Disentangled Representation Learning for Unsupervised Neural Quantization

Haechan Noh[1], Sangeek Hyun[1], Woojin Jeong[1], Hanshin Lim[2], Jae-Pil Heo[1,*]
[1]Sungkyunkwan University, [2]Electronics and Telecommunications Research Institute
{noru0114, hsi1032, us03385, jaepilheo}@skku.edu, hslim@etri.re.kr

Abstract

The inverted index is a widely used data structure to avoid the infeasible exhaustive search. It accelerates retrieval significantly by splitting the database into multiple disjoint sets and restricts distance computation to a small fraction of the database. Moreover, it even improves search quality by allowing quantizers to exploit the compact distribution of residual vector space. However, we firstly point out a problem that an existing deep learning-based quantizer hardly benefits from the residual vector space, unlike conventional shallow quantizers. To cope with this problem, we introduce a novel disentangled representation learning for unsupervised neural quantization. Similar to the concept of residual vector space, the proposed method enables more compact latent space by disentangling information of the inverted index from the vectors. Experimental results on large-scale datasets confirm that our method outperforms the state-of-the-art retrieval systems by a large margin.

1. Introduction

Measuring the distances among feature vectors is a fundamental requirement in various fields of computer vision. One of the tasks most relevant to distance measurement is the nearest neighbor search, which finds the closest data in the database from a query. The task is especially challenging in high-dimensional and large-scale databases due to huge computational costs and memory overhead.

By relaxing the complexity, Approximate Nearest Neighbor (ANN) search is popular in practice. Recent approaches for ANN typically learn the compact representation by exploiting Multi-Codebook Quantization (MCQ) [2, 9, 16]. Compared to hashing-based approaches [1, 10, 12], the MCQ provides a more informative asymmetric distance estimator where the query side is not compressed. Moreover, all possible distances between the query and codewords can be stored in a lookup table for efficiency.

Although the MCQ accelerates the distance computation with the lookup table, exhaustive search on the large-scale dataset is still prohibited. The Inverted File with Asymmetric Distance Computation (IVFADC) [16] is proposed for non-exhaustive ANN search by cooperating with the inverted index [30]. It splits the database into multiple disjoint sets and restricts distance computations to small portions close to the query to accelerate the retrieval speed. Moreover, the compactness of residual vector space between data points and inverted indices substantially enhances the quantization quality.

Thanks to the rapid advances in deep learning, most areas of computer vision benefit from its great learning capacity compared to shallow methods. However, the state-of-the-art methods of unsupervised quantization remain shallow for a long time because selecting the maximum value (i.e. argmax), which is an essential operation of quantization, is not differentiable. Inspired by a recent generative model with discrete hidden variables [35], the Unsupervised Neural Quantization (UNQ) [23] introduced an encoder-decoder-based architecture for ANN search. The large learning capacity of deep neural architecture significantly improves the retrieval quality compared to conventional shallow methods.

Despite the outperforming performance of the UNQ, its superiority is validated only on the exhaustive search. To verify its effectiveness on non-exhaustive search, we conduct an experiment of non-exhaustive UNQ with an inverted index. Interestingly, we observe that this deep architecture does not benefit from the residual vector space and it even harms the search quality as reported in Table 1. We hypothesize the reasons for this performance degradation from two perspectives. First, both the residual vector space and latent space of the neural network transform the data into a quantization-friendly distribution, thus deep quantizer has a scant margin to be improved by the residual space. Second, residual space sacrifices the distributional characteristics of each cluster, since the information of cluster center in the original space is removed. For conventional shallow quantizers, the drawback of residual space is obscured by its huge advantage of making a compact distribution. However, deep quantizer only takes the disadvantages (information loss) from residual space without leveraging the effec-

[*] Corresponding author

tiveness such as compactness of residuals.

In this paper, we focus on extending the application of deep architectures for non-exhaustive search. To this end, we learn a disentangled representation to harmonize a deep architecture with the inverted index, inspired by recent representation learning techniques for generative models [8, 34]. In our disentangled representation learning, both encoder and decoder get information of cluster center as an additional input. Since the information of cluster center is redundant to decoder if latent feature contains information of cluster center, the encoder is trained to remove the information of cluster centers from the latent embedding. The disentangled representation learning is similar to concept of the residual vector space that provides more compact distribution by taking out the information of cluster centers. The experimental results verify that the learning disentangled representation enables the neural quantization to collaborate with inverted index and outperforms the state-of-the-art methods.

The contributions of our paper include:

- We point out that the residual encoding of the inverted index is incompatible with the neural multi-codebook quantization method.
- We propose a novel disentangled representation learning for neural multi-codebook quantization to combine deep quantization and inverted index.
- The experimental results show that the proposed method outperforms the state-of-the-art retrieval systems by a large margin.

2. Related Work

2.1. Multi-codebook Quantization for ANN

Vector quantization [11] maps a vector to its nearest codeword within a learned codebook. Specifically, a high-dimensional real-valued vector can be efficiently represented by an integer index of its corresponding cluster. To ensure a high search accuracy, the vector quantization requires a codebook with an extremely large number of codewords. However, enlarging the number of codewords is far from efficiency because the cost of assigning the nearest codeword is proportionally increased with codebook sizes.

Product Quanztiaton (PQ) [16] decomposes the space into a Cartesian product of lower-dimensional subspaces. By quantizing each subspace separately, multiple codebooks are produced and they enable better search quality with a manageable number of codewords. Optimized Product Quantization (OPQ) [9] and Cartesian K-means [25] introduced to learn transformation to optimize the decomposition step in PQ according to the data distribution.

While the PQ encodes a vector into a concatenation of assigned codewords, a number of methods [2, 6, 20, 21, 26, 36] approximated a vector by a sum of assigned codewords. These addition-based methods have a higher degree of freedom than the concatenation-based methods, since the concatenation is a special case of addition where components of addition are mutually orthogonal.

The state-of-the-art methods of such multi-codebook quantization remain shallow for a long time, while most areas of computer vision benefit from rapid advances in deep learning. The non-differentiability of codeword assignment operation makes the multi-codebook quantization hard to collaborate with deep learning. Vector-Quantized Variational Autoencoder (VQ-VAE) [35] in the area of the generative model proposed a gradient estimation to propagate gradient through discrete variables. Inspired by the VQ-VAE, Unsupervised Neural Quantization (UNQ) [23] introduces a deep neural network for multi-codebook quantization. The much larger learning capacity of UNQ significantly improves the conventional shallow methods.

2.2. Non-exhaustive ANN Search

While the quantization techniques accelerate the distance computation, the search is still exhaustive. IVFADC [16] proposed a non-exhaustive approximate nearest neighbor search by cooperating with the inverted index [30]. In addition to the significant acceleration of retrieval speed, the non-exhaustive manner even improves the search quality by allowing more compact residual vector space. The Inverted Multi-Index (IMI) [3] extends the IVFADC by decomposing data space into two sub-spaces and applying the inverted index independently. Similar to the idea of PQ, the decomposed data space of IMI enables much finer partition with a manageable number of indices. Several subsequent non-exhaustive ANN techniques are proposed to improve the data partitioning rather than the quantization [4, 5, 7].

2.3. Disentangled Representation Learning

In the context of the generative model, disentangled representation is introduced to analyze and utilize the independent effects of latent factors. One straightforward example can be found in conditional generative models [22, 32]. They decompose the conditional information from the others so that they can control the specific conditions of generated images such as spatial layout [8] or class information [22]. For instance, variational U-Net [8] disentangles the shape and appearance representations to synthesize human images of diverse poses with a shared appearance. Similarly, VarSR [13] builds the latent space of high-resolution information disentangled from the low-resolution images. Hence, they can synthesize diverse high-resolution images corresponding to the given low-resolution image.

The aforementioned examples mainly focus on the controllable generation by disentangling conditional information. However, in this paper, we concentrate on the maximization of the representation capability of the latent space

Method	R@1	R@10	R@100
PQ	0.228	0.653	0.953
+IVF (w/o residual)	0.228	0.653	0.950
+IVF (with residual)	**0.272**	**0.735**	**0.969**
UNQ	**0.346**	**0.828**	**0.990**
+IVF (w/o residual)	0.331	0.798	0.939
+IVF (with residual)	0.272	0.718	0.867

Table 1. Comparison of retrieval performances between the PQ and the UNQ on SIFT-1M. The residual vector space improves the performance with the PQ, but does not work with the UNQ.

by disentanglement. Specifically, we build a compact latent space by disentangling the cluster center similar to residual encoding but compatible with the neural multi-codebook quantization method.

3. Background

Let us briefly introduce the ANN task, the MCQ approach, and the residual vector space. With D-dimensional database $X = \{x_n\}_{n=1}^N, x_n \in \mathbb{R}^D$ and query $y \in \mathbb{R}^D$, the nearest neighbor of query y is defined as follows:

$$\mathrm{NN}(y) = \underset{n \in \{1,\dots,N\}}{\arg\min} \; ||y - x_n||^2. \tag{1}$$

To alleviate time and memory costs of Eq. 1, the MCQ quantizer q_{MCQ} encodes the database $X = \{x_n\}_{n=1}^N$ utilizing M number of multi-codebooks $C^m = \{c_k^m\}_{k=1}^K, c_k^m \in \mathbb{R}^D$ where each codebook has K number of codewords. Then, a vector x_n is represented as M number of indices $[i_n^1, ..., i_n^M]$ and reconstructed by referring the codebooks:

$$x_n \approx q_{\mathrm{MCQ}}(x_n) = \sum_{m=1}^{M} c_{i_n^m}^m. \tag{2}$$

The goal of this lossy compression process is to learn codebooks that minimize quantization distortion E defined as:

$$E(x_n, C) = \sum_{m=1}^{M} ||x_n^m - c_{i_n^m}^m||^2, \tag{3}$$

where $x_n = \sum x_n^m$, and x_n^m is a subvector of x_n corresponding to index i_n^m. The PQ belongs to the MCQ approach where x_n^m are mutually orthogonal.

The IVFADC [16] further accelerates ANN search by collaborating with an inverted index. The inverted index q_{IVF} splits database X into K' number of disjoint subsets, and the ANN search is then restricted to w number of nearest subsets from query y, where w is much smaller than K'. For a given data point x_n, the inverted index $q_{\mathrm{IVF}}(\cdot)$ quantize it to its nearest cluster center $q_{\mathrm{IVF}}(x_n)$, and the data points belong to the same cluster are stored adjacently. Moreover, the IVFADC suggested learning codebooks that encode residual vector $r(x_n) = x_n - q_{\mathrm{IVF}}(x_n)$ rather than the original vector as follows:

$$x_n \approx \tilde{x}_n = q_{\mathrm{MCQ}}(x_n - q_{\mathrm{IVF}}(x_n)) + q_{\mathrm{IVF}}(x_n). \tag{4}$$

Then the distance between query y and target data x_n is estimated as follows:

$$d(y, x_n) \approx d(y - q_{\mathrm{IVF}}(x_n), q_{\mathrm{MCQ}}(x_n - q_{\mathrm{IVF}}(x_n))). \tag{5}$$

where the $d(\cdot,\cdot)$ is the Euclidean distance between two vectors. The IVFADC accelerates the search speed by reducing the number of distance estimations and improves the quantization distortion (Eq. 3) thanks to the compact distribution of the residual vector space.

Method	Compression ratio	Distortion
PQ	128×	0.8368
+IVF (with residual)		**0.7208**
PQ	64×	0.6252
+IVF (with residual)		**0.5846**
PQ	32×	**0.4127**
+IVF (with residual)		0.4220
PQ	16×	**0.2438**
+IVF (with residual)		0.2604
PQ	4×	**0.0212**
+IVF (with residual)		0.0257

Table 2. Quantization distortions of the database with varying compression ratios. The residual vector space is profitable with a large compression ratio, but it leads performance drop with a small compression ratio.

4. Proposed Method

4.1. Motivation

Recently, the UNQ [23] introduced a deep architecture for MCQ learning and improves conventional shallow MCQ methods by a large margin. However, its effectiveness is only validated by an exhaustive ANN search. To examine the performance of UNQ on non-exhaustive search, we test the retrieval performances of PQ and UNQ with an inverted index on exhaustive and non-exhaustive settings. As reported in Table. 1, the residual encoding (Eq. 4) provides better search quality with the PQ, while it makes the search quality of UNQ even worse.

To investigate the reasons why residual encoding does not help the UNQ, we need to analyze the mechanism of residual encoding in PQ. To this end, we hypothesize the effect of residual representation in two aspects. First, the advantage of residuals in PQ comes from the transformation of data space into compact and quantization-friendly distribution. Second, the residual encoding is not always beneficial, since they sacrifice the information of the cluster center so that the distributional characteristics of each cluster are suppressed.

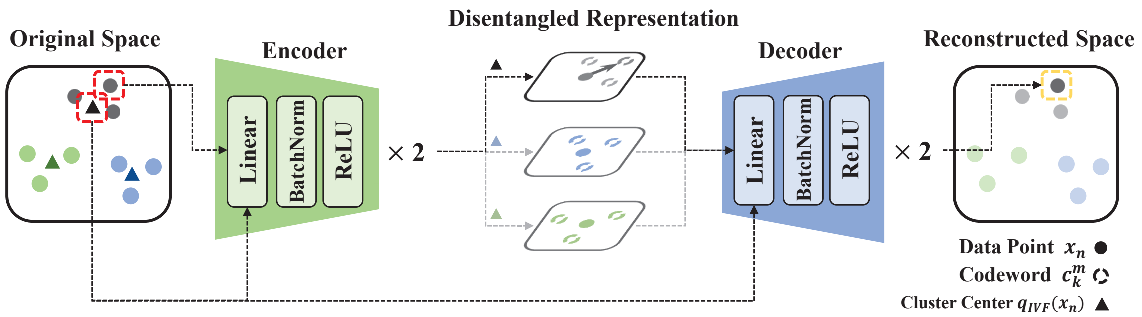

Figure 1. By providing additional input of $q_{\mathrm{IVF}}(x_n)$ to both encoder and decoder, the encoder is not necessary to learn features to reconstruct $q_{\mathrm{IVF}}(x_n)$ since it is redundant information for the decoder. Thus the information of the cluster center is disentangled from the latent vectors and codebooks. The disentangling enables the quantizer to learn compact representation while preserving the original distribution.

To validate these hypotheses, we observe the performance variation of PQ with residual encoding by varying compression ratios. The compression ratios are controlled by adjusting the number of codebooks M. As reported in Table. 2, the residual encoding is profitable with large compression ratios, since the sparse original distribution is hard to quantize with the deficient bit-budget. On the other hand, the residual encoding leads to performance drops with small compression ratios, because it is not necessary to represent the data compactly thanks to sufficient numbers of codewords, but the information loss of cluster-wise distributional characteristics still exists.

Based on these observations, we explain the reasons for UNQ's performance drop with residual encoding. Firstly, UNQ does not benefit from the compact representation of residual encoding. The deep encoder transforms the data into latent space with a reconstruction objective function. That is, the objective of UNQ also encourages to transform the data into compact latent space, so the residual encoding and UNQ affect the model in a similar way. Hence, the residual representation is hardly beneficial to the neural quantizer. Secondly, UNQ takes the disadvantage of residual encoding, the information loss of distributional characteristics of each cluster.

These interpretations motivate us to come up with a method that only takes the advantage of residual representation. That is, we aim to make the latent space more compact without the information loss of distributional characteristics of each cluster. To this end, we introduce a disentangled representation learning for unsupervised neural quantization in the following sections.

4.2. Disentangled Neural Quantization

Inspired by the disentangled representation of conditional generative models [22, 32], we propose disentangled neural quantization. The previous methods exploit the disentanglement for a controllable generation. However, we propose to learn a disentangled representation to reduce the quantization distortion. Specifically, a latent feature (l_n) of UNQ contains information of its input vector x_n where x_n consists of $q_{\mathrm{IVF}}(x_n)$ and $x_n - q_{\mathrm{IVF}}(x_n)$. However, the $q_{\mathrm{IVF}}(x_n)$ can be restored without any information loss thanks to the data structure of the inverted index. Thus, encoding the $q_{\mathrm{IVF}}(x_n)$ is quite wasteful because the MCQ approach has a scarce bit budget in practice. In this paper, we suggest disentangling the $q_{\mathrm{IVF}}(x_n)$ from l_n to provide more compact information.

To disentangle the $q_{\mathrm{IVF}}(x_n)$ from l_n, we feed additional input of $q_{\mathrm{IVF}}(x_n)$ to encoder $E(\cdot)$ and decoder $D(\cdot)$. Then the $q_{\mathrm{IVF}}(x_n)$ is redundant information for $D(\cdot)$ if l_n contains information of $q_{\mathrm{IVF}}(x_n)$. Thus, the $E(\cdot)$ is trained to embed as little information of $q_{\mathrm{IVF}}(x_n)$ as possible in l_n to fully exploit the network capacity. The disentangled representation learning is similar to the concept of the residual encoding of PQ (Eq. 4) in terms of taking out the information of the cluster center. However, there is no information loss because the cluster center $q_{\mathrm{IVF}}(x_n)$ is provided to $D(\cdot)$.

Figure. 1 describes the proposed disentangled representation learning. The cluster center of input data is provided to the encoder and decoder, and the scarce resource of finite codewords can be fully utilized to quantize the compact latent feature where the information of the cluster center is disentangled. Not only the compact information, but the decoder also preserves the original distribution by reconstructing the original space in an end-to-end manner.

4.3. Network Structure

We adopt the network structure of [23] except our disentangling module. The network consists of encoder $E(\cdot)$, decoder $D(\cdot)$, and learnable codebook parameters $[C^1, ..., C^M], C^m = \{c_k^m\}_{k=1}^K$. Both encoder $E(\cdot)$ and decoder $D(\cdot)$ are a concatenation of two fully-connected layers with Batch Normalization [14] and ReLU activation function. We feed separate information of cluster center $q_{\mathrm{IVF}}(x_n)$ to the encoder for the disentanglement. Then the encoder $E(\cdot)$ embeds the given concatenated inputs

Search manner	Method	64bits encoding			128bits encoding		
		R@1	R@10	R@100	R@1	R@10	R@100
Exhaustive	OPQ	0.208	0.643	0.953	0.409	0.898	0.999
	Catalyst + Lattice	0.289	0.758	0.979	0.491	0.941	**1.**
	LSQ	0.292	0.777	0.987	0.571	0.975	**1.**
	UNQ	0.346	0.828	**0.990**	0.593	**0.980**	**1.**
Non-Exhaustive	IVFADC (2^{10})	0.296	0.704	0.957	0.471	0.903	0.993
	IVFADC + QAI (2^{10})	0.311	0.730	0.965	0.496	0.928	0.992
	IVFOADC (2^{10})	0.295	0.719	0.963	0.475	0.913	0.995
	IVFOADC + QAI (2^{10})	0.321	0.756	0.972	0.500	0.931	0.993
	Multi-D-ADC ($2^8 \times 2^8$)	0.303	0.729	0.967	0.484	0.924	0.998
	Multi-D-ADC + QAI ($2^8 \times 2^8$)	0.318	0.748	0.977	0.513	0.939	0.997
	UNQ+IVF (2^{10})	0.331	0.798	0.939	0.548	0.963	0.994
	Ours (2^{10})	**0.398**	**0.877**	0.971	**0.600**	0.974	0.990

Table 3. Experimental result on the SIFT-1M dataset. All the quantities except Ours and UNQ+IVF are taken from [23, 24]

$[x_n, q_{\text{IVF}}(x_n)]$ to M number of latent features as follows:

$$E([x_n, q_{\text{IVF}}(x_n)]) = [l^1_n, ..., l^M_n]. \tag{6}$$

The embedded feature l^m_n is quantized to index i^m_n of the codeword that has the biggest dot-product with l^m_n as follows:

$$i^m_n = \underset{k\in\{1,...,K\}}{\arg\max}\ l^m_n \cdot c^m_k. \tag{7}$$

Because the argmax function is not differentiable, we follow [23] that exploits the Gumbel trick [15]. The Gumbel trick provides a differentiable approximation of Eq. 7 by substituting the argmax with softmax during backpropagation. Then, the sum of assigned codewords $c^m_{i^m_n}$ becomes the input for the decoder $D(\cdot)$. Likewise, we concatenate it with the cluster center $q_{\text{IVF}}(x_n)$ for the disentanglement. The approximated vector $\tilde{x_n}$ of original vector x_n is obtained as follows:

$$\tilde{x}_n = D\left(\left[\ \sum_{m=1}^{M} c^m_{i^m_n}\ ,\ q_{\text{IVF}}(x_n)\right]\right). \tag{8}$$

The model is trained with two losses. The first one is a reconstruction loss which directly minimizes the quantization distortion as follows:

$$L_1 = \sum_{n=1}^{N} ||x_n - \tilde{x}_n||^2. \tag{9}$$

Similar to [23], we adopt the square Coefficient Variation regularizer [29] to encourage the codewords to be evenly selected:

$$L_2 = \sum_{m=1}^{M} \frac{\sum_{n=1}^{N} (p^m_n - p^m_{\text{avg}})^2}{{p^m_{\text{avg}}}^2}, \tag{10}$$

where p^m_n is the softmax probability of i^m_n by the Gumbel approximation in Eq. 7, and p^m_{avg} is the average of p^m_n along N. The final loss of the model is described as follows:

$$L = L_1 + \lambda * L_2 \tag{11}$$

The loss ratio λ starts from 1.0 and decreases to 0.05 during the training. For a fair comparison, we adopt other details of [23] including Quasi-Hyperbolic Adam algorithm [19], and One Cycle learning rate schedule [31].

4.4. Retrieval Procedure

Firstly, each database vector x_n is encoded into M number of indices $l_n = [i^1_n, ..., i^M_n]$ by Eq. 7. Because the codebooks have K number of codewords, the required bit-budget for the indices l_n is $M * \lceil log_2 K \rceil$. Moreover, the data points that belong to the same inverted index are stored adjacently, and the size of each cluster is stored. Thanks to this data structure of the inverted index, we can identify the $q_{\text{IVF}(x_n)}$ without any additional costs in the retrieval stage.

Secondly, a given query y is embedded into latent features l_y, and the cluster center of the target is concatenated.

$$[l^1_y, ..., l^M_y] = E(\ [\ y\ ,\ q_{\text{IVF}}(x_n)\ ]\ ). \tag{12}$$

Then the symmetric distance between encoded query l_y and encoded target $[i^1_n, ..., i^M_n]$ is computed by the cosine similarity as follows:

$$D_{\text{sym}}(y, x_n) = \sum_{m=1}^{M} \frac{l^m_y \cdot c^m_{i^m_n}}{||l^m_y||\ ||c^m_{i^m_n}||}. \tag{13}$$

Note that, Eq. 13 can be efficiently computed by a lookup table storing precomputed all possible symmetric distances.

Finally, the R top-ranked retrieved data according to the symmetric distance are then re-ranked by the more accurate asymmetric distances between uncompressed query y and $\tilde{x}_n$, as follows:

$$D_{\text{asym}}(y, x_n) = ||y - \tilde{x}_n||^2. \tag{14}$$

We set the hyper-parameter $R = 200$, and we conduct an ablation study of R in Table. 7.

Search type	Method	64bits encoding			128bits encoding		
		R@1	R@10	R@100	R@1	R@10	R@100
Exhaustive	OPQ	0.159	0.513	0.886	0.350	0.825	0.991
	Catalyst + Lattice	0.246	0.683	0.961	0.448	0.908	**0.998**
	LSQ	0.217	0.640	0.945	0.411	0.886	0.995
	UNQ	0.267	0.726	**0.973**	0.479	**0.930**	**0.998**
Non-exhaustive	IVFADC (2^{10})	0.210	0.616	0.925	0.418	0.890	0.995
	IVFOADC (2^{10})	0.227	0.659	0.950	0.416	0.897	0.996
	Multi-D-ADC ($2^8 \times 2^8$)	0.200	0.613	0.929	0.398	0.883	0.995
	UNQ+IVF (2^{10})	0.250	0.688	0.910	0.440	0.907	0.996
	Ours (2^{10})	**0.329**	**0.799**	0.932	**0.491**	0.929	0.970

Table 4. Experimental results on DEEP-1M dataset. The quantities of Exhaustive methods are taken from [23]

5. Evaluation

5.1. Protocol

We evaluate our method on three datasets:

- **SIFT-1M, SIFT-1B** [16]: 128-dimensional SIFT descriptor vectors [18]. The training set includes $\{500k, 10^8\}$ vectors for 1M and 1B, respectively. Sets for evaluation have 10^4 queries and $\{10^6, 10^9\}$ number of database for 1M and 1B, respectively.
- **DEEP-1M** [4]: 96-dimensional DNN features extracted from the last fully-connected layer of GoogLeNet [33]. It also includes $500k$ training data, 10^4 queries, and 10^6 database.

The ground truth of a given query is defined as the top-1 nearest neighbor from an uncompressed database. Then, the retrieval performance in compressed space is measured by average *recall@R* for $R = \{1, 10, 100\}$. The *recall@R* represents the ratio of queries that contains the ground truth in top-R retrieved vectors.

Unless specified, we consistently use the following hyper-parameters. We set $K' = 2^{10}$ for inverted index on million-scale datasets, and $K' = 2^{12} \times 2^{12}$ for inverted multi-index on billion-scale dataset. The bit budget is controlled by the number of codebooks M and the number of codewords K. We fix the K as 256, and control the bit-budgets by changing $M = \{8, 16\}$ for 64bits and 128bits, respectively. We follow the setting of w from [7, 24] for a fair comparison. For each query, it visits w inverted indices that guarantee at least $\{50k, 10^5\}$ candidates for million-scale datasets and billion-scale datasets, respectively.

We compare the proposed method with two sets of state-of-the-arts that are exhaustive and non-exhaustive methods. In particular, exhaustive methods include the followings:

- **OPQ** [9, 25]: Product Quantization [16] with optimized space decomposition.
- **Catalyst + Lattice** [28]: Learning spreaded space that fits with pre-defined lattice.
- **LSQ** [20]: The state-of-the-art of shallow MCQ.
- **UNQ** [23]: A deep generalization of MCQ.

Plus, the non-exhaustive methods include the followings:

- **IVFADC** [16]: Product Quantization [16] with the Inverted index [30] and the residual encoding.
- **IVFOADC** [9, 25]: OPQ with the inverted index [30] and the residual encoding.
- **Multi-D-ADC** [3]: The Inverted Multi-Index (IMI).
- **OMulti-D-OADC** [9, 25]: The IMI with the OPQ.
- **QAI** [24]: Jointly optimized inverted index and quantizer.
- **IVFOADC + GP** [5]: IVFOADC with grouping and pruning algorithm of [5].
- **UNQ + IVF**: Inverted file system with the UNQ. We omit the residual encoding because it is incompatible with the deep MCQ algorithm (Table. 1).
- **Ours**: Proposed disentangled representation learning. We select a single index for the million-scale datasets and a multi-index for the billion-scale dataset.

5.2. Result on Million-Scale Dataset

Table. 3 and Table. 4 report the *recall@R* scores of exhaustive and non-exhaustive ANN search methods with $\{64, 128\}$bits encoding. For all combinations of the dataset and bit budget, the proposed method outperforms the non-exhaustive baselines by a large margin. For example, **Ours** improves the R@1 score of **Multi-D-ADC+QAI** about 25% and 17% for 64 and 128bits encoding respectively on the SIFT-1M dataset. On the DEEP-1M dataset, the performance gains between **Ours** and **Multi-D-ADC** are much larger than the gains on the SIFT-1M, where the gaps are about 65% and 23% for 64 and 128bits encoding, respectively.

Notably, the proposed method even outperforms the exhaustive ANN methods while the proposed method is a non-exhaustive search where only 5% of the database is considered for the distance computation. For instance, **Ours** improves the R@1 score of the exhaustive **UNQ** by about

Method	R@1	R@10	R@100
Exhaustive Search			
Catalyst + Lattice	0.311	0.778	0.983
LSQ	0.380	0.856	0.993
UNQ	0.383	0.868	**0.994**
Non-Exhaustive Search			
IVFADC (2^{20})	0.351	0.786	0.918
Multi-D-ADC ($2^{14} \times 2^{14}$)	0.344	0.809	0.960
OMulti-D-OADC ($2^{14} \times 2^{14}$)	0.373	0.841	0.973
IVFOADC+GP (2^{20})	0.405	0.851	0.957
Ours ($2^{12} \times 2^{12}$)	**0.458**	**0.859**	0.903

Table 5. Experimental results on the SIFT-1B dataset and 128bits bit-budget. All the quantities except Ours are taken from [5,9,23].

Method	w	R@1	R@10	R@100
UNQ+IVF (w/o residual)	1	0.216	0.412	**0.442**
Ours		**0.253**	**0.430**	**0.442**
UNQ+IVF (w/o residual)	2	0.272	0.556	0.612
Ours		**0.317**	**0.588**	**0.613**
UNQ+IVF (w/o residual)	4	0.305	0.674	0.763
Ours		**0.361**	**0.720**	**0.765**
UNQ+IVF (w/o residual)	8	0.322	0.753	0.877
Ours		**0.387**	**0.809**	**0.880**
UNQ+IVF (w/o residual)	16	0.331	0.791	0.945
Ours		**0.397**	**0.857**	**0.947**
UNQ+IVF (w/o residual)	32	0.332	0.804	0.977
Ours		**0.399**	**0.876**	**0.979**
UNQ+IVF (w/o residual)	64	0.332	0.806	0.985
Ours		**0.399**	**0.880**	**0.987**

Table 6. Ablation study about the number of visited inverted indices on the SIFT-1M dataset with 64bits bit budget.

15% on the SIFT-1M and 23% on the DEEP-1M dataset with 64bits bit budget.

As our motivation, a naive combination of the UNQ and the inverted index does not improve the performance of the exhaustive UNQ. For example, the **UNQ+IVF** drops the R@1 score of the exhaustive **UNQ** about 5% and 7% for SIFT-1M and DEEP-1M with 64bits bit budget, respectively. The disentangled representation of **Ours** improves the R@1 scores of **UNQ+IVF** about 20% and 32% on SIFT-1M and DEEP-1M with 64bits bit budget, respectively.

5.3. Result on Billion-Scale Dataset

We conduct an experiment on a billion-scale dataset to verify the scalability of the proposed method. Table. 5 reports the experimental results on the SIFT-1B dataset with 128bit budget. The improvement in the R@1 score of **Ours** over **IVFOADC+GP** is about 13%. Moreover, **Ours** shows 20% of improvement compared to the **UNQ** while **Ours** is a non-exhaustive search and the **UNQ** is an exhaustive search.

Notably, the margin of improvement is much larger with the SIFT-1B dataset than the SIFT-1M dataset with 128bits encoding. Because the billion-scale task is a much more challenging task than the million-scale task with an identical bit budget, the billion-scale task has much more room for performance gain.

Furthermore, we choose a conventional Multi-D-ADC algorithm with $2^{12} \times 2^{12}$ number of indices for learning the inverted index because our contribution is not about the inverted index. However, the experimental results demonstrate the superiority of **Ours** compared to methods with much finer inverted index ($2^{14} \times 2^{14}$) and a sophisticated inverted indexing algorithm (**IVFOADC+GP**).

5.4. Ablation Study

We validate our method with ablation study on two hyper-parameters. First, Table. 6 shows ablation study about the number of visited inverted indices w on the SIFT-1M with 64bits encoding. **Ours** consistently improves **UNQ+IVF** regardless of the number of w. For example, the performance gain of the R@1 score is about 17% with $w = 1$ and 20% with $w = 64$. Moreover, these experimental results also highlight that the R@1 score of **Ours** with only $w = 4$ (out of 1024) is even superior to the exhaustive **UNQ** in Table. 3.

Second, we verify with various numbers of re-reranking candidates $R = \{1, 10, 100, 1000\}$. It is identical to exclude the re-ranking if $R = 1$. Table. 7 shows the experimental results on the SIFT-1M and the DEEP-1M datasets with 64bits encoding. For the SIFT-1M dataset, **Ours** improves the more performance with the fewer number of re-ranking. For example, the performance gains of the R@1 score are about {62%, 28%, 21%, 20%} with $R = \{1, 10, 100, 1000\}$, respectively. On the other hand, **Ours** improves the more performance with the larger number of re-ranking for the DEEP-1M dataset where the performance gains of the R@1 score are about {17%, 29%, 32%, 32%}.

5.5. Validation of Disentanglement

We validate disentangled representation of the proposed method by generating new data points. Likewise disentanglement methods for generative models, we generate data points that have desired characteristics by controlling the input of the decoder. In detail, a reconstructed data point $\tilde{x}_n$ should belong to a specific cluster by substituting the $q_{\text{IVF}}(x_n)$ by a desired cluster center T. We randomly sample 10^4 data points from the database and generate about 10^7 new data points by concatenating $K' = 1024$ number of cluster centers to each l_n. Table. 8 reports the ratio of generated data points that belong to the desired cluster within top-R nearest clusters. The R@1 score of SIFT-1M and DEEP-1M datasets are 0.709 and 0.827, respectively

Method	# of re-rank	R@1
SIFT-1M 64bits		
UNQ+IVF (w/o residual)	w/o re-rank	0.146
Ours		**0.236**
UNQ+IVF (w/o residual)	10	0.292
Ours		**0.375**
UNQ+IVF (w/o residual)	100	0.329
Ours		**0.397**
UNQ+IVF (w/o residual)	1000	0.332
Ours		**0.399**
DEEP-1M 64bits		
UNQ+IVF (w/o residual)	w/o re-rank	0.134
Ours		**0.157**
UNQ+IVF (w/o residual)	10	0.228
Ours		**0.295**
UNQ+IVF (w/o residual)	100	0.248
Ours		**0.328**
UNQ+IVF (w/o residual)	1000	0.249
Ours		**0.329**

Table 7. Ablation study about the number of re-ranked candidates on million-scale datasets with 64bits bit budget.

while the probability of belonging to the desired cluster of a randomly generated vector is 0.001. The experimental results confirm that the proposed representation is well disentangled as we intended.

5.6. Overhead Analysis

Finally, we analyze the search time overhead of the proposed method. Table. 9 demonstrates the average search time of **Multi-D-ADC** and **Ours** with identical cluster centers of the inverted index. The retrieval procedure of **Ours** is similar to the **Multi-D-ADC** except the additional encoding (Eq. 12) and decoding (Eq. 8) steps. In encoding query (Eq. 12), a query y should be concatenated with w number of cluster centers and embedded w times to builds the lookup table. To accelerate the encoding query, we propose batch-forwarding w number of lookup tables at once while the **Multi-D-ADC** build the lookup table in sequence. Similarly, the decoding top-R (Eq. 8) also can be simply accelerated by batch-forwarding. The re-ranking time (Eq. 14) is neglectable because the number of re-ranked data ($R = 200$) is small. The overall search time of **Ours** is about 12% slower than the **Multi-D-ADC**, but we believe it is permissible compared to the performance gain of **Ours**. Nevertheless, **Ours** speeds up the search procedure of the UNQ, which requires more than a minute in an exhaustive search. The experimental results are obtained in a single-CPU mode of a 2.2 GHz Intel Xeon processor with Faiss [17], and the encoding and decoding times are measured in a single Nvidia Tesla V100 GPU with Pytorch [27].

Dataset	Method	R@1	R@3	R@5
SIFT-1M	$D(\,[\,l_n,\ T\,]\,)$	0.709	0.923	0.968
DEEP-1M	$D(\,[\,l_n,\ T\,]\,)$	0.827	0.965	0.987
-	Random	0.001	0.003	0.005

Table 8. Experiment to validate disentanglement of proposed method with 64bits bit-budget. The score represents the ratio of generated data points that belong to desired cluster within top-R nearest clusters.

Method	Time (ms)
Multi-D-ADC ($2^{12} \times 2^{12}$)	**16.89**
Ours ($2^{12} \times 2^{12}$)	18.88
1. encoding query (Eq. 12)	1.27
2. symmetric search (Eq. 13)	16.86
3. decoding top-R (Eq. 8)	0.70
4. asymmetric re-ranking (Eq. 14)	0.04

Table 9. Search time analysis on the SIFT-1B dataset and 128bits bit-budget.

The proposed method requires {21, 32}MB for {64, 128} bits encoding where the UNQ requires {20, 30}MB for the same budget. Note that those additional memory usage is not proportional to the database size, thus it is same for 1M and 1B database.

6. Conclusion

In this paper, we propose a novel disentangled representation learning for unsupervised neural quantization. We firstly discover the cause of limited application of the deep quantization (UNQ [23]) on non-exhaustive search with the inverted index. Specifically, the deep quantizer is hard to exploit the compact representation of residual vector space and even takes disadvantages such as loss of cluster center information. To address the problem, we disentangle the information of the cluster center and the latent space by feeding the center to both the encoder and decoder to make center information redundant for the decoder. Similar to residual encoding, this disentanglement encourages the latent space compact and quantization-friendly by taking out the center information from latent space, so the quantization distortion is significantly reduced with disentangled representation. Extensive experiments on various datasets from million to billion scales validate the superiority of the proposed method over state-of-the-art retrieval systems.

Acknowledgements. This work was supported in part by ETRI grant funded by the Korean government [23ZH1200, The research of the fundamental media·contents technologies for hyper-realistic media space], and MSIT/IITP [2019-0-00421], MSIT/NRF [NRF-2022R1F1A1076132], and MSIT&KNPA/KIPoT [Police Lab 2.0, 210121M06].

References

[1] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008. 1

[2] Artem Babenko and Victor Lempitsky. Additive quantization for extreme vector compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 931–938, 2014. 1, 2

[3] Artem Babenko and Victor Lempitsky. The inverted multi-index. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1247–1260, 2014. 2, 6

[4] Artem Babenko and Victor Lempitsky. Efficient indexing of billion-scale datasets of deep descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2055–2063, 2016. 2, 6

[5] Dmitry Baranchuk, Artem Babenko, and Yury Malkov. Revisiting the inverted indices for billion-scale approximate nearest neighbors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 202–216, 2018. 2, 6, 7

[6] Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010. 2

[7] Chih-Yi Chiu, Amorntip Prayoonwong, and Yin-Chih Liao. Learning to index for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):1942–1956, 2019. 2, 6

[8] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8857–8866, 2018. 2

[9] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):744–755, 2013. 1, 2, 6, 7

[10] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2916–2929, 2012. 1

[11] Robert M. Gray and David L. Neuhoff. Quantization. *IEEE transactions on information theory*, 44(6):2325–2383, 1998. 2

[12] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon. Spherical hashing. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2957–2964. IEEE, 2012. 1

[13] Sangeek Hyun and Jae-Pil Heo. Varsr: Variational super-resolution network for very low resolution images. In *European Conference on Computer Vision*, pages 431–447. Springer, 2020. 2

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 4

[15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017. 5

[16] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010. 1, 2, 3, 6

[17] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 8

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 6

[19] Jerry Ma and Denis Yarats. Quasi-hyperbolic momentum and adam for deep learning. In *International Conference on Learning Representations*, 2018. 5

[20] Julieta Martinez, Joris Clement, Holger H Hoos, and James J Little. Revisiting additive quantization. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016. 2, 6

[21] Julieta Martinez, Shobhit Zakhmi, Holger H Hoos, and James J Little. Lsq++: Lower running time and higher recall in multi-codebook quantization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 491–506, 2018. 2

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2, 4

[23] Stanislav Morozov and Artem Babenko. Unsupervised neural quantization for compressed-domain similarity search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3036–3045, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[24] Haechan Noh, Taeho Kim, and Jae-Pil Heo. Product quantizer aware inverted index for scalable nearest neighbor search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12210–12218, 2021. 5, 6

[25] Mohammad Norouzi and David J Fleet. Cartesian k-means. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3017–3024, 2013. 2, 6

[26] Ezgi Can Ozan, Serkan Kiranyaz, and Moncef Gabbouj. Competitive quantization for approximate nearest neighbor search. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):2884–2894, 2016. 2

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 8

[28] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *ICLR 2019-7th International Conference on Learning Representations*, pages 1–13, 2019. 6

[29] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean.

Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview.net, 2017. 5

[30] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society, 2003. 1, 2, 6

[31] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 5

[32] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015. 2, 4

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[34] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 2

[35] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1, 2

[36] Ting Zhang, Chao Du, and Jingdong Wang. Composite quantization for approximate nearest neighbor search. In *International Conference on Machine Learning*, pages 838–846. PMLR, 2014. 2