforementioned data structures that assist with the real-time search. For a more comprehensive review of ANNS structures please refer to (Cai, 2021; Li et al., 2020; Wang et al., 2021)

Composite indices. ANNS pipelines often benefit from the complementary nature of various building blocks (Johnson et al., 2019; Radford et al., 2021). In practice, often the data structures (coarse-quantizer) like IVF and HNSW are combined with cheaper distance alternatives like PQ

(fine-quantizer) for massive speed-ups in web-scale search. While the data structures are built on d-dimensional real vectors, past work and deployments consistently show that PQ can be safely used for distance computation during search time. As evident in modern web-scale ANNS systems like DiskANN (Jayaram Subramanya et al., 2019), the data structures are built on d-dimensional real vectors but work with PQ vectors (often 32-byte) for fast distance computations.

Status quo. Despite the Herculean advances in representation learning (He et al., 2016; Radford et al., 2021), ANNS progress is often only benchmarked on fixed representation vectors provided for about a dozen million to billion scale datasets (Aumüller et al., 2020; Simhadri et al., 2022) with limited access to the raw data. This resulted in the improvement of algorithmic design for rigid representations (RRs) that are often not specifically designed for search. All the existing ANNS methods work with the assumption of using the provided d-dimensional representation which might not be Pareto-optimal for the accuracy-compute trade-off in the first place. Note that the lack of raw-image-based benchmarks led us to use ImageNet-1K and ImageNet-4K (Kusupati et al., 2022) based image retrieval for experimentation.

In this paper, we investigate the utility of adaptive representations – embeddings of different dimensionalities having similar semantic information – in improving the design space of ANNS algorithms. This helps in transitioning out of restricted construction and inference on rigid representations for ANNS. To this end, we extensively use Matryoshka Representations (Kusupati et al., 2022) which have desired adaptive properties in-built. To the best of our knowledge, this is the first work to leverage adaptive representations for different phases of ANNS data structure construction and inference thus improving accuracy-compute trade-off.

3. Problem Setup, Notation and Preliminaries

The problem setup of approximate nearest neighbor search (ANNS) consists of a database of N data points, $[x_1, x_2, \ldots, x_N]$, and a query q where the goal is to "approximately" retrieve the nearest data point to the query. Both the database and query are embedded to $\mathbb{R}^d$ using a representation function $\phi : \mathcal{X} \to \mathbb{R}^d$, often a neural network that can be learned through various paradigms of representation learning (Bengio, 2012; He et al., 2016; Nayak, 2019; He et al., 2020; Radford et al., 2021).

The d-dimensional representations from ϕ can also have a nested structure in-built – Matryoshka Representations (MRs) (Kusupati et al., 2022) – $\phi^{\text{MR}(d)}$. MR inherently contains low-dimensional representations of varying granularities that can be accessed for free – first m-dimensions ($m \in [d]$) ie., $\phi^{\text{MR}(d)}[1:m]$ from the d-dimensional Matryoshka Representation (MR) form an m-dimensional representation which is as accurate as its independently trained rigid representation (RR) counterpart – $\phi^{\text{RR}(m)}$.

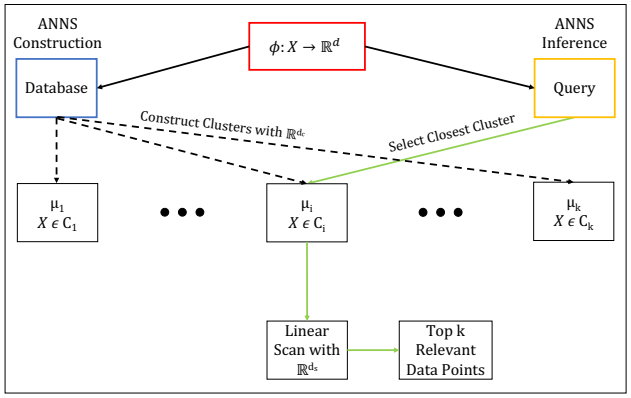


Figure 2. The schematic of inverted file index (IVF) outlaying the construction and inference phases. Adaptive representations can be utilized effectively in the decoupled components of clustering and searching for a better accuracy-compute trade-off.

Inverted File Index (IVF). In this paper, we use an Inverted File Index (IVF) (Sivic & Zisserman, 2003) ANNS data structure by default. IVF is an ANNS data structure used in web-scale search systems (Guo et al., 2020) owing to its simplicity, minimal compute overhead and high accuracy. IVF construction involves clustering (coarse quantization often through k-means) (Lloyd, 1982) on d-dimensional representation that results in an inverted file list (Witten et al., 1999) of all the data points in each cluster. During the search, d-dimensional query representation is assigned to the most relevant cluster (C_i; $i \in [k]$) by finding the closest centroid (μ_i) using an appropriate distance metric (L_2 or cosine). This is followed by an exhaustive linear search across all data points in the cluster which gives the closest NN. Lastly, IVF can scale to web-scale by utilizing a hierarchical IVF structure within each cluster (Guo et al., 2020). Figure 2 shows the high-level overview of an IVF-based ANNS system and Table 1 describes the retrieval formula for multiple variants of IVF.

Experimental setup. We evaluate the ANNS algorithms while changing the representations used for the search thus making it impossible to evaluate on the usual benchmarks (Aumüller et al., 2020). Hence we experiment with the public ImageNet-1K (Russakovsky et al., 2015) dataset on the task of image retrieval – where the goal is to retrieve images from a database (train set) belonging to the same class as the query image (validation set). We encode both the database and query set using a ResNet50 model (ϕ) (He et al., 2016) trained on ImageNet-1K. The performance of ANNS is often measured using recall@k, however, the presence of labels allows us to compute 1-NN (top-1) accuracy

where the top retrieved image should be of the same class. Top-1 accuracy is more fine-grained and correlates well with typical retrieval metrics like recall and mean average precision (mAP@k). Even though we report top-1 accuracy by default during experimentation, we discuss other metrics in Appendix C. Finally, we measure the compute overhead of ANNS using Mega FLOPS per Query (MFLOPS/Query) (see Appendix E.4).

We use the independently trained ResNet50 models with varying representation sizes ($d = [8, 16, \ldots, 2048]$) provided by Kusupati et al. (2022) alongside the MRL-ResNet50 models trained with matryoshka representation learning (MRL) for the same data dimensionalities. The RR and MR models are trained to ensure the supervised one-vs-all classification accuracy across all data dimensionalities is nearly the same – 1-NN accuracy of 2048-d RR and MR models are 71.19% and 70.97% respectively on ImageNet-1K. Independently trained models, $\phi^{\text{RR}(d)}$, output $d = [8, 16 \ldots, 2048]$ dimensional RRs while a single MRL-ResNet50 model, $\phi^{\text{MR}(d)}$, outputs a $d = 2048$-dimensional MR that contains all the 9 granularities We use the 9 exponentially separated dimensionalities to evaluate the benefits *adaptive representations* bring to ANNS design. More implementation details can be found in Appendix B and additional experiment-specific information is provided at appropriate places.

4. AdANNS – Adaptive ANNS

Construction and inference of common ANNS data structures like IVF (Sivic & Zisserman, 2003), HNSW (Malkov & Yashunin, 2020) etc., can be split into multiple components that can leverage adaptive representations. For example, IVF can be divided into its clustering and linear scan components. Each of these components can utilize representations of different dimensionalities i.e, we can use a 32-d representation for clustering, and then during the search, we can select the cluster using a 32-d representation and follow up with a more accurate linear scan using 2048-d. In principle, we can leverage different representations for different phases of ANNS which forms the ethos of AdANNS.

The challenge is to obtain adaptive representations, embeddings of varying capacities/dimensionalities but containing similar semantic information. A common way is to use post-hoc dimensionality reduction (SVD (Golub & Kahan, 1965) and random projection (Johnson, 1984)) on the highest-dimensional (2048-d) rigid representation (RR) but these are often inaccurate. While independently learned rigid representations (RRs) can be used to achieve this, it is often expensive to store copies of the database with varying dimensionalities along with multiple expensive model inferences for the query. We find that utilizing RRs of varying dimensionalities, results in less accurate adaptive ANNS indices than the matryoshka representation powered AdANNS due to the relative independence present across RRs (See Figure 3). Matryoshka Representations (MRs) solve these issues with their inherent multi-granularity and thus align well with our objectives for ANNS design with adaptive representations.

We present AdANNS, a novel design paradigm powered by the inherent flexibility of Matryoshka Representations. In this paper, we instantiate AdANNS in IVF framework (AdANNS-IVF) but it can also be easily integrated into core ANNS data structures like HNSW making AdANNS complementary to other ANNS techniques. Before delving further into AdANNS, we exposit two notions of adaptivity that can be leveraged through AdANNS in IVF.

Tale of two approximations. Clustering in IVF happens on high-d representation (2048-d) and can be approximated accurately using low-d representations (32-d). During inference, after finding the most relevant cluster for the query using 32-d, we can proceed to linear scan the data points in the cluster with a higher-dimensional representation (128-d). This can be easily enabled using Matryoshka Representations and we call it AdANNS-IVF-C (for construction) – Section 4.1. AdANNS-IVF-C caters to scenarios that require precise control over accuracy-compute trade-offs during construction, inference, and maintenance.

At the same time, when an IVF index is built on a high-dimensional (2048-d) representation inference can be done with approximate distance computation using a low-dimensional alternative. This is naturally enabled by Matryoshka Representations because of the accurate low-d representations present within for free. We call this AdANNS-IVF-D (for distance computation, see Section 4.2). Typically, product quantization (PQ) (Jegou et al., 2010) is used for cheaper distance computation in composite ANNS indices (Jaiswal et al., 2022) and remains complementary to AdANNS-IVF-D. AdANNS-IVF-D caters to scenarios that demand elasticity of search on a single database across various deployment requirements. Please see Table 1 for precise mathematical formulae corresponding to inference on IVF and both variants of AdANNS-IVF.

In this section, we describe and evaluate the two variants of AdANNS: (a) AdANNS-C which leverages adaptive representations during construction and inference, and (b) AdANNS-D which enables inference-time elastic-search with adaptive distance computation.

4.1. AdANNS-IVF-C

AdANNS-IVF-C decouples the clustering, with d_c dimensions, and the linear scan within each cluster, with d_s dimensions – setting $d_c = d_s$ results in non-adaptive regular IVF. This helps in the smooth search of design space for the

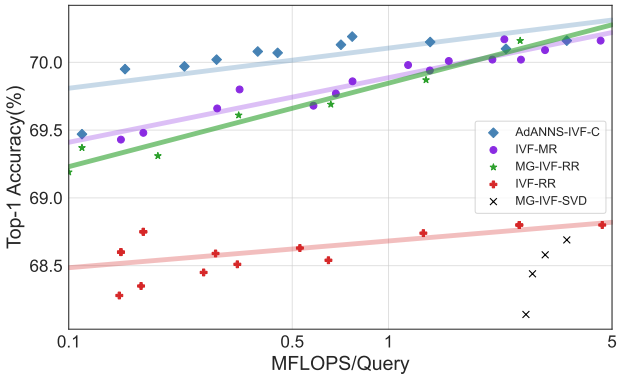


Figure 3. 1-NN accuracy using AdANNS-IVF-C compared across various rigid and adaptive baselines shows it achieves near-optimal accuracy-compute trade-off. Both adaptive variants of MR and RR significantly outperform their rigid counterparts while post-hoc compression on RR using SVD lags behind.

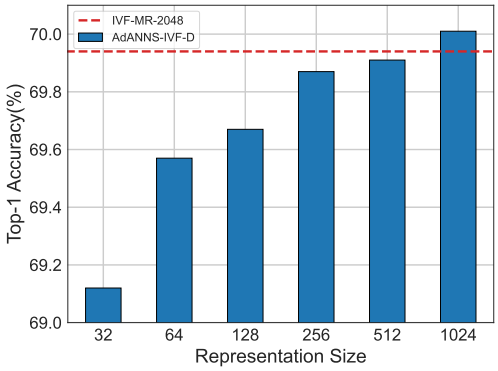


Figure 4. AdANNS-IVF-D enables adaptive inference using low-dimensional MRs on IVF built using 2048-d MRs that can be used based on deployment requirements on a shared index/database.

optimal accuracy-compute trade-off.

Experimentation on regular IVF with MRs and RRs (IVF-MR & IVF-RR) of varying dimensionalities and IVF configurations (# clusters, # probes) shows that (Figure 3) matryoshka representations result in significantly better compute-accuracy trade-off. We further studied and found that learned lower-dimensional representations offer better compute-accuracy trade-offs for IVF than higher-dimensional embeddings (see Appendix E for more results).

AdANNS utilizes d-dimensional matryoshka representation to get accurate d_c and d_s dimensional vectors with no extra compute cost. The resulting AdANNS-IVF-C provides a much better accuracy-compute trade-off (Figure 3) compared to IVF-MR, IVF-RR, and MG-IVF-RR– multi-granular IVF with rigid representations – a strong baseline that uses d_c and d_s dimensional RRs. Finally, we exhaustively search the design space of IVF by varying $d_c, d_s \in [8, 16, \ldots, 2048]$, number of clusters in $k \in [8, 16, \ldots, 2048]$. Please see Appendix F for more details.

Empirical results. Figure 3 shows that AdANNS-IVF-C outperforms the baselines across all accuracy-compute settings. AdANNS-IVF-C results in $10\times$ lower compute for the best accuracy of the extremely expensive MG-IVF-RR and non-adaptive IVF-MR. Specifically, as shown in Figure 1, AdANNS-IVF-C is up to 1.5% more accurate for the same compute and has up to $100\times$ lesser FLOPs/query than the status quo of performing ANNS on rigid representations (IVF-RR). We filter out points for sake of presentation and encourage the reader to check out Figure 13 in Appendix F for an expansive plot of all the configurations searched.

The advantage of AdANNS for construction is evident from the improvements in IVF (AdANNS-IVF-C) and can be easily extended to other ANNS structures. For example, HNSW consists of multiple layers with graphs of NSW graphs (Malkov et al., 2014) of increasing complexity. AdANNS can be adopted to HNSW, where the construction of each level can be powered by appropriate dimensionalities for an optimal accuracy-compute tradeoff. In general, AdANNS provides fine-grained control over compute overhead (storage, working memory, inference, and construction cost) during construction and inference while providing the best possible accuracy.

4.2. AdANNS-IVF-D

AdANNS-C structures cater to many specific large-scale use scenarios that need to satisfy precise resource constraints during construction as well as inference. However, in many cases, construction and storage of the indices are not the bottlenecks or the user is unable to search the design space. In these settings, AdANNS-D enables adaptive inference through accurate yet cheaper distance computation using inherent low-dimensional representations of the matryoshka representation. Akin to composite indices (Section 5.3) that use PQ vectors for cheaper distance computation, we can use the low-dimensional MR for faster distance computation on ANNS structure built with a high-dimensional MR.

Empirical results. Figure 4 shows the results of AdANNS-IVF-D using various low-dimensional representations on an IVF-MR built with 2048-d. We notice that AdANNS-IVF-D can enable up to $16\times$ faster search for a minimal ($< 0.5\%$) loss in accuracy. This enables elastic latency-aware search during inference and generalizes to other data structures when built with matryoshka representations. Lastly, adaptive inference with MR (AdANNS-D) provides an accurate end-to-end learned alternative to the existing posthoc compression methods for faster distance computation. Please see Figure 11b in Appendix E.2 for an

Table 1. Mathematical formulae of the retrieval phase across various methods built on IVF. See Section 3 for notations.

Method	Retrieval Formula during Inference
IVF-RR	$\arg\min_{j \in C_{h(q)}} \|\phi^{\text{RR}(d)}(q) - \phi^{\text{RR}(d)}(x_j)\|$, s.t. $h(q) = \arg\min_h \|\phi^{\text{RR}(d)}(q) - \mu_h^{\text{RR}(d)}\|$
IVF-MR	$\arg\min_{j \in C_{h(q)}} \|\phi^{\text{MR}(d)}(q) - \phi^{\text{MR}(d)}(x_j)\|$, s.t. $h(q) = \arg\min_h \|\phi^{\text{MR}(d)}(q) - \mu_h^{\text{MR}(d)}\|$
AdANNS-IVF-C	$\arg\min_{j \in C_{h(q)}} \|\phi^{\text{MR}(d_s)}(q) - \phi^{\text{MR}(d_s)}(x_j)\|$, s.t. $h(q) = \arg\min_h \|\phi^{\text{MR}(d_c)}(q) - \mu_h^{\text{MR}(d_c)}\|$
MG-IVF-RR	$\arg\min_{j \in C_{h(q)}} \|\phi^{\text{RR}(d_s)}(q) - \phi^{\text{RR}(d_s)}(x_j)\|$, s.t. $h(q) = \arg\min_h \|\phi^{\text{RR}(d_c)}(q) - \mu_h^{\text{RR}(d_c)}\|$
AdANNS-IVF-D	$\arg\min_{j \in C_{h(q)}} \|\phi^{\text{MR}(d)}(q)[1:\hat{d}] - \phi^{\text{MR}(d)}(x_j)[1:\hat{d}]\|$, s.t. $h(q) = \arg\min_h \|\phi^{\text{MR}(d)}(q)[1:\hat{d}] - \mu_h^{\text{MR}(d)}[1:\hat{d}]\|$
IVFPQ	$\arg\min_{j \in C_{h(q)}} \|\phi^{\text{PQ}(m,b)}(q) - \phi^{\text{PQ}(m,b)}(x_j)\|$, s.t. $h(q) = \arg\min_h \|\phi(q) - \mu_h\|$

analysis across dimensionalities on cluster selection.

We discuss the ideal scenarios for choosing between AdANNS-C and AdANNS-D in Section 6.1. While we have utilized real-valued representations for distance computation in AdANNS, this is complementary to the ubiquitous strategy of utilizing product quantization for further speeding-up search. The complementary relation between PQ and MRs along with IVFPQ can be found in Section 5. We also provide psuedocode for AdANNS in Appendix A.

5. Further Analysis on ANNS Components

State-of-the-art ANNS pipelines (Johnson et al., 2019; Jayaram Subramanya et al., 2019) use two other key techniques alongside IVF: (a) Product Quantization (PQ) and (b) Hierarchical Navigable Small World graphs (HNSW). Naturally, a question arises if AdANNS can be combined with such ANNS components. While we theoretically argue about the complementary nature of AdANNS to PQ and HNSW (Section 4), here we provide preliminary results in that direction. We show that: (a) PQ-based approximate distance computation is complementary to MRs and can be easily added to AdANNS-D; (b) MRs are better than RRs when using HNSW and PQ can be leveraged on top of them to find optimal accuracy-compute trade-off. For simplicity of exposition, and highlighting the key aspects of AdANNS, our primary adaptation and experiments are on IVF which is another state-of-the-art ANNS data structure.

5.1. Product Quantization

Product quantization (PQ) (Jegou et al., 2010) works by splitting a d-dimensional real vector into m sub-vectors and quantizing each sub-vector with an independent 2^b length codebook across the database. After PQ each d-dimensional vector can be represented by a compact $m \times b$ bit vector – we fix $b = 8$ making the vectors m bytes long. The generality of PQ encompasses scalar/vector quantization (Gray, 1984; Lloyd, 1982) as special cases. While, significant work has been done to improve PQ further (Ge et al., 2013; Chen et al., 2020), we only experiment with standard PQ as the improvements are complementary (See Appendix D).

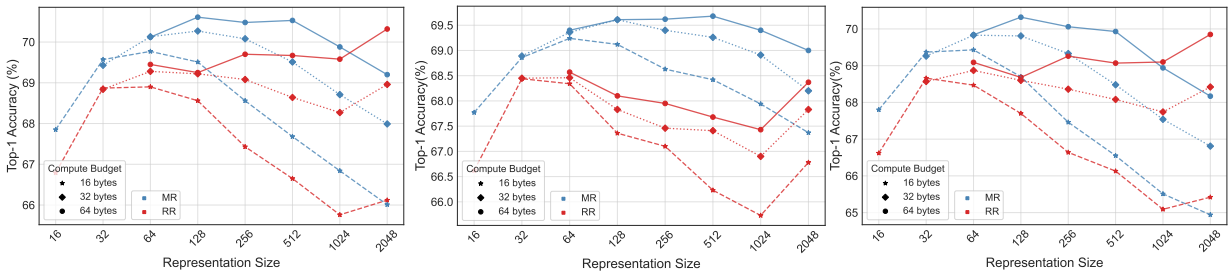
Figure 5a shows that often lower-dimensional representations provide better compute-vs-accuracy trade-off for PQ. For 32-byte PQ, a deployment default, it is better to use a 128-d representation instead of the full 2048-d. At the same time, we also find that MRs are more aligned for PQ owing to the information packing compared to RRs. For a fixed budget of PQ, the accuracy steadily increases till a specific dimensionality and then deteriorates. Finally, we also note that on ImageNet-1K, a net 2-bit quantization seems to work the best (ie., 32-d is the best for 8-byte PQ). See Appendix D for evaluation across more PQ budgets that further strengthen the observations. This shows that PQ is complementary to low-dimensional MRs and can be combined with AdANNS-D to further speed-up inference.

5.2. HNSW

Hierarchical Navigable Small World graph (HNSW) is a state-of-the-art ANNS index with more memory overhead compared to IVF. HNSW is often the go-to out-of-the-box choice for high-accuracy ANNS (Beaumont, 2021). We investigate the compute-accuracy trade-off of the default HNSW index (Johnson et al., 2019) built on varying capacity MRs and RRs. We find that HNSW is slightly more accurate than IVF but comes with a large index memory overhead. Figure 14 in Appendix G.1 shows that the accuracy gain is marginal after certain data dimensionality for HNSW, but the compute and memory cost grow linearly with d – making low-dimensional representation better for the accuracy-compute trade-off. For example, the accuracy gain from 128-d to 2048-d MR is marginal while the overhead increases $16\times$. Similar to the observations in PQ and IVF, MRs outperform RRs even for graph-based while showing that MRs are complementary to HNSW as well.

5.3. Composite Indices – IVFPQ & HNSWPQ

Composite indices represent the existing adaptivity in ANNS pipelines. An ANNS data structure is often built on a high-dimensional RR, but can not afford distance computation with the same. In these cases, we use cheaper distance computation through dimensionality reduction –



(a) PQ (b) IVFPQ (c) HNSWPQ

Figure 5. Evaluation of accuracy-compute trade-off for (a) standard PQ, (b) IVFPQ and (c) HNSWPQ indices built on varying dimensional MRs and RRs. We investigate 16, 32, and 64-byte PQ budgets and find that the highest-dimensional representation need not be the most aligned for quantization owing to hardness of clustering.

typically through PQ (Jayaram Subramanya et al., 2019). Composite indices trade off the accurate construction of the expensive structure with efficient inference.

Figures 5b and 5c show that even for composite indices (IVFPQ and HNSWPQ), lower dimensions result in a better accuracy-compute trade-off as they can cluster and quantize better compared to higher-dimensions. The trends of IVFPQ and HNSWPQ follow that of PQ where MRs are consistently better than RRs and the best accuracy peaks at a much lower dimension than 2048-d.

A key thing to note is that the AdANNS-D proposed for adaptive inference in Section 4.2 is a generalized variant of composite indices where data structure construction and inference happen with representations of different capacities. While AdANNS-D uses low-d MRs for distance computation, as observed here they also can result in better PQ vectors for further speed-ups. The design space of composite indices blows up when using adaptive representations hence using MRs for ANNS design and deployment can result in a smooth and flexible inference time adoption with minimal design search overhead.

6. Discussion

6.1. AdANNS-C vs. AdANNS-D

The two alternatives, AdANNS-C, and AdANNS-D within the AdANNS paradigm rely on bringing adaptivity to the construction and inference of ANNS indices respectively. Figure 6 shows that for a given compute budget, AdANNS-C is better than AdANNS-D due to the explicit control during the building of the ANNS structure. However, these methods are applicable in specific scenarios of deployment. AdANNS-C is more tailored for precise deployment constraints that require maximum possible accuracy for a given budget. Obtaining optimal AdANNS-C relies on a relatively expensive design space search but delivers indices that fit the storage, memory, compute, and accuracy constraints all at once. On the other hand AdANNS-D does not require a precisely built ANNS index but can enable compute-aware search during inference. AdANNS-D achieves this by using low-dimensional MRs for distance computation based on compute budget. AdANNS-D is a great choice for setups that can afford only one single database/index but need to cater to varying deployment constraints – e.g, one task requires 70% accuracy while another task has a strict compute cut-off at 1 MFLOPS/query.

6.2. Difficulty of NN Search

Relative contrast (C_r) (He et al., 2012) intuitively measures the difficulty of nearest neighbour search on a given database. C_r is lower bounded by 1 and C_r is inversely correlated to the difficulty of the nearest neighbour search. Figure 7 shows that MRs have better C_r than RRs across dimensionalities further supporting that matryoshka representations are more aligned (easier) for NN search than existing rigid representations for the same accuracy. This translates to higher accuracy for the same NN search cost as observed in our experiments (Appendix E). Lastly, Figure 7 also shows that C_r increases as the dimensionality

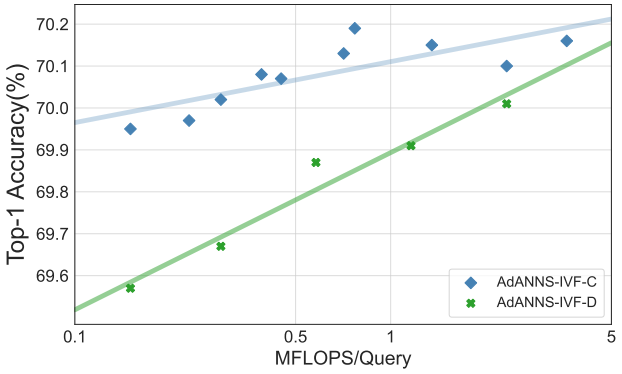


Figure 6. Compute-accuracy trade-off for the two variants of AdANNS, AdANNS-C & AdANNS-D on IVF. AdANNS-C is better than AdANNS-D due to the control during construction.

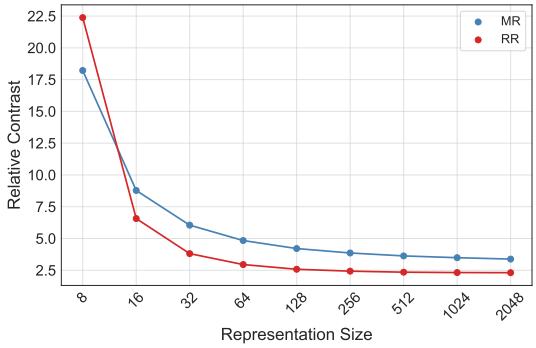


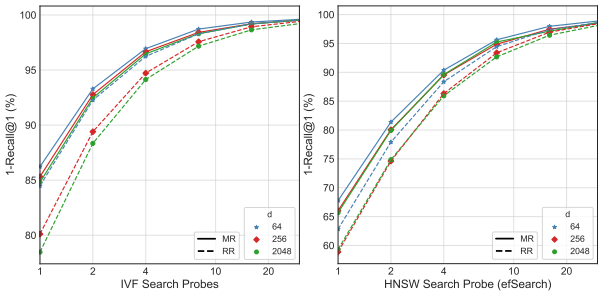
Figure 7. Relative contrast of varying capacity MRs and RRs on ImageNet-1K corroborating the findings of He et al. (2012).

decreases – owing to distance concentration in high dimensional spaces (Blum et al., 2020; He et al., 2012). More details about this experiment can be found in Appendix G.2.

6.3. Recall Score Analysis

A labeled dataset like ImageNet-1K lets us compute top-1 accuracies which often correlate with overall recall metrics. However, in larger noisy datasets the quality of ANNS is evaluated using the recall of the "true" NN across search complexities – measured using k-Recall@N which denotes the recall of k true NN when N datapoints are retrieved. While this metric makes sense for evaluating ANNS algorithms on a RR, it needs to be combined with top-1 accuracy for understanding the differences across various learned representations. For a similar top-1 accuracy, a better recall-score plot implies easier searchability for ANNS.

Figure 8 shows that for a similar top-1 accuracy, lower-dimensional representations have better 1-Recall@1 across search complexities for IVF and HNSW on ImageNet-1K. We also observe similar results for the easier 40-Recall@2048 metric as shown in Figure 15 in Appendix G.1. Across the board, MRs have higher recall scores and top-1 accuracy pointing to the better suitability of matryoshka representations for ANNS. Lastly, we also experimented on ImageNet-4K (Kusupati et al., 2022), a $4\times$ larger benchmark than ImageNet, and found similar results even at a much larger scale (See Appendix G.1).



(a) IVF (b) HNSW

Figure 8. Recall score plots with 1-Recall@1 for ImageNet-1K using IVF and HNSW on MRs and RRs of 64, 256 & 2048-d.

6.4. Clustering Distributions – MRs vs. RRs

We also investigate the potential deviation in clustering distributions for MRs across dimensionalities compared to RRs. Unlike the RRs where the information is uniformly diffused across dimensions (Soudry et al., 2018), MRs have hierarchical information packing. Figure 12 in Appendix E.3 shows that matryoshka representations result in clusters similar (measured by total variation distance (Levin & Peres, 2017)) to that of rigid representations and do not result in any unusual artifacts.

6.5. Robustness

Figure 10 in Appendix E shows MRs continue to be better than RRs even for out-of-distribution (OOD) image queries (ImageNetV2 (Recht et al., 2019)) using ANNS. It also shows that the highest data dimensionality need not always be the most robust which is further supported by the higher recall using lower dimensions. Further, we show that adaptive inference on fixed IVF structure holds true even for OOD queries and enables up to $16\times$ (Table 3 in Appendix E.1) compute gains for similar accuracy.

6.6. Generality across Encoders

We also find that our observations on better alignment of MRs for NN search hold across neural network architectures, ResNet18/34/101 (He et al., 2016). IVF-MR consistently has higher accuracy compared to IVF-RR across dimensionalities despite having similar accuracies with exhaustive NN search. Adaptive representations like MRs also allow for easy searching of optimal dimensionality for accuracy-compute trade-offs across all neural architectures. Appendix G.3 delves deep into the experimentation done using various neural architectures on ImageNet-1K.

7. Conclusions

ANNS methods typically use a fixed (high-dimensional) representation for both query and the document. We proposed a novel approach based on usage of adaptive representations for different phases of ANNS pipelines. Our AdANNS paradigm leverages the inherent adaptivity of matryoshka representations (Kusupati et al., 2022) during the construction and inference of ANNS structures. AdANNS's variants achieves SOTA accuracy-compute trade-off, while also ensuring compute-aware elastic search. We also found that AdANNS is generalizable and complementary to other ANNS techniques. Finally, we make a case for adaptive representations to be used in semantic search.

Acknowledgments

We are grateful to Kaifeng Chen, Venkata Sailesh Sanampudi, Gantavya Bhatt and Matthew Wallingford for helpful discussions and feedback. Aditya Kusupati also thanks Tom Duerig and Rahul Sukthankar for their support. Part of the paper's large-scale experimentation is supported through a research GCP credit award from Google Cloud and Google Research. Sham Kakade acknowledges funding from the ONR award N00014-22-1-2377 and NSF award CCF-2212841. Ali Farhadi acknowledges funding from the NSF awards IIS 1652052, IIS 17303166, DARPA N66001-19-2-4031, DARPA W911NF-15-1-0543, and gifts from Allen Institute for Artificial Intelligence and Google.

References

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

Beaumont, R. *Clip Retrieval*, 2021. URL https://github.com/rom1504/clip-retrieval.

Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Bernhardsson, E. *Annoy: Approximate Nearest Neighbors in C++/Python*, 2018. URL https://pypi.org/project/annoy/. Python package version 1.13.0.

Blum, A., Hopcroft, J., and Kannan, R. *Foundations of data science*. Cambridge University Press, 2020.

Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

Cai, D. A revisit of hashing algorithms for approximate nearest neighbor search. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2337–2348, 2021. doi: 10.1109/TKDE.2019.2953897.

Chen, T., Li, L., and Sun, Y. Differentiable product quantization for end-to-end embedding compression. In *International Conference on Machine Learning*, pp. 1617–1626. PMLR, 2020.

Clarkson, K. L. An algorithm for approximate closest-point queries. In *Proceedings of the tenth annual symposium on Computational geometry*, pp. 160–164, 1994.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pp. 253–262, 2004.

Dean, J. Challenges in building large-scale information retrieval systems. In *Keynote of the 2nd ACM International Conference on Web Search and Data Mining (WSDM)*, volume 10, 2009.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

Ge, T., He, K., Ke, Q., and Sun, J. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2946–2953, 2013.

Golub, G. and Kahan, W. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

Gray, R. Vector quantization. *IEEE Assp Magazine*, 1(2): 4–29, 1984.

Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., and Kumar, S. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pp. 3887–3896. PMLR, 2020.

Gupta, N., Chen, P. H., Yu, H.-F., Hsieh, C.-J., and Dhillon, I. S. End-to-end learning to index and search in large output spaces. *arXiv preprint arXiv:2210.08410*, 2022.

He, J., Kumar, S., and Chang, S.-F. On the difficulty of nearest neighbor search. In *International Conference on Machine Learning (ICML)*, 2012.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Indyk, P. and Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.

Jaiswal, S., Krishnaswamy, R., Garg, A., Simhadri, H. V., and Agrawal, S. Ood-diskann: Efficient and scalable graph anns for out-of-distribution queries. *arXiv preprint arXiv:2211.12850*, 2022.

Jayaram Subramanya, S., Devvrit, F., Simhadri, H. V., Krishnawamy, R., and Kadekodi, R. Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems*, 32, 2019.

Jegou, H., Douze, M., and Schmid, C. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7 (3):535–547, 2019.

Johnson, W. B. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26:189–206, 1984.

Jolliffe, I. T. and Cadima, J. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

Kraska, T., Beutel, A., Chi, E. H., Dean, J., and Polyzotis, N. The case for learned index structures. In *Proceedings of the 2018 international conference on management of data*, pp. 489–504, 2018.

Kusupati, A., Wallingford, M., Ramanujan, V., Somani, R., Park, J. S., Pillutla, K., Jain, P., Kakade, S., and Farhadi, A. Llc: Accurate, multi-purpose learnt low-dimensional binary codes. *Advances in Neural Information Processing Systems*, 34:23900–23913, 2021.

Kusupati, A., Bhatt, G., Rege, A., Wallingford, M., Sinha, A., Ramanujan, V., Howard-Snyder, W., Chen, K., Kakade, S., Jain, P., and Farhadi, A. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, December 2022.

Levin, D. A. and Peres, Y. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Li, W., Zhang, Y., Sun, Y., Wang, W., Zhang, W., and Lin, X. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Malkov, Y., Ponomarenko, A., Logvinov, A., and Krylov, V. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems*, 45: 61–68, 2014.

Malkov, Y. A. and Yashunin, D. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 42(04):824–836, 2020.

Nayak, P. Understanding searches better than ever before. *Google AI Blog*, 2019. URL https://blog.google/products/search/search-language-understanding-bert/.

Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J., Yuan, Q., Tezak, N., Kim, J. W., Hallacy, C., et al. Text and code embeddings by contrastive pretraining. *arXiv preprint arXiv:2201.10005*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Salakhutdinov, R. and Hinton, G. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7): 969–978, 2009.

Simhadri, H. V., Williams, G., Aumüller, M., Douze, M., Babenko, A., Baranchuk, D., Chen, Q., Hosseini, L., Krishnaswamy, R., Srinivasa, G., et al. Results of the neurips'21 challenge on billion-scale approximate nearest neighbor search. *arXiv preprint arXiv:2205.03763*, 2022.

Sivic, J. and Zisserman, A. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pp. 1470–1470. IEEE Computer Society, 2003.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Waldburger, C. As search needs evolve, microsoft makes ai tools for better search available to researchers and developers. *Microsoft AI Blog*, 2019. URL https://blogs.microsoft.com/ai/bing-vector-search/.

Wang, M., Xu, X., Yue, Q., and Wang, Y. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 14(11):1964–1978, 2021.

Weber, R., Schek, H.-J., and Blott, S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pp. 194–205, 1998.

Witten, I. H., Witten, I. H., Moffat, A., Bell, T. C., Bell, T. C., Fox, E., and Bell, T. C. *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.

A. AdANNS Code

Algorithm 1 AdANNS Psuedocode

```
# Index database to construct clusters and build inverted file system

def adannsConstruction(database, d_cluster, num_clusters):
    # Slice database with cluster construction dim (d_cluster)
    xb = database[:d_cluster]
    cluster_centroids = constructClusters(xb, num_clusters)

    return cluster_centroids


def adannsInference(queries, centroids, d_shortlist, d_search, num_probes, k):
    # Slice queries and centroids with cluster shortlist dim (d_shortlist)
    xq = queries[:d_shortlist]
    xc = centroids[:d_shortlist]

    for q in queries:
        # compute distance of query from each cluster centroid
        candidate_distances = computeDistances(q, xc)
        # sort cluster candidates by distance and choose small number to probe
        cluster_candidates = sortAscending(candidate_distances)[:num_probes]
        database_candidates = getClusterMembers(cluster_candidates)
        # Linear Scan all shortlisted clusters with search dim (d_search)
        k_nearest_neighbors[q] = linearScan(q, database_candidates, d_search, k)

    return k_nearest_neighbors
```

B. Implementation Details

A bulk of our experimentation was carried out via Faiss (Johnson et al., 2019), a library for efficient similarity search and clustering. AdANNS was implemented from scratch due to difficulty in decoupling clustering and linear scan with Faiss, code available: https://github.com/RAIVNLab/AdANNS. All ANNS experiments (HNSW, HNSWPQ, IVFPQ) were run on an Intel Xeon 2.20GHz CPU with 12 cores. Clustering (IVF-MR and IVF-RR), Adaptive Retrieval (AdANNS-IVF and MG-IVF-RR) and Exact Search experiments were run with CUDA 11.0 on a A100-SXM4 NVIDIA GPU with 40G RAM.

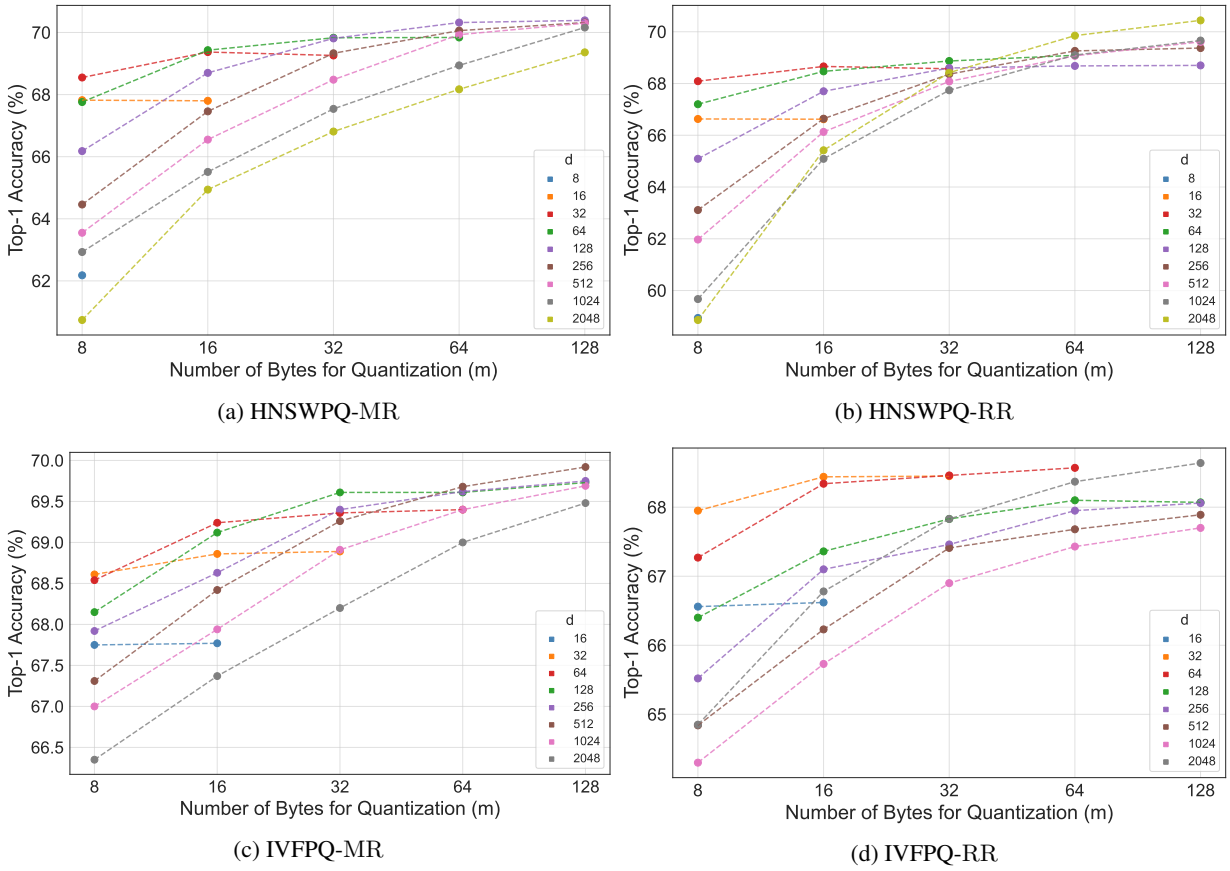
C. Evaluation Metrics

In this work, we primarily use top-1 accuracy (i.e. 1-Nearest Neighbor), recall@k, corrected mean average precision (mAP@k) (Kusupati et al., 2021) and k-Recall@N, which are defined over all queries Q over indexed database of size N_D as:

$$\text{top-1} = \frac{\sum_Q \text{correct\_pred@1}}{|Q|}$$

$$\text{Recall@k} = \frac{\sum_Q \text{correct\_pred@}k}{|Q|} * \frac{\text{num\_classes}}{|N_D|}$$

where correct_pred@k is the number of k-NN with correctly predicted labels for a given query. As noted in Section 6.3, k-Recall@N is the overlap between k exact search nearest neighbors (which are considered ground truth) and the top N retrieved documents. As Faiss (Johnson et al., 2019) supports a maximum of 2048-NN while searching the indexed database, we report 40-Recall@2048 in Figures 15 and 16. Also note that for ImageNet-1K, which constitutes a bulk of the experimentation in this work, $|Q| = 50000$, $|N_D| = 1281167$ and num_classes$= 1000$. For ImageNetv2, $|Q| = 10000$ and num_classes$= 1000$, and for ImageNet-4K, $|Q| = 210100$, $|N_D| = 4202000$ and num_classes$= 4202$.



(a) HNSWPQ-MR (b) HNSWPQ-RR

(c) IVFPQ-MR (d) IVFPQ-RR

Figure 9. Top-1 Accuracy of MR compared to RR baseline models on ImageNet-1K with IVFPQ and HNSWPQ.

D. Product Quantization

In this section, we discuss additional product quantization behavior of MR. We perform an exhaustive study of compression across embedding dimensionalities d for composite PQm$\times$b indices with both IVF and HNSW, i.e. IVFPQ and HNSWPQ, as seen in Figure 9. It is evident from these results that:

1. For a fixed m-byte compression provided by PQ with m sub-quantizers and $b = 8$ bit precision, the highest accuracy does not come from the highest embedding dimensionality d. This trend is ubiquitous with Matryoshka Representations (MRs) in both graph (HNSW) and tree (IVF) search space partitioning structures across all embedding dims d, and for baseline RR for $m \leq 32$. It is interesting to note that $d = 2048$ has the *worst* Top-1 accuracy across all compression sizes from 8 to 128 bytes with MRs.
2. In IVFPQ, MRs are on average 1.5% more accurate, and at max 3.3% more accurate than their baseline RR counterparts for all (m, d) PQ compression tuples.
3. In HNSWPQ, MRs are on average 1% more accurate, and at max 3.2% more accurate than their baseline RR counterparts for all (m, d) PQ compression tuples with $d \leq 1024$.
4. Note that we indirectly evaluate Scalar Quantization (SQ) (Gray, 1984) at all PQm where $m = d$. As seen in Figure 9, SQ is never the best configuration for fixed compression. MR-SQ is on average 1.5% more accurate than RR-SQ for both IVFSQ and HNSWSQ.

We also explore the potential gains offered by Optimized Product Quantization (OPQ) (Ge et al., 2013) in Table 2. The space rotation and dimensionality permutation performed by OPQ alongside sub-vector quantization offer slight gains in top-1 and mAP@10 accuracy with IVF clustering at $d \in \{8, 16, 32, 64, 256\}$, and more substantial recall@100 gains at all $d \geq 16$ (See Appendix C for metric definitions).

Table 2. Comparison of IVFPQ-MR with OPQ-MR for fixed $d = 2048$ across quantized compression $m \in \{1, 2, 4, ..., 2048\}$.

Config		IVFPQ-MR			OPQ-MR		
d	m	Top-1	mAP@10	R@100	Top-1	mAP@10	R@100
2048	1	**66.69**	**61.76**	**5.00**	64.81	59.66	4.97
	2	**67.02**	**61.96**	5.00	66.03	60.97	**5.02**
	4	**67.19**	**62.28**	5.01	67.13	61.96	**5.06**
	8	67.84	62.67	5.01	**67.90**	**62.72**	**5.09**
	16	68.49	63.19	5.02	**68.89**	**63.49**	**5.12**
	32	69.03	63.68	5.03	**69.35**	**64.08**	**5.15**
	64	69.38	64.06	5.03	**69.59**	**64.36**	**5.16**
	128	69.68	64.33	5.04	**69.77**	**64.46**	**5.17**
	256	69.80	**64.59**	5.05	**69.90**	64.53	**5.17**
	512	**70.06**	**64.83**	5.05	69.87	64.56	**5.17**
	1024	**70.17**	**64.95**	5.06	69.89	64.60	**5.17**
	2048	**70.10**	**64.98**	5.06	69.87	64.62	**5.17**

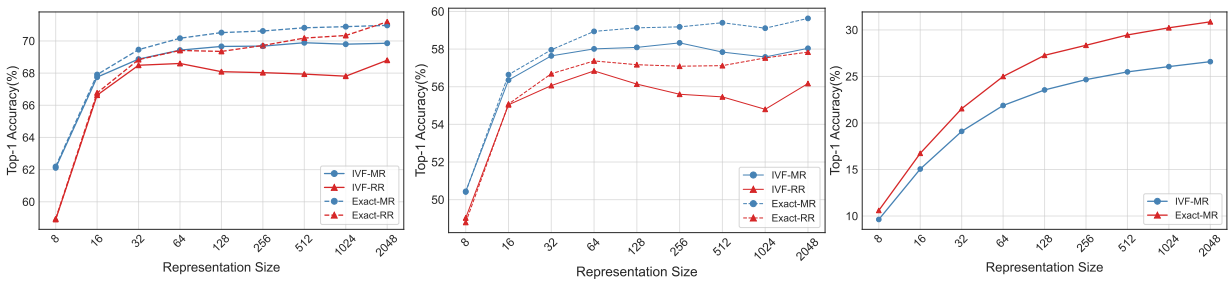
E. IVF

Inverted file index (IVF) (Sivic & Zisserman, 2003) is a simple yet powerful ANNS data structure used in web-scale search systems (Guo et al., 2020). IVF construction involves clustering (coarse quantization often through k-means) (Lloyd, 1982) on d-dimensional representation that results in an inverted file list (Witten et al., 1999) of all the data points in each cluster. During search, the d-dimensional query representation is first assigned to the closest clusters (# probes, typically set to 1) and then an exhaustive linear scan happens within each cluster to obtain the nearest neighbors.

Our proposed adaptive variant of IVF, AdANNS-IVF-C, decouples the clustering, with d_c dimensions, and the linear scan within each cluster, with d_s dimensions – setting $d_c = d_s$ results in non-adaptive vanilla IVF. This helps in the smooth search of design space for the optimal accuracy-compute trade-off. A naive instantiation yet strong baseline would be to use explicitly trained d_c and d_s dimensional rigid representations (called MG-IVF-RR, for multi-granular IVF with rigid representations). We also examine the setting of adaptively choosing low-dimensional MR to linear scan the shortlisted clusters built with high-dimensional MR, i.e. AdANNS-IVF-D, as seen in Table 3. We discuss the inference compute for these settings in Appendix E.4.

E.1. ImageNetV2 and ImageNet-4K

As shown in Figure 10, we examined the clustering capabilities of MRs on both in-distribution queries via ImageNet-1K and out-of-distribution queries via ImageNetV2 (Recht et al., 2019), as well as on larger-scale ImageNet-4K (Kusupati et al., 2022). For ID queries on ImageNet-1K (Figure 10a), IVF-MR is at least as accurate as Exact-RR for $d \leq 256$ with a single search probe, demonstrating the quality of in-distribution low-d clustering with MR. On OOD queries



(a) ImageNet-1K (b) ImageNetV2 (c) ImageNet-4K

Figure 10. Top-1 Accuracy variation of IVF-MR of ImageNet 1K, ImageNetV2 and ImageNet-4K. RR baselines are omitted on ImageNet-4K due to high compute cost.

Table 3. Top-1 Accuracy of AdANNS-IVF-D on out-of-distribution queries from ImageNetV2 compared to both IVF and Exact Search with MR and RR embeddings. Note that for AdANNS-IVF-D, the dimensionality used to build clusters $d_c = 2048$.

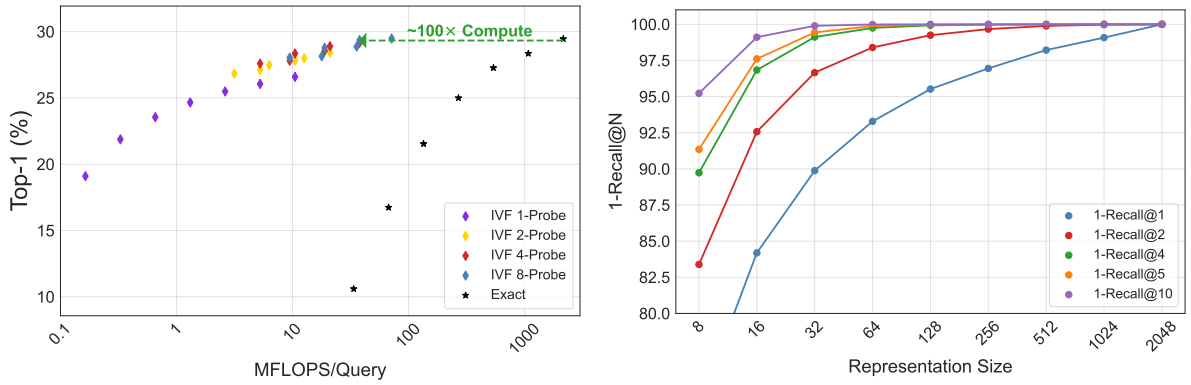
d	AdANNS-IVF-C	IVF-MR	Exact-MR	IVF-RR	Exact-RR
8	**53.51**	50.44	50.41	49.03	48.79
16	**57.32**	56.35	56.64	55.04	55.08
32	57.32	57.64	**57.96**	56.06	56.69
64	57.85	58.01	**58.94**	56.84	57.37
128	58.02	58.09	**59.13**	56.14	57.17
256	58.01	58.33	**59.18**	55.60	57.09
512	58.03	57.84	**59.40**	55.46	57.12
1024	57.66	57.58	**59.11**	54.80	57.53
2048	58.04	58.04	**59.63**	56.17	57.84

(Figure 10b), we observe that IVF-MR is on average 2% more robust than IVF-RR across all cluster construction and linear scan dimensionalities d. It is also notable that clustering with MRs followed by linear scan with # probes $= 1$ is more robust than exact search with RR embeddings across all $d \leq 2048$, indicating the adaptability of MRs to distribution shifts during inference. As seen in Table 3, on ImageNetV2 AdANNS-IVF-D is the best configuration for $d \leq 16$, and is similarly accurate to IVF-MR at all other d. AdANNS-IVF-D with $d = 128$ is able to match its own accuracy with $d = 2048$, a $16\times$ compute gain during inference. This demonstrates the potential of AdANNS to adaptively search pre-indexed clustering structures.

On 4-million scale ImageNet-4K (Figure 10c), we observe similar accuracy trends of IVF-MR compared to Exact-MR as in ImageNet-1K (Figure 10a) and ImageNetV2 (Figure 10b). We omit baseline IVF-RR and Exact-RR experiments due to high compute cost at larger scale.

E.2. Ablations

As seen in Figure 11a, IVF-MR can match the accuracy of Exact Search on ImageNet-4K with $\sim 100\times$ less compute. We also explored the capability of MRs at retrieving cluster centroids with low-d compared to a ground truth of 2048-d with k-Recall@N, as seen in Figure 11b. MRs were able to saturate to near-perfect 1-Recall@N for $d \geq 32$ and $N \geq 4$, indicating the potential of AdANNS at matching exact search performance with less than 10 search probes n_p.



(a) 4K Search Probes (n_p)

(b) Centroid Recall

Figure 11. Ablations on IVF-MR Clustering: a) Analysis of accuracy-compute tradeoff with increasing IVF-MR search probes n_p on ImageNet-4K compared to Exact-MR and b) k-Recall@N on ImageNet-1K cluster centroids across representation sizes d. Cluster centroids retrieved with highest embedding dim $d = 2048$ were considered ground-truth centroids.

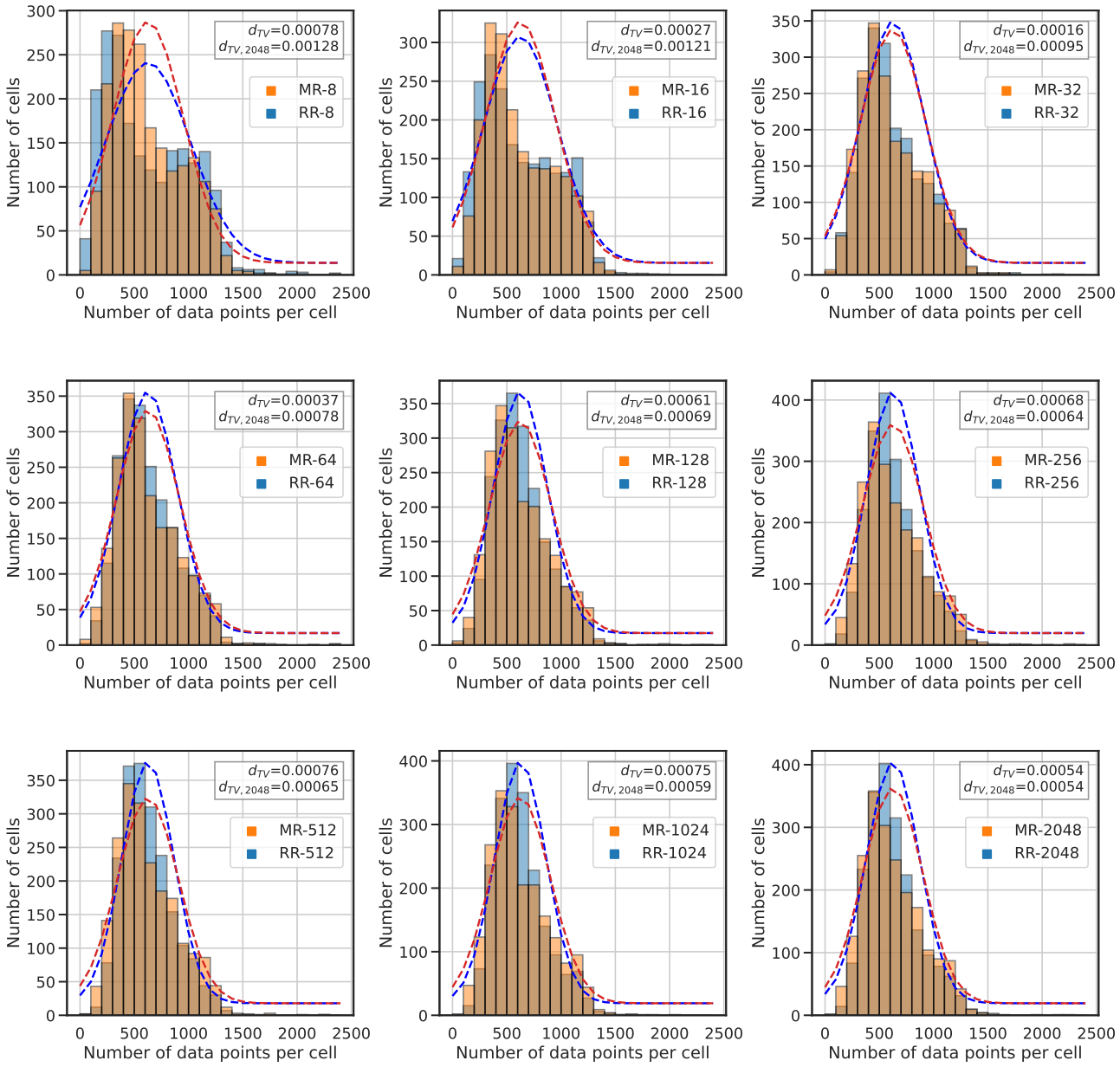


Figure 12. Clustering distributions for IVF-MR and IVF-RR across embedding dimensionality d on ImageNet-1K. An IVF-MR and IVF-RR clustered with $d = 16$ embeddings is denoted by MR-16 and RR-16 respectively.

E.3. Clustering Distribution

We examined the distribution of learnt clusters across embedding dimensionalities d for both MR and RR models, as seen in Figure 12. We observe IVF-MR to have less variance than IVF-RR at $d \in \{8, 16\}$, and slightly higher variance for $d \geq 32$, while IVF-MR outperforms IVF-RR in top-1 across all d (Figure 10a). This indicates that although MR learns clusters that are less uniformly distributed than RR at high d, the quality of learnt clustering is superior to RR across all d. Note that a uniform distribution is N/k data points per cluster, i.e. ~ 1250 for ImageNet-1K with $k = 1024$. We quantitatively evaluate the proximity of the MR and RR clustering distributions with Total Variation Distance (Levin & Peres, 2017), which is

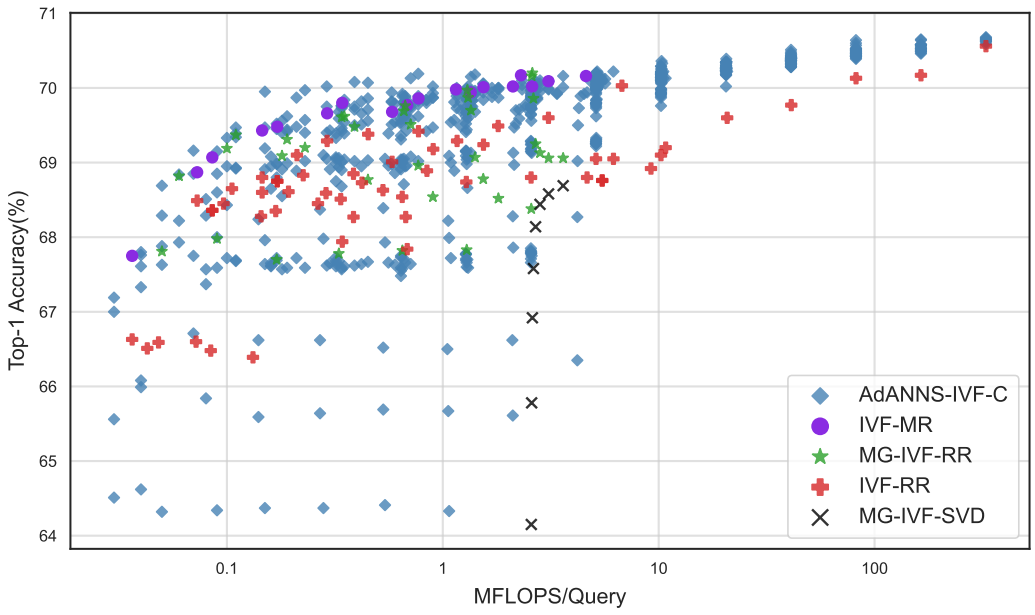


Figure 13. Top-1 accuracy vs compute cost per query of AdANNS-IVF-C compared to IVF-MR, IVF-RR and MG-IVF-RR baselines on ImageNet-1K.

defined over two discrete probability distributions p, q over $[n]$ as follows:

$$d_{TV}(p, q) = \frac{1}{2} \sum_{i \in [n]} |p_i - q_i|$$

We also compute $d_{TV,2048}(\text{MR-d}) = d_{TV}(\text{MR-d}, \text{RR-2048})$, which evaluates the total variation distance of a given low-d MR from high-d RR-2048. We observe a monotonically decreasing $d_{TV,2048}$ with increasing d, which demonstrates that MR clustering distributions get closer to RR-2048 as we increase the embedding dimensionality d. We observe in Figure 12 that $d_{TV}(\text{MR-d}, \text{RR-d}) \sim 7e-4$ for $d \in \{8, 256, ..., 2048\}$ and $\sim 3e-4$ for $d \in \{16, 32, 64\}$. These findings agree with the top-1 improvement of MR over RR as shown in Figure 10a, where there are smaller improvements for $d \in \{16, 32, 64\}$ (smaller d_{TV}) and larger improvements for $d \in \{8, 256, ..., 2048\}$ (larger d_{TV}). These results demonstrate a correlation between top-1 performance of IVF-MR and the quality of clusters learnt with MR.

E.4. Inference Compute

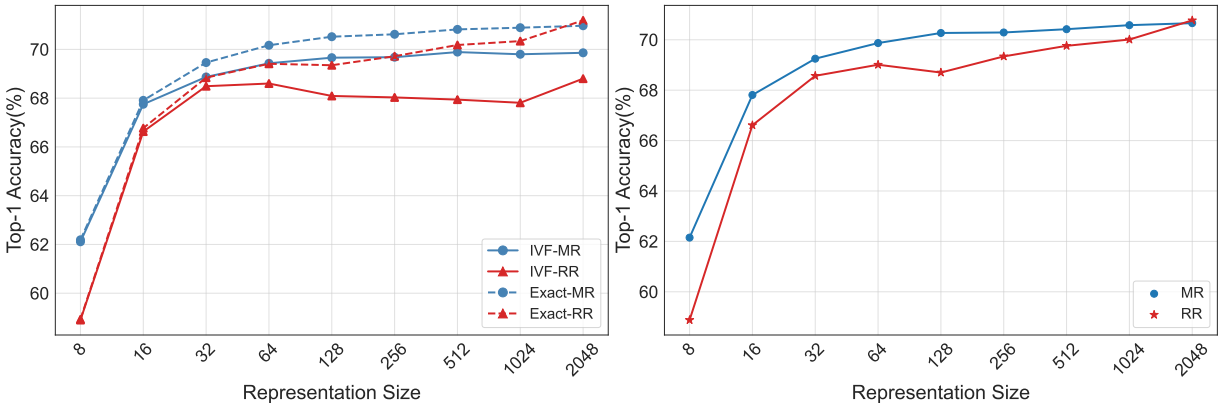
We evaluate inference compute for IVF in MegaFLOPS per query (MFLOPS/query) as shown in Figures 1, 3, and 6 as follows:

$$C = d_s k + \frac{n_p d_s N_D}{k}$$

where d_c is the **cluster** construction embedding dimensionality, d_s is the embedding dim used for linear **scan** within each **probed** cluster, which is controlled by # of search probes n_p. Finally, k is the number of clusters $|C_i|$ indexed over database of size N_D. The default setting in this work unless otherwise mentioned is $n_p = 1$, $k = 1024$, $N_D = 1281167$ (ImageNet-1K trainset). Vanilla IVF supports only $d_c = d_s$, while AdANNS-IVF-C provides flexibility via decoupling clustering and search (Section 4). AdANNS-IVF-D is a special case of AdANNS-IVF-C with the flexibility restricted to inference, i.e. d_c is a fixed high-dimensional MR.

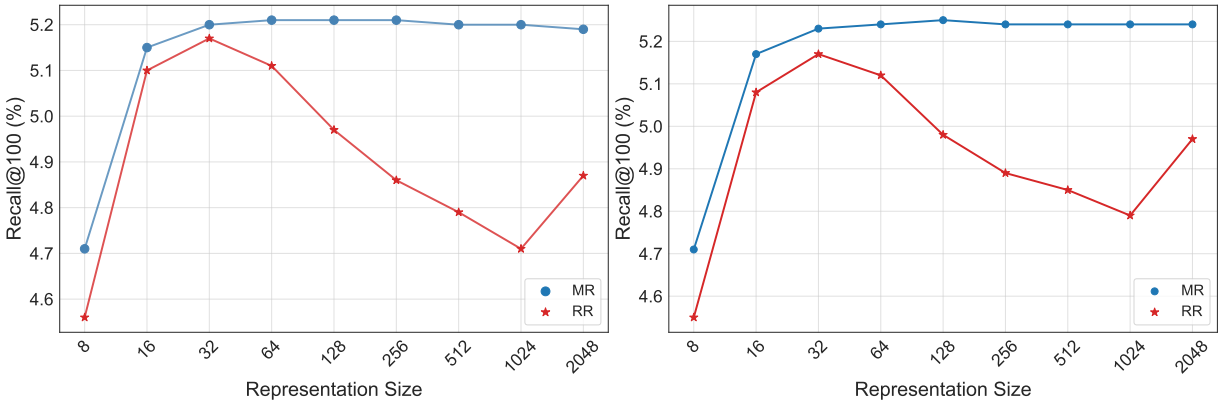
F. AdANNS-IVF-C

As seen in Figure 13, AdANNS-IVF-C provides pareto-optimal compute-accuracy tradeoff across inference compute. This figure is a more exhaustive indication of AdANNS-IVF-C behavior compared to baselines than Figures 1 and 3. AdANNS-IVF-C is evaluated for all possible tuples of $d_c, d_s, k = |C| \in \{8, 16, ..., 2048\}$. MG-IVF-RR configurations



(a) IVF top-1 accuracy

(b) HNSW top-1 accuracy



(c) IVF Recall@100

(d) HNSW Recall@100

Figure 14. Top-1 and Recall@100 of MR compared to RR baselines on ImageNet-1K with HNSW and IVF.

are evaluated for $d_c \in \{8, ..., d_s\}$, $d_s \in \{32, ..., 2048\}$ and $k = 1024$ clusters. A study over additional k values is omitted due to high compute cost. Finally, IVF-MR and IVF-RR configurations are evaluated for $d_c = d_s \in \{8, 16, ..., 2048\}$ and $k \in \{256, ..., 8192\}$. Note that for a fair comparison, we use $n_p = 1$ across all configurations.

G. Ablations

G.1. IVF and HNSW

We observe that the top-1 improvements shown by MR over RR also extend to recall@100, as shown in Figure 14. In this section we also examine the variation of k-Recall@N with search probes in more detail. For IVF, search probes represent the number of clusters shortlisted for linear scan during inference. For HNSW, search quality is controlled by the $efSearch$ parameter (Malkov & Yashunin, 2020), which represents the closest neighbors to query q at level l_c of the graph and is analogous to number of search probes in IVF. As seen in Figure 15, general trends show a) an intuitive increase in recall with increasing search probes n_p) for fixed search probes, a decrease in recall with increasing search dimensionality d. These trends extend from ImageNet-1K to $4\times$ larger ImageNet-4K, as seen in Figure 16.

G.2. Relative Contrast

We utilize Relative Contrast (He et al., 2012) to capture the difficulty of nearest neighbors search with IVF-MR compared to IVF-RR. For a given database $X = \{x_i \in \mathbb{R}^d, i = 1, ..., N_D\}$, a query $q \in \mathbb{R}^d$, and a distance metric $D(.,.)$ we compute relative contrast C_r as a measure of the difficulty in finding the 1-nearest neighbor (1-NN) for a query q in database X as follows:

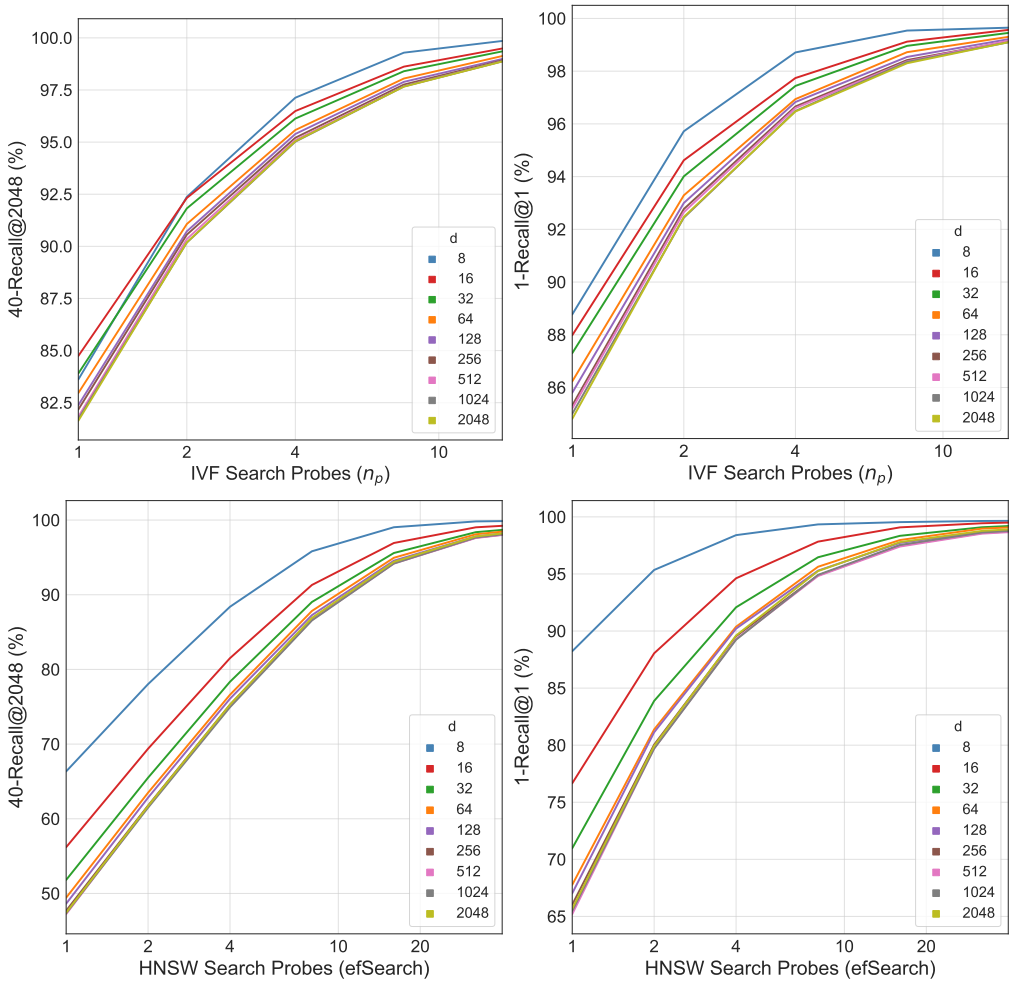


Figure 15. k-Recall@N of d-dimensional MR for IVF and HNSW with increasing search probes n_p on ImageNet-1K.

1. Compute $D^q_{min} = \min_{i=1\ldots n} D(q, x_i)$, i.e. the distance of query q to its nearest neighbor $x^q_{nn} \in X$

2. Compute $D^q_{mean} = E_x[D(q, x)]$ as the average distance of query q from all database points $x \in X$

3. Relative Contrast of a given query $C^q_r = \dfrac{D^q_{mean}}{D^q_{min}}$, which is a measure of how *separable* the query's nearest neighbor x^q_{nn} is from an average point in the database x

4. Compute an expectation over all queries for Relative Contrast over the entire database as

$$C_r = \frac{E_q[D^q_{mean}]}{E_q[D^q_{min}]}$$

It is evident that C_r captures the difficulty of Nearest Neighbor Search in database X, as a $C_r \sim 1$ indicates that for an average query, its nearest neighbor is almost equidistant from a random point in the database. As demonstrated in Figure 7, MRs have higher R_c than RR Embeddings for an Exact Search on ImageNet-1K for all $d \geq 16$. This result implies that a portion of MR's improvement over RR for 1-NN retrieval across all embedding dimensionalities d (Kusupati et al., 2022) is due to a higher average separability of the MR 1-NN from a random database point.

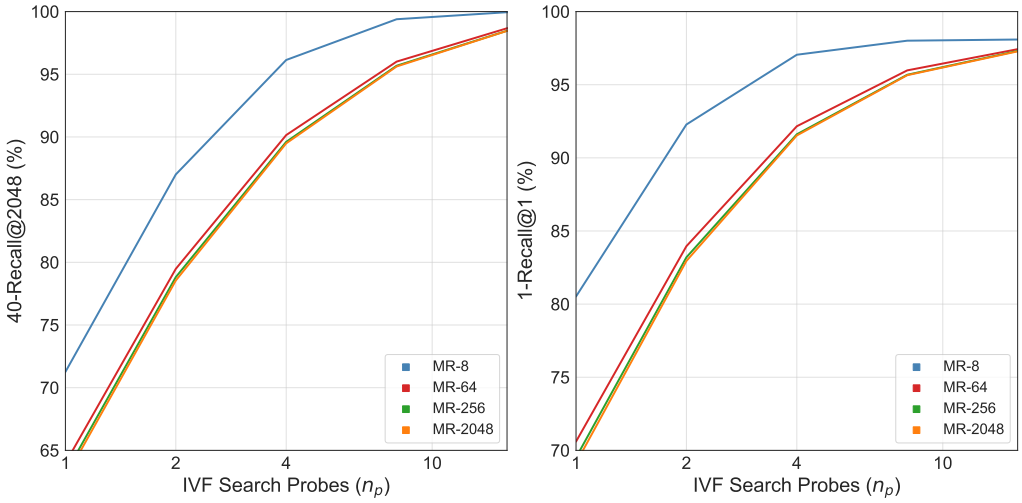


Figure 16. k-Recall@N for IVF-MR-d on ImageNet-4K for $d \in \{8, 64, 256, 2048\}$. Other embedding dimensionalities, HNSW-MR and RR baselines are omitted due to high compute cost. We observe that trends from ImageNet-1K with increasing d and n_p extend to ImageNet-4K, which is $4\times$ larger.

G.3. ResNet Architectures

We perform an ablation over the representation function $\phi : X \to \mathbb{R}^d$ learnt via a backbone neural network (primarily ResNet50 in this work), as detailed in Section 3. We train MRL models (Kusupati et al., 2022) $\phi^{MR(d)}$ on ResNet18/34/101 (He et al., 2016) that are as accurate as their independently trained RR baseline models $\phi^{RR(d)}$, where d is the default max representation size of each architecture. We then compare clustering the MRs via IVF-MR with $k = 2048, n_p = 1$ on ImageNet-1K to Exact-MR, which is shown in Figure 17. IVF-MR shows similar trends across ResNet families compared to Exact-MR, i.e. a maximum top-1 accuracy drop of $\sim 1.6\%$ for a single search probe. This suggests the clustering capabilities of MR extend beyond an inductive bias of $\phi^{MR(d)} \in$ ResNet50, though we leave a detailed exploration for future work.

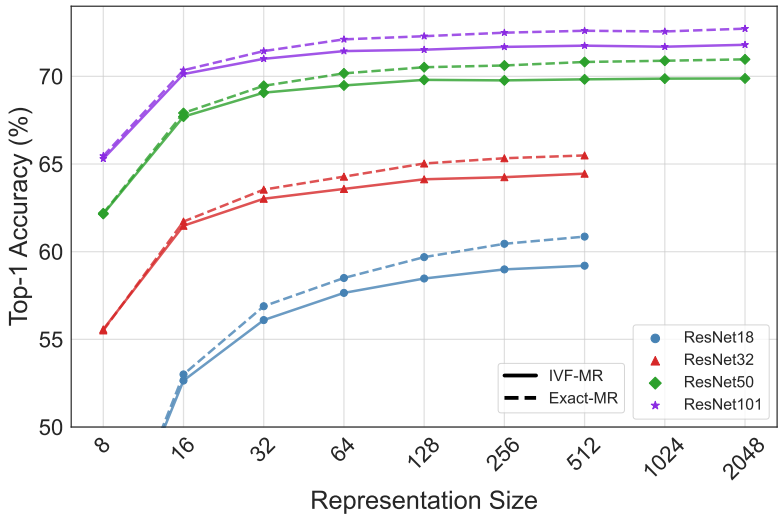


Figure 17. Top-1 Accuracy variation of IVF-MR on ImageNet-1K with different embedding representation function $\phi^{MR(d)}$ (see Section 3), where $\phi \in$ ResNet18/34/101. We observe similar trends between IVF-MR and Exact-MR on ResNet18/34/101 when compared to ResNet50 (Figure 14a) which is the default in all experiments in this work.