# Supplementary Material: Hierarchical Quantization Consistency for Fully Unsupervised Image Retrieval

Chao Zhang[1]
chao.zhang@crl.toshiba.co.uk
Stephan Liwicki[1]
stephan.liwicki@crl.toshiba.co.uk
Roberto Cipolla[1,2]
rc10001@cam.ac.uk

[1] Cambridge Research Lab
Toshiba Europe Ltd
Cambridge, UK

[2] Department of Engineering
University of Cambridge
Cambridge, UK

**Abstract**

We provide details of the datasets and evaluation protocol used in this work in Sec.1. Implementations details such as hyperparameters (e.g. learning rate and temperature parameters) and weighting parameters are given in Sec.2. Given the trained model, the details of inference step is described in Sec.2.2. In Sec.2.3, the pseudo code of the propose method is given. Additionally, we show the PR curves and P@1000 curves on NUS-WIDE and FLICKR25K for the completeness of evaluations. Lastly, we report the full results and ablations.

## 1 Dataset and Evaluation Protocol

**CIFAR-10** consists of 60,000 images of 10 class, where each class has 5,000 images for training and 1,000 images for testing. We use 1,000 images per class as the query set, while the remaining images are used as the training set and the retrieval database.
**NUS-WIDE** is a multi-label large-scale dataset with around 270,000 images of 81 categories. We select images of the 21 most frequent categories for evaluation, where 100 images per categories are selected to form 21,000 images as the query set while the remaining images form the training set and the retrieval database.
**FLICKR25K** is a relatively small dataset with 25,000 images of 24 categories. We randomly select 2,000 images as the query set while the remaining images are used as the training set and the retrieval database. On the multi-label NUS-WIDE and FLICKR25K, if a query image and a database image share at least one label, then they are defined as the true match [9, 14].

## 2 Implementation Details

We implement our approach with Python and PyTorch. Following [9], we use the modified ResNet-18 [8, 9] as the backbone (feature extractor) for CIFAR-10 where the first convolutional layer is modified with small kernel size and stride to adapt to the small 32×32 input image size, and use standard ResNet-50 [8] as the backbone for NUS-WIDE and FLICKR25K. We use strong random augmentation [2], including random cropping, horizontal flipping, color jitter, gray scaling and Gaussian blur, to generate augmented samples.

The number of codewords in each codebook is fixed to $K{=}2^4$, the dimension of each codeword is fixed to $D/M{=}16$ and the number of codebook is varying as $M{=}\{4,8,16\}$, so we can generate $\{M{\cdot}log_2K\}{=}\{16,32,64\}$ bits codes for image retrieval. We use Adam [11] as the optimizer with the initial learning rate of $5e{-}4$ for CIFAR-10 and $2e{-}4$ for NUS-WIDE and FLICKR25K, and set the weight decay of $1e{-}5$. We warm up the learning rate with 10 epochs and decay it with the cosine decay schedule [17] without restart. On CIFAR-10, we set the batch size $N_B{=}256$ with the original input image size of 32×32, while on NUS-WIDE and FLICKR25K, $N_B{=}128$ with the input of 224×224.

In part consistent quantization, we set $\lambda_{pn}{=}0.1$, $\lambda_{cd}{=}0.2$, $N_k{=}20$, $\tau_{pn}{=}0.5$. In global consistent quantization, we set $\tau_{sq}{=}0.2$ and $\tau_{ic}{=}0.5$ following [9], and use $\lambda_{cc}{=}0.4$ and $\tau_{cc}{=}0.2$. In fully unsupervised image retrieval, we train our model from scratch without using ImageNet pre-trained weights. Despite our approach is devised for deep fully unsupervised image retrieval, it is compatible with the deep pre-trained unsupervised setting, so we also report SSCQ-p that employs an ImageNet pre-trained VGG16 model as the backbone.

## 2.1 Details at Training Stage

During training, we add two loss terms, namely codewords diversity regularization and instance embedding contrastive loss. The former one is to stabilize the training process, while the latter one mimisc the instance quantization contrastive loss.

**Codewords Diversity Regularization.** Simply applying (??) may result in reduced diversity of the subspace features. To encourage diverse codeword distribution, we compute the similarity between sub-embedding representations and codewords in each codebook and encourage the mean probability distribution to be diverse, as:

$$\mathcal{L}_{cd} = \frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\hat{p}_{m,k}{\cdot}\log(\hat{p}_{m,k}), \tag{1}$$

where $\hat{p}_{m,k}{=}\frac{1}{2N_b}\sum_{i=1}^{2N_b}\frac{\exp(s(f_{i,m},c_{m,k}))}{\sum_{t=1}^{K}\exp(s(f_{i,m},c_{m,t}))}$ is the mean output probability over all samples in a mini-batch. Note that similar codewords diversity regularization has been used in previous quantization method [12], but here $\mathcal{L}_{cd}$ in our approach is an auxiliary term based on entropy maximization [13, 15] for unsupervised part consistent quantization and is not directly computed using soft quantization code.

**Instance Embedding Contrastive Loss.** Previous contrastive quantization based methods [9, 21] use contrastive learning for quantized representations. However, it is inevitable that quantized representations lose useful embedding representation information during the quantization process. This leads to sub-optimal performance when the feature extractor is trainable, as we found in this work. Cross-quantized learning is proposed in SPQ [9] to mitigating the effect. Unlike SPQ, we propose a much simpler alternative to maximizing the similarity between embeddings and quantized representation. Similar to $\mathcal{L}_{icz}$, we add an instance contrastive loss $\mathcal{L}_{icf}$ for the embedding representations $f$, as:

$$\mathcal{L}_{icf} = -\log \frac{\exp(s(f,f^+)/\tau_{ic})}{\sum_{j=1}^{2N_b} 1_{[f_j \neq f]} \exp(s(f,f_j)/\tau_{ic})}. \tag{2}$$

With Eq.(2), we can simultaneously optimize the quantized representations and the embedding representations.

### 2.2 Details at Inference Stage

Once the model is trained, it could be deployed for inference purpose. In inference, following the previous work [9, 21], we use hard quantization to generate the $(M \cdot log_2 K)$-bits code for each sample in the database by finding the most similar codeword $\{c_{m,k}\}_{k=1}^{K}$ from each codebook $\{C_m\}_{m=1}^{M}$ for each sub-embedding representation. Then, we use asymmetric distance [10] to measure the distance between each query sample and database samples. Specifically, given a query image, we extract its embedding representation and divide it into $M$ sub-embedding representations. Next, we compute the Euclidean distance between each sub-embedding representation and all codewords in all codebooks to set up a query-specific look-up table. Finally, we can approximately calculate the distance between the query sample and each database sample by using the code to get the sub-vector distance from the query-specific look-up table and then summing up.

### 2.3 Summary of the Proposed Method

We summarize the training process of the proposed Self-Supervised Consistent Quantization in Algorithm 1.

## 3 Complete Evaluation Results

Complete evaluation results are shown in Table 1 on CIFAR-10, in Table 2 on NUS-WIDE, in Table 3 on FLICKR25K.

In Fig. 1, we report PR curves and P@1000 curves. It can be observed that our SSCQ (blue curve) consistently outperforms SPQ (green curve) under the fully unsupervised setting, while our SSCQ-p (orange curve) performs competitively against the state-of-the-art pre-trained methods. This further demonstrate that our approach is capable of learning effective embeddings and codes for image retrieval at different required recall rates and numbers of top returned samples.

## 4 Further Analysis and Discussion

**Codeword Diversity Regularization Variants.** In Fig. 2 (left), we test SSCQ with different codeword regularization strategies, where $\mathcal{L}_{cd-soft}$ and $\mathcal{L}_{cd-ed}$ denote soft quantization and squared Euclidean distance in Eq.(1). $\mathcal{L}_{cd-spro}$ denotes squared probability [12] in Eq.(1). We observe that SSCQ with entropy maximization $\mathcal{L}_{cd}$ achieves encouraging result.

**Algorithm 1** Self-Supervised Consistent Quantization.

```
Input: A baseline model, unlabeled training data $\mathcal{X}$
for sampled mini-batch $\{x_i\}_{i=1}^{N_b}$ do
    Generate two augmented samples for each $x$
    Extract embedding representation $f$ of all samples
    Extract quantized representation $z$ of all samples
    Compute $\mathcal{L}_{icz}$ for $z$
    Compute $\mathcal{L}_{icf}$ for $f$
    /* Part consistent quantization */
    Compute $\mathcal{L}_{pn}$ for $z$
    /* Global consistent quantization */
    Compute $\mathcal{L}_{cc}$ for fused $\phi(f,z)$
    /* Codeword diversity loss*/
    Compute $\mathcal{L}_{cd}$ for $f_m$ and $c_{m,k}$
    /* Unified learning objective */
    Optimize the model with $\mathcal{L}$
end for
end for
Output: A trained model for image retrieval.
```

**Representation Fusion Variants.** In Fig. 2 (right), we report the performance using different embedding and quantized representations fusion strategies, including concatenation, summation, cross consistent contrastive regularization, and quantized representations only. We observe that SSCQ with $\mathcal{L}_{cc-con}$ and $\mathcal{L}_{cc-sum}$ yield better results than using $\mathcal{L}_{cc-cro}$ and $\mathcal{L}_{cc-qua}$ as [22].

**Temperature Parameter Sensitivity.** We evaluate the performance of our SSCQ with the temperature parameters. SSCQ is robust to the values of $\tau_{cc}$, $\tau_{pn}$ and $\tau_{sq}$, and performs competitively. Since $\tau_{ic}$ relates to the basic component of contrastive quantization, it is more sensitive and gives competitive result when set at 0.5. Detailed evaluations can be found in the *supp. mat.* for the completeness.

**Effect of $N_k$ in $\mathcal{L}_{pn}$.** Most hyper-parameters are set following SPQ [9], while $N_k$ and weighting parameters are empirically selected. In Table 4, we report the performance using different values for $N_k$ on CIFAR-10.

**Qualitative Visualizations.** In Fig. 3, the tSNE visualizations show that the class-wise distribution has a better separability after applying part loss. We also visualize some retrieval results of our SSCQ and SPQ [9] in Fig. 4. We can see that both SSCQ and SPQ can retrieve visually similar images from the database, but SSCQ is capable of exploring more discriminative information and results in more relevant retrieval results with higher accuracy.

## References

[1] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

| Type | Method | 16 bits | 32 bits | 64 bits |
|---|---|---|---|---|
| Shallow + pre-trained | LSH [1] | 13.2 | 15.8 | 16.7 |
| | SpectralH [23] | 27.2 | 28.5 | 30.0 |
| | PQ [10] | 23.7 | 25.9 | 27.2 |
| | ITQ [7] | 30.5 | 32.5 | 34.9 |
| | OPQ [6] | 29.7 | 31.4 | 32.3 |
| Deep pre-trained unsupervised | DeepBit [16] | 22.0 | 24.9 | 27.7 |
| | SAH [5] | 41.8 | 45.6 | 47.4 |
| | GreedyHash [20] | 44.8 | 47.3 | 50.1 |
| | SSDH [24] | 36.2 | 40.2 | 44.0 |
| | TBH [19] | 53.2 | 57.3 | 57.8 |
| | CIBHash [18] | 59.4 | 63.7 | 65.2 |
| | Bi-half [14] | 56.1 | 57.6 | 59.5 |
| | MeCoQ [21] | 68.2 | 69.7 | 71.1 |
| | SSCQ-p (ours) | **76.1** | **76.8** | **78.1** |
| Deep fully unsupervised | SGH [3] | 43.5 | 43.7 | 43.3 |
| | HashGAN [4] | 44.7 | 46.3 | 48.1 |
| | BinGAN [25] | 47.6 | 51.2 | 52.0 |
| | SPQ [9] | 76.8 | 79.3 | 81.2 |
| | SSCQ (ours) | **78.3** | **81.3** | **82.9** |

Table 1: Comparison with the classic and state-of-the-art unsupervised methods on CIFAR-10 in terms of mAP (%). Some results are cited from [9, 21].

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[3] Bo Dai, Ruiqi Guo, Sanjiv Kumar, Niao He, and Le Song. Stochastic generative hashing. In *International Conference on Machine Learning*, pages 913–922. PMLR, 2017.

[4] Kamran Ghasedi Dizaji, Feng Zheng, Najmeh Sadoughi, Yanhua Yang, Cheng Deng, and Heng Huang. Unsupervised deep generative adversarial hashing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3664–3673, 2018.

[5] Thanh-Toan Do, Dang-Khoa Le Tan, Trung T Pham, and Ngai-Man Cheung. Simultaneous feature aggregating and hashing for large-scale image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6618–6627, 2017.

[6] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE CVPR*, pages 2946–2953, 2013.

[7] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image

| Type | Method | 16 bits | 32 bits | 64 bits |
|---|---|---|---|---|
| Shallow + pre-trained | LSH [1] | 38.5 | 41.4 | 43.9 |
| | SpectralH† [23] | 48.9 | 53.0 | 62.7 |
| | PQ [10] | 65.4 | 67.4 | 68.6 |
| | ITQ† [7] | 68.0 | 70.9 | 72.8 |
| | OPQ [6] | 65.7 | 68.4 | 69.1 |
| Deep pre-trained unsupervised | DeepBit [16] | 39.2 | 40.3 | 42.9 |
| | GreedyHash [20] | 63.3 | 69.1 | 73.1 |
| | SSDH [24] | 58.0 | 59.3 | 61.0 |
| | CIBHash† [18] | 79.5 | 81.2 | 81.7 |
| | Bi-half [14] | 76.9 | 78.3 | 79.9 |
| | MeCoQ† [21] | 77.2 | 81.5 | 82.3 |
| | SSCQ-p (ours) | **80.3** | **81.9** | **82.6** |
| Deep fully unsupervised | SGH [3] | 59.3 | 59.0 | 60.7 |
| | HashGAN [4] | 68.4 | 70.6 | 71.7 |
| | BinGAN [25] | 65.4 | 70.9 | 71.3 |
| | SPQ† [9] | 75.7 | 79.4 | 80.2 |
| | SSCQ (ours) | **78.7** | **79.9** | **80.8** |

Table 2: Comparison with the classic and state-of-the-art unsupervised methods on NUS-WIDE in terms of mAP (%). Some results are cited from [21]. † Reproduced results.

retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12): 2916–2929, 2012.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Young Kyun Jang and Nam Ik Cho. Self-supervised product quantization for deep unsupervised image retrieval. In *Proceedings of the IEEE ICCV*, pages 12085–12094, 2021.

[10] 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.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Benjamin Klein and Lior Wolf. End-to-end supervised product quantization for image search and retrieval. In *Proceedings of the IEEE CVPR*, pages 5041–5050, 2019.

[13] Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in Neural Information Processing Systems*, 23, 2010.

| Type | Method | 16 bits | 32 bits | 64 bits |
|---|---|---|---|---|
| Shallow + pre-trained | LSH [1] | 58.8 | 60.4 | 64.2 |
| | SpectralH [23] | 59.2 | 60.6 | 63.2 |
| | ITQ [7] | 68.4 | 69.5 | 70.3 |
| Deep pre-trained unsupervised | GreedyHash [20] | 70.5 | 72.3 | 75.1 |
| | SSDH [24] | 78.7 | 79.4 | 79.5 |
| | CIBHash [18] | 77.0 | 78.5 | 79.8 |
| | Bi-half [14] | 81.1 | 82.4 | **82.9** |
| | MeCoQ [21] | 80.4 | 81.7 | 81.7 |
| | SSCQ-p (ours) | **81.9** | **82.6** | 82.8 |
| Deep fully unsupervised | SPQ [9] | 71.8 | 74.0 | 74.5 |
| | SSCQ (ours) | **73.8** | **75.9** | **76.7** |

Table 3: Comparison with the classic and state-of-the-art unsupervised methods on FLICKR25K in terms of mAP (%).

| $N_k$ | 0 | 1 | 10 | 20 | 50 |
|---|---|---|---|---|---|
| mAP(%) | 80.1 | 80.7 | 81.4 | 81.3 | 80.5 |

Table 4: Effect of $N_k$ in loss $\mathcal{L}_{pn}$ on CIFAR-10 (32 bits).

[14] Yunqiang Li and Jan van Gemert. Deep unsupervised image hashing by maximizing bit entropy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2002–2010, 2021.

[15] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[16] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1183–1192, 2016.

[17] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[18] Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. Unsupervised hashing with contrastive information bottleneck. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[19] Yuming Shen, Jie Qin, Jiaxin Chen, Mengyang Yu, Li Liu, Fan Zhu, Fumin Shen, and Ling Shao. Auto-encoding twin-bottleneck hashing. In *Proceedings of the IEEE CVPR*, pages 2818–2827, 2020.

[20] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. *Advances in Neural Information Processing Systems*, 31, 2018.

[21] Jinpeng Wang, Ziyun Zeng, Bin Chen, Tao Dai, and Shu-Tao Xia. Contrastive quantization with code memory for unsupervised image retrieval. In *Proceedings of the AAAI*, 2022.

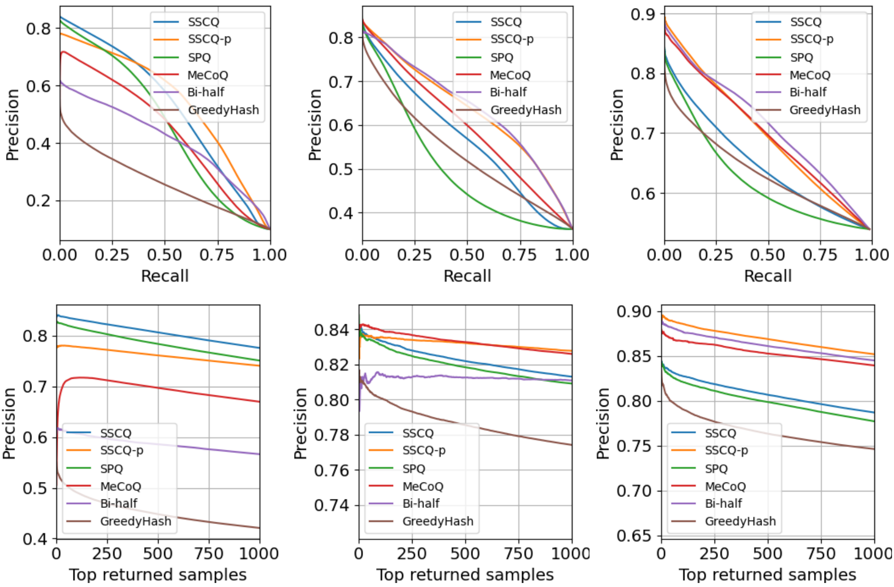

Figure 1: PR curves (*Top*) and P@1000 curves (*Bottom*) on CIFAR-10, NUS-WIDE and FLICKR25K (32 bits).

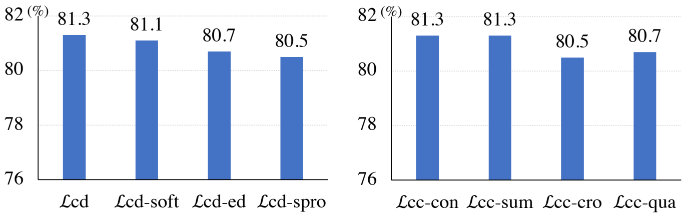

Figure 2: Evaluating (*Left*) codeword diversity regularization variants and (*Right*) representation fusion variants on CIFAR-10 (32 bits).

[22] Chen Wei, Huiyu Wang, Wei Shen, and Alan Yuille. Co2: Consistent contrast for unsupervised visual representation learning. In *International Conference on Learning Representations*, 2021.

[23] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *Advances in Neural Information Processing Systems*, 21, 2008.

[24] Erkun Yang, Cheng Deng, Tongliang Liu, Wei Liu, and Dacheng Tao. Semantic structure-based unsupervised deep hashing. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 1064–1070, 2018.

[25] Maciej Zieba, Piotr Semberecki, Tarek El-Gaaly, and Tomasz Trzcinski. Bingan: Learning compact binary descriptors with a regularized gan. *Advances in Neural Information Processing Systems*, 31, 2018.

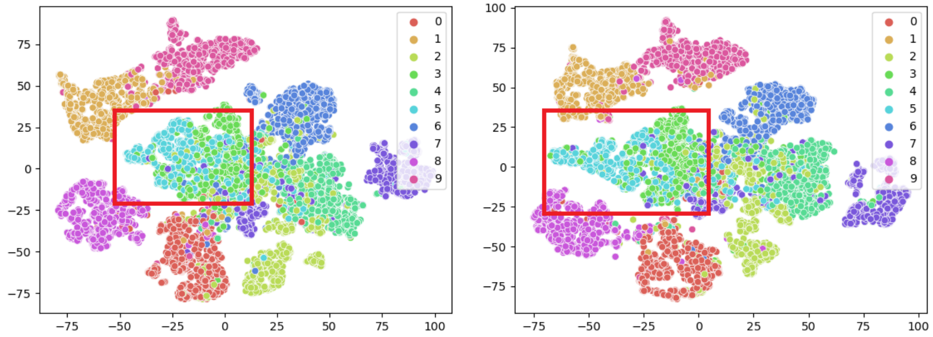

Figure 3: tSNE on CIFAR-10 validation queries for $\mathcal{L}_{icz}+\mathcal{L}_{icf}$ (left) and $\mathcal{L}_{icz}+\mathcal{L}_{icf}+\mathcal{L}_{pn}$ (right). *Cat* (class 3) and *Dog* (class 5) show better separability after applying part loss as highlighted in the red bounding box.

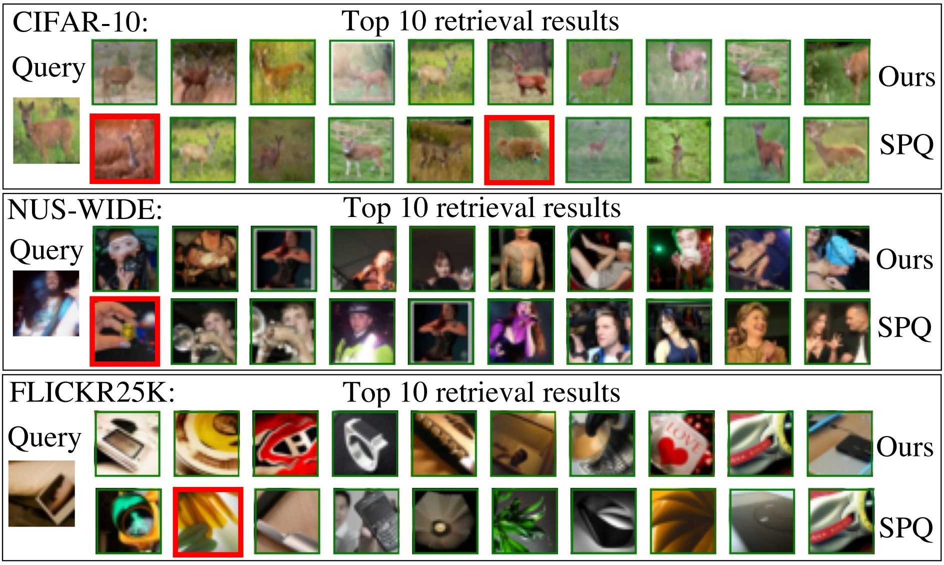

Figure 4: Retrieval results of our approach and SPQ on CIFAR-10, NUS-WIDE and FLICKR25K (32 bits). False retrieval results are denoted in red bounding boxes.