Supplementary Material: Correlation Verification for Image Retrieval

Seongwon Lee Hongje Seong Suhyeon Lee Euntai Kim*
School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
{won4113, hjseong, hyeon93, etkim}@yonsei.ac.kr

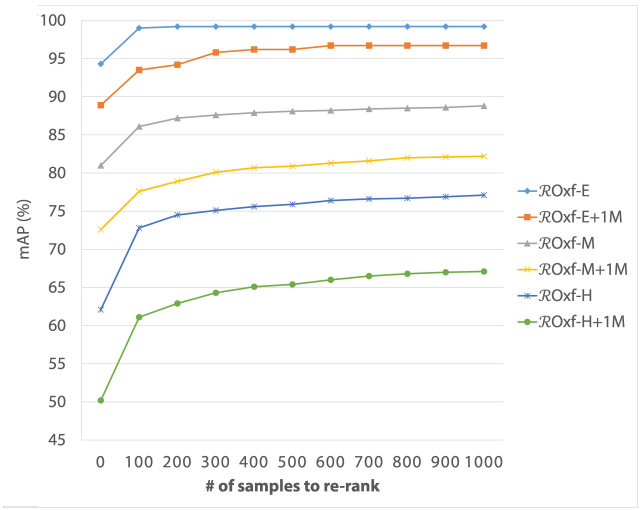
S1. Data Selection and Sampling Process

Overlapped positive selection. In this study, we use the 'clean' subset [18] of Google Landmarks dataset v2 (1.58M images from 81k landmarks) [16] as a training set. This dataset has large intra-class variability and includes multiple viewpoints, such as indoor and outdoor views of landmarks. Therefore, when sampling the same-class image pair from this dataset, we cannot guarantee an overlap between the two images, and non-overlapping query-positive pairs can interfere with learning image matching. To avoid the non-overlapping case, we select overlapped pairs for each class in advance with the help of the DELF [9] local feature. The overall process is similar to the data cleaning process of [18]: The primary difference is that [18] aims to remove outlier data from the dataset, whereas we aim to select same-class pairs that actually overlap. To select an overlapped pair, for every dataset sample x_i, we first select up to ten of the nearest neighbors that are assigned to the same class as x_i with a global descriptor extracted from R50-CVNet-Global. After the nearest neighbors are selected, spatial verification using RANSAC with a pre-trained DELF feature is performed on the nearest neighbors selected for each sample. Subsequently, we select the pair with 30 or more inlier matches as an overlapped pair. Furthermore, only classes with more than 10 samples belonging to overlapped pairs are used for training. Finally, we select 1M images from 31k landmarks of the GLDv2-clean dataset and use this subset as a training set for CVNet-Rerank. Although this selection process is quite expensive because of the use of RANSAC, it only needs to be performed once.

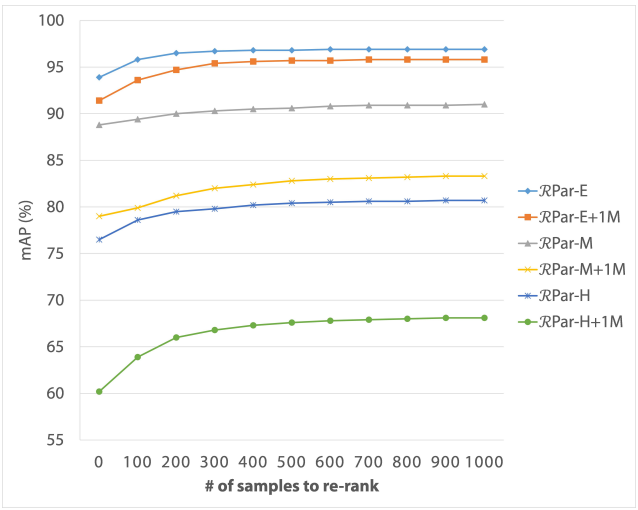
Sampling process. CVNet-Rerank is trained for 200 epochs (6.3M steps) for all selected classes. For every epoch, we construct tuples of query, positive, and negative samples for each class. The query image is randomly sampled from each class, a positive image is randomly chosen from among the overlapped positives of the query, and a negative image is sampled from random or hard-negative

Figure S1. Example of query, overlapped positive, and hard negative samples sampled from the selected subset of the GLDv2-clean dataset. Our proposed re-ranking network learns better discrimination ability by learning the cue for equivalence from a overlapped positive and the cue for the difference from a hard negative.

*Corresponding author.



(a) $\mathcal{R}$Oxford5k and its +1M Experiments.



(b) $\mathcal{R}$Paris6k and its +1M Experiments.

Figure S2. **Analysis about number of samples to re-rank.**

rerank	r_H	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0		81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	0.0	81.6	72.8	88.8	79.0	62.6	50.2	76.6	60.3
	1.0	85.5	77.1	89.3	**80.0**	70.8	59.5	77.5	63.7
	0.5	85.5	77.3	89.2	79.8	71.5	60.5	77.9	63.4
	0.2-1.0	**86.1**	**77.6**	**89.4**	79.9	**72.8**	**61.1**	**78.6**	**63.9**
200	0.0	81.5	72.7	88.8	79.0	62.6	50.1	76.7	60.4
	1.0	86.2	78.2	89.5	81.1	71.5	60.7	76.5	65.3
	0.5	86.4	78.4	89.7	80.8	73.1	62.0	78.7	65.3
	0.2-1.0	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
400	0.0	81.4	72.6	88.9	79.1	62.6	50.1	77.1	60.6
	1.0	86.2	79.5	88.5	81.8	71.0	62.0	74.0	65.5
	0.5	86.9	79.8	90.3	82.0	74.0	63.9	79.5	66.7
	0.2-1.0	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**

Table S1. **Hard-Negative Sampling Ratio.**

samples according to the hard negative sampling ratio r_{neg}. Fig. S1 shows examples of our sampling results. By learning with well-constructed training pairs, the network can achieve improved discriminating ability.

S2. Additional Ablation Studies and Analysis

S2.1. Curriculum Learning

Learning focused on hard samples can improve the robustness of the network in hard situations. However, this could lead to a loss of generality. Accordingly, we apply curriculum learning to focus on hard samples without losing generality. In this subsection, we show that the proposed network performs re-ranking well regardless of the matching difficulty with the help of curriculum learning. Furthermore, we show a more detailed analysis of curriculum learning.

Generality of learning (Fig. S2). By gradually increasing the number of samples to be re-ranked, we can verify whether the network distinguishes hard samples well

rerank	p_{has}	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0		81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	0.0	85.8	77.5	89.3	79.9	71.6	60.5	78.1	63.7
	0.2	86.1	77.1	89.3	79.9	72.3	60.1	78.1	63.6
	0.0-0.2	**86.1**	**77.6**	**89.4**	79.9	**72.8**	**61.1**	**78.6**	**63.9**
200	0.0	86.9	78.7	89.7	81.0	73.4	62.1	78.6	65.6
	0.2	87.1	78.3	89.7	81.1	74.0	61.8	78.7	65.7
	0.0-0.2	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
400	0.0	87.5	80.3	89.9	82.0	74.2	64.3	78.9	66.4
	0.2	87.8	80.1	90.1	82.2	75.1	64.0	79.3	66.8
	0.0-0.2	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**

Table S2. **Hide-and-Seek Probability.**

while retaining generality for normal samples. As shown in Fig. S2, the proposed re-ranking network dramatically improves performance when it is applied to top ranks where many hard samples exist. Even if the re-ranking targets are expanded to easier samples, our proposed re-ranking model continues to exhibit improved performance without losing generality.

Hard negative mining (Tab. S1). To prove the effectiveness of hard negative mining applied simultaneously with the curriculum approach, we conduct experiments by varying the hard-negative sampling ratio r_H. The results are presented in Tab. S1. When the network learns using randomly sampled negatives ($r_H = 0$), global retrieval results do not improve when re-ranking. This indicates that learning to discriminate hard samples using only random negative is difficult. Accordingly, when sampling hard negatives with a fixed ratio ($r_H = 1.0, 0.5$), the network exhibits a significantly improved performance. Moreover, when a hard-negative ratio is set through the curriculum manner (r_H = 0.2-1.0), the proposed re-ranking network exhibits its best performance. This proves that hard negatives are a critical key to re-ranking learning, and hard negative mining

#	C_l'	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0		81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	16	85.4	76.3	89.2	79.7	70.9	57.9	77.5	62.5
	32	85.5	76.5	89.2	79.8	71.7	59.7	77.6	63.1
	64	85.4	76.8	89.3	79.9	71.6	60.2	77.6	63.3
	128	85.5	76.9	89.3	79.9	71.4	60.0	77.9	63.6
	256	**86.1**	**77.6**	**89.4**	**79.9**	**72.8**	**61.1**	**78.6**	**63.9**
	512	85.5	76.9	89.3	79.9	71.3	59.7	77.8	63.7
	1024	85.7	77.3	89.4	80.0	71.6	59.7	78.1	63.7
200	16	86.2	77.2	89.6	80.5	71.9	58.8	78.1	63.9
	32	86.5	77.6	89.6	80.7	73.2	61.1	78.1	64.5
	64	86.2	77.9	89.7	80.9	72.9	61.3	77.9	64.9
	128	86.4	78.0	89.9	81.2	72.8	61.2	78.6	65.5
	256	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
	512	86.3	77.9	89.8	81.1	72.6	61.0	78.4	65.6
	1024	86.5	78.6	89.9	81.1	72.6	61.2	79.1	65.5
400	16	86.5	78.4	89.9	81.2	72.3	60.2	78.5	64.5
	32	87.0	79.1	89.7	81.5	74.1	63.0	78.2	65.1
	64	86.7	79.3	89.8	81.7	73.6	63.3	78.1	65.6
	128	87.0	79.6	90.1	82.1	73.5	63.2	78.9	66.3
	256	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**
	512	86.6	79.3	90.0	82.0	73.1	62.8	78.5	66.3
	1024	87.0	80.1	90.3	82.1	73.4	63.0	79.6	66.6

Table S3. **Channel Compression.**

is even more effective when used with curriculum learning.

Hide-and-Seek (Tab. S2). Similarly, to prove the effectiveness of the Hide-and-Seek [13] augmentation, we conduct experiments by varying the Hide-and-Seek probability p_{has}. Tab. S2 also shows that Hide-and-Seek is an appropriate strategy to help re-ranking learning and that it can be even more effective when used with curriculum learning.

S2.2. Memory Footprint Reduction

Despite having significant potential, the proposed re-ranking method possesses a large memory owing to its dense nature. In this subsection, we present several effective solutions for reducing the memory footprint of the proposed re-ranking model.

Channel compression (Tab. S3). We pre-extract and store a multi-scale feature pyramid for every database sample for online re-ranking, which is where memory consumption primarily occurs. To reduce the memory footprint of the proposed model, we compress the channel of the feature map C_l to C_l' using a 3×3 convolution layer in the process of constructing the multi-scale feature pyramid. Here, we conduct experiments by varying the compressed channel dimension C_l', to find a balance between memory footprint and re-ranking performance. The results are presented in Tab. S3. When the C_l' is 256, the proposed re-ranking model exhibited its best performance; therefore, we finally selected C_l' as 256 in our study. However, on systems where memory management is more important, choosing a smaller dimension, such as 16 ($\frac{1}{16}$ of our model's memory footprint) or 32 ($\frac{1}{8}$ of our model's memory footprint) can be a good option. This quantization reduces the memory footprint even further, albeit at the cost of a marginally reduced performance.

#	type	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0		81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	float32	**86.1**	**77.6**	**89.4**	**79.9**	**72.8**	**61.1**	**78.6**	**63.9**
	int8	**86.1**	**77.6**	**89.4**	**79.9**	72.7	**61.1**	**78.6**	**63.9**
	int4	86.0	77.3	89.3	79.8	72.5	60.5	78.0	63.5
200	float32	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
	int8	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	62.8	**79.5**	**66.0**
	int4	86.9	78.6	89.7	81.0	73.8	62.1	78.7	65.3
400	float32	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**
	int8	**87.9**	80.6	**90.5**	**82.4**	75.5	**65.1**	**80.2**	**67.3**
	int4	87.4	80.1	90.1	82.0	74.6	63.8	79.3	66.3

Table S4. **Feature Quantization.**

#	layer	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0		81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	f_3	**86.1**	**77.6**	**89.4**	**79.9**	**72.8**	**61.1**	**78.6**	**63.9**
	f_4	85.4	76.0	89.2	79.8	69.0	56.2	76.7	63.0
	fuse	85.2	76.8	89.3	79.9	69.6	57.6	77.0	63.2
200	f_3	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
	f_4	85.8	76.7	89.2	80.7	69.7	57.0	75.5	63.8
	fuse	85.4	77.5	89.4	80.9	69.8	58.2	75.8	64.0
400	f_3	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**
	f_4	86.0	77.7	88.2	80.9	69.8	58.0	73.7	63.2
	fuse	85.1	78.3	88.2	80.9	68.9	58.3	74.0	63.3

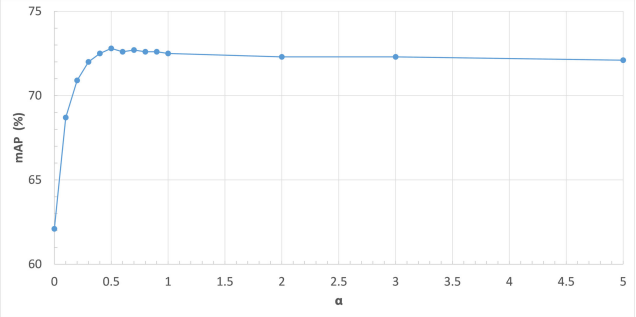
Table S5. **Feature Extraction Layer Selection.**

Quantization (Tab. S4). To reduce the memory burden, we measured the re-ranking performance while taking the correlation of quantized features as an input. The results are presented in Tab. S4. Similar to the case of channel compression, feature quantization also reduces the memory footprint at the risk of marginal performance degradation.

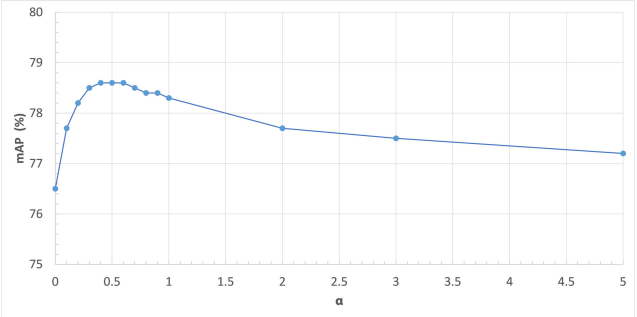
S2.3. Model Design and Parameter Selection

In this subsection, we present several analyses of the design of the re-ranking model and its parameter selection.

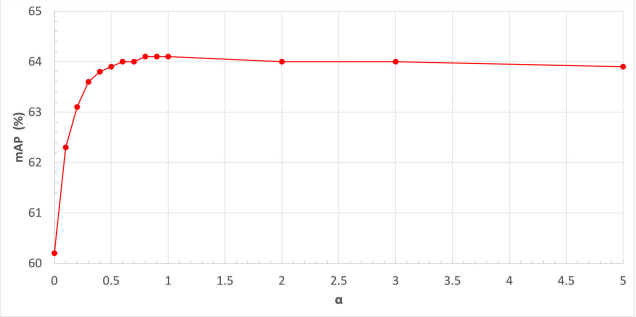
Feature extraction layer selection (Tab. S5). First, we analyze CVNet-Global to determine which of its stages is more suited for use as an input for the re-ranking network. The results are presented in Tab. S5. f_i denotes the ith *ResBlock*. When receiving an output of f_4 as an input, the stride and kernel size in the first block are reduced by 1 and 3, respectively; therefore, the output resolution is identical to that when an output of f_3 is received as an input. In the "fuse" case, both the output feature maps of f_3 and f_4 are received as input. In this case, the outputs of f_3 and f_4 pass through the first two convolutional blocks separately and



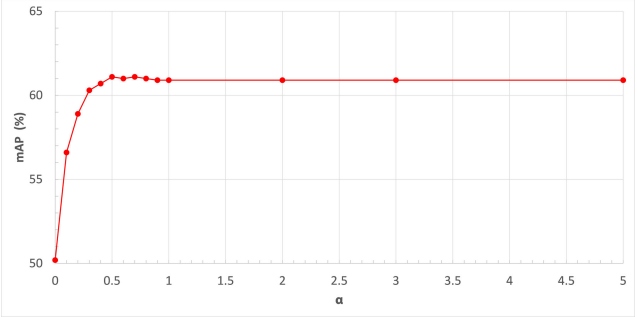
(a) $\mathcal{R}$Oxford5k-Hard Experiments.



(b) $\mathcal{R}$Paris6k-Hard Experiments.



(c) $\mathcal{R}$Oxford5k-Hard+1M Experiments.



(d) $\mathcal{R}$Paris6k-Hard+1M Experiments.

Figure S3. **Experiments about the score fusion weight α.** α is tuned in $\mathcal{R}$Oxf-Hard (Fig. S3a)/$\mathcal{R}$Par-Hard (Fig. S3b) and fixed for $\mathcal{R}$Oxf-Hard+1M (Fig. S3c)/$\mathcal{R}$Par-Hard+1M (Fig. S3d). We finally set an α to 0.5.

#	Scale			Medium				Hard			
	1	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0				81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	✓			84.9	76.1	88.8	79.3	69.9	57.4	76.3	61.1
	✓	✓		85.8	77.1	89.3	**80.0**	71.5	59.8	78.3	63.9
	✓		✓	85.7	77.3	89.3	79.9	71.1	59.9	78.0	63.4
	✓	✓	✓	**86.1**	**77.6**	**89.4**	79.9	**72.8**	**61.1**	**78.6**	**63.9**
200	✓			85.3	76.7	88.9	79.5	70.5	58.3	76.3	61.5
	✓	✓		86.6	78.2	90.0	81.2	72.8	61.3	79.0	**66.0**
	✓		✓	86.5	78.5	89.9	81.0	72.2	61.1	78.7	65.3
	✓	✓	✓	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
400	✓			85.5	77.6	89.0	79.7	70.7	59.3	76.4	61.6
	✓	✓		87.1	79.9	90.3	**82.4**	73.6	63.4	79.3	67.2
	✓		✓	87.0	80.2	90.3	82.0	72.9	63.1	79.0	66.4
	✓	✓	✓	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**

Table S6. **Scale Selection.**

merged, and finally pass through the remaining blocks. As in many studies [1, 9, 17] utilizing local information, using the output of f_3 as an input results in the best performance; thus, we select the feature map from f_3 as the input of the re-ranking network.

Scale selection (Tab. S6). We conduct experiments with a selection of scales to construct a multi-scale feature pyramid. Note that a high-scale feature can be helpful in terms of performance. However, considering the limitation of time and memory, we only scale the feature to a lower scale. The results show that constructing a cross-scale correlation using several scales has a clear performance advantage over the single-scale feature correlation method. Based on the experimental results, we finally select $S = 3$ scales.

#	kernel	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0		81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
100	asymmetric	**86.1**	**77.6**	**89.4**	**79.9**	**72.8**	**61.1**	**78.6**	**63.9**
	symmetric	85.2	76.8	89.3	79.9	69.6	57.6	77.0	63.2
200	asymmetric	**87.2**	**78.9**	**90.0**	**81.2**	**74.5**	**62.9**	**79.5**	**66.0**
	symmetric	85.4	77.5	89.4	80.9	69.8	58.2	75.8	64.0
400	asymmetric	**87.9**	**80.7**	**90.5**	**82.4**	**75.6**	**65.1**	**80.2**	**67.3**
	symmetric	85.1	78.3	88.2	80.9	68.9	58.3	74.0	63.3

Table S7. **Kernel Symmetrization.**

Symmetric kernel (Tab. S7). Image similarity is essentially permutation-invariant, except in special cases. When we train a 4D convolutional network to predict image similarity, we can induce the network to be permutation-invariant in several ways. For instance, we can set the loss function to ensure that the output does not vary regardless of the input order. Another method is to make the 4D convolution kernel symmetrical. We experiment with the latter case as shown in Tab. S7. However, forcing the kernel to be symmetric did not yield good performance. Therefore, in this study, we softly induce permutation-invariant properties in the re-ranking network using loss symmetrization.

Score fusion weight (Fig. S3). To simultaneously verify the global and local relationships between two images, we re-rank the retrieval results based on the combined score $s_g + \alpha s_r$, where s_g is the cosine similarity of the global de-

#	layer	Medium				Hard			
		$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
0	Global	81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
0	αQE	85.4	77.5	**90.7**	**83.5**	67.5	57.8	79.8	66.9
100	CV	86.1	77.6	89.4	79.9	72.8	61.1	78.6	63.9
200	CV	87.2	78.9	90.0	81.2	74.5	62.9	79.5	66.0
400	CV	**87.9**	**80.7**	90.5	82.4	**75.6**	**65.1**	**80.2**	**67.3**
0	αQE	85.4	77.5	90.7	83.5	67.5	57.8	79.8	66.9
100	αQE + CV	88.0	80.5	90.9	84.1	74.6	65.2	80.9	70.0
200	αQE + CV	88.8	82.1	91.2	84.9	75.9	67.4	81.6	71.5
400	αQE + CV	**89.3**	**82.8**	**91.6**	**85.3**	**77.1**	**68.6**	**82.2**	**70.7**

Table S8. **Comparison with alphaQE.**

scriptors, s_r is the output score of the re-ranking network, and α is the given weight for the re-ranking network output score s_r. Parameter α is tuned in $\mathcal{R}$Oxf/$\mathcal{R}$Par and fixed for a large-scale experiment and GLDv2-retrieval test, as in previous studies [1, 8, 12, 15]. Fig. S3a and Fig. S3b shows $\mathcal{R}$Oxf-Hard/$\mathcal{R}$Par-Hard performances according to score fusion weight α. In these results, the re-rank score significantly improves the retrieval performance even if an extremely small re-rank score is added to the global descriptor matching score. Moreover, the best performance corresponded to an α value of 0.5. Based on these experimental results, we set $\alpha = 0.5$ for the re-ranking process.

S2.4. Comparison with Query Expansion

Comparison with αQE (Tab. S8). This study focuses on improving the image matching ability for single pairs. Therefore, we have not considered certain re-ranking methods such as diffusion [2,5] or query expansion [3,11], which require additional expenses to traverse the entire database mentioned in the main body of this paper. Although we do not consider them because of their different scopes, in this subsection we show that these re-ranking methods and the proposed re-ranking method can be harmoniously fused when they are used. Specifically, we compared and fused CV with one of the representative query expansion methods: α-weighted query expansion (αQE).

In contrast to geometric verification (GV) or our proposed correlation verification (CV), which evaluates the similarity between *two images*, the query expansion aggregates the query itself and its top-ranked neighbors *across the dataset* and creates an aggregated query to perform re-ranking. In the αQE method, aggregation is performed with weighted averaging, and the weight of the ith ranked image is given by $(\mathbf{d}_q \cdot \mathbf{d}_i)^{\alpha_{QE}}$, where $\mathbf{d}_q$ is the global descriptor of the query image and $\mathbf{d}_i$ is the global descriptor of the ith ranked image for the query. Finally, the aggregated query descriptor $\mathbf{d}'_q$ is computed as follows:

$$\mathbf{d}'_q = \frac{\mathbf{d}_q + \sum_{i=1}^{n} \left((\mathbf{d}_q \cdot \mathbf{d}_i)^{\alpha_{QE}} \cdot \mathbf{d}_i\right)}{1 + \sum_{i=1}^{n} (\mathbf{d}_q \cdot \mathbf{d}_i)^{\alpha_{QE}}}, \tag{S1}$$

where n is the number to aggregates, and α_{QE} is a parameter that amplifies or reduces the weight. n and α_{QE} are tuned in $\mathcal{R}$Oxf/$\mathcal{R}$Par over the ranges: $n \in [1, 20]$ and $\alpha_{QE} \in [0.1, 2.0]$ and we finally set them to 5 and 2.0, respectively. Tab. S8 shows the re-ranking results using αQE, the re-ranking results using CV, and the re-ranking results using αQE and CV sequentially. For all settings, CV exhibits performance that is superior to the αQE method, and even more superior when fused with the αQE method.

K	Medium				Hard			
	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
576	77.5	70.0	**89.8**	78.0	55.7	44.6	77.9	58.9
4608	78.5	71.3	**89.8**	78.7	58.1	46.4	**78.3**	59.3
73728	**81.0**	**72.6**	88.8	**79.0**	**62.1**	**50.2**	76.5	**60.2**

Table S9. **Queue Size of Momentum Contrastive Loss.**

Loss	Medium				Hard			
	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
SupCon [6] (Eq. (S2))	79.9	72.1	**89.4**	78.9	59.1	48.6	**77.4**	59.3
Ours (Eq. (S3))	**81.0**	**72.6**	88.8	**79.0**	**62.1**	**50.2**	76.5	**60.2**

Table S10. **Comparison with SupCon Loss.**

S2.5. Momentum Contrastive Loss Analysis

Queue size (Tab. S9). Our global backbone network, CVNet-Global, constructs a queue to store and leverage numerous samples for contrastive learning. Because queue size is one of the crucial factors that is directly related to the number of contrastive samples, we conduct experiments by varying the queue size K. Tab. S9 shows performances for different queue sizes. Overall, our global model benefits from a large K value. A large queue size implies that several contrastive samples can be utilized, which can lead the global model to learn a more generalized representation.

Differences in SupCon loss (Tab. S10). Our momentum contrastive loss is similar to SupCon [6] loss. Similar to the SupCon loss, it performs contrast learning with multiple positives using labels. The primary difference between these losses is that the SupCon loss assumes a relatively constant number of positives. However, a large difference exists in the number of positives for each sample because of the class imbalance data and queue structure. In SupCon loss (Eq. (S2)), because all query-positive cosine similarities are included in the denominator, the scale of the loss is significantly affected by the number of positives:

$$\mathcal{L}_s = \frac{-1}{|P(q)|} \sum_{p \in P(q)} \log \frac{\exp\left(\bar{C}\left(\mathbf{d}_g^q \cdot \bar{\mathbf{d}}_g^p, 1\right)/\tau\right)}{\sum_{i \in P(q) \bigcup N(q)} \exp\left(\bar{C}\left(\mathbf{d}_g^q \cdot \bar{\mathbf{d}}_g^i, \mathbb{1}_q^i\right)/\tau\right)}. \tag{S2}$$

To solve this scale problem, we design our contrastive loss (Eq. (S3)) similar to the SupCon loss $\mathcal{L}_s$. However, only the target positive p is included in the denominator.

model	Medium				Hard			
(Rerank top-100)	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
R50-DELG†	71.1	60.4	86.9	70.9	47.0	32.0	73.6	48.1
+ CVNet-Rerank	**78.7**	**67.7**	**87.9**	**72.3**	**63.0**	**46.1**	**76.8**	**52.5**
R50-DOLG†	79.0	70.0	88.3	76.2	57.5	43.2	75.0	55.4
+ CVNet-Rerank	**83.7**	**74.9**	**89.0**	**77.2**	**69.1**	**55.7**	**77.1**	**59.0**

Table S11. **Performance with Different Backbones.**

model (R50)	Medium				Hard			
(Rerank top-100)	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
CVNet-Global	81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
+ CorrNet	81.3	72.7	88.8	79.0	62.3	50.3	76.5	60.2
+ HNM	84.6	75.9	89.0	79.3	69.3	57.0	76.9	61.1
+ CSC	85.8	77.5	89.3	**79.9**	71.6	60.5	78.1	63.7
+ HaS	**86.1**	**77.6**	**89.4**	**79.9**	**72.8**	**61.1**	**78.6**	**63.9**

Table S12. **Module Ablation Study.**

$$\mathcal{L}_{con} = \frac{-1}{|P(q)|} \sum_{p \in P(q)} \log \frac{\exp\left(\bar{\mathcal{C}}\left(\mathbf{d}_g^q \cdot \bar{\mathbf{d}}_g^p, 1\right)/\tau\right)}{\sum_{i \in \{p\} \bigcup N(q)} \exp\left(\bar{\mathcal{C}}\left(\mathbf{d}_g^q \cdot \bar{\mathbf{d}}_g^i, \mathbb{1}_q^i\right)/\tau\right)}. \tag{S3}$$

Tab. S10 shows the results of training CVNet-Global using each of the two losses. When using the proposed contrastive loss $\mathcal{L}_{con}$, the results are more stable than when using SupCon loss L_s.

S2.6. Performance with Different Backbones

Tab. S11 shows the results when CVNet-Rerank is combined with the standard global models, DELG and DOLG (both reproduced†). Both models are trained using the settings of the original paper except for setting the maximum number of epochs to 25 epochs. Afterward, the proposed CVNet-Rerank is trained with each global backbone. As shown in Tab. S11, CVNet-Rerank also works well when combined with other global backbones.

S2.7. Module Ablation Study

Tab. S12 shows the results when the components of CVNet-Rerank are added to CVNet-Global one by one. From the Tab. S12, we can see that when the Correlation encoding Network (CorrNet) is added, the accuracy is slightly improved but when Hard Negative Mining (HNM) is applied, the accuracy is significantly improved. From the observation, we can see that the correlation encoding network and hard negative mining is a good combination but the network is quite hard to train if the hard negative mining is not used. Based on our experience, when we train the network on random negatives without hard negative mining, the training is dominated by the very rare high correlations between the query and the random negatives degrading the accuracy. To prevent the degradation, we have to train the network using hard negative mining. In summary, we can say that the correlation encoding network and hard negative mining is the core component of the CVNet-Rerank, and the combination significantly improves the accuracy. Cross-Scale Correlation (CSC) and Hide-and-Seek (HaS) are the optional choices that can incrementally improve the accuracy of the re-ranking network.

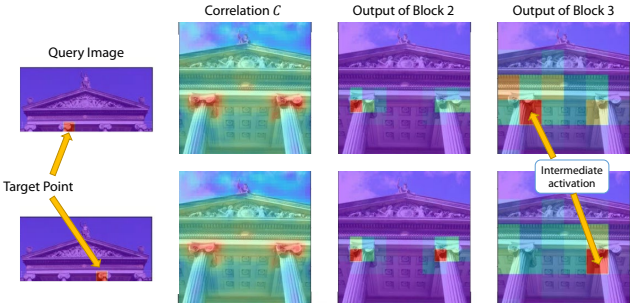


Figure S4. **Intermediate Feature Visualization.** Our network naturally learns the correct geometric relationship of dense matching and pays attention to the correct position by compressing the surrounding matching information from the 4D correlation.

S3. Intermediate Feature Visualization

We visualize the intermediate features of our re-ranking model to see how the network interprets and compresses the correlation. To visualize the intermediate 4D features, we select one target point from the query side and visualize the magnitude of the corresponding feature parts on the key side. The visualized results are presented in Fig. S4. In Fig. S4, we observe that the model focus on the correct position by compressing the surrounding matching information from the 4D correlation. As shown in the results, the network naturally learns the geometric pattern of dense matching without any predefined geometric model (*e.g.* Affine model). Additional intermediate feature visualizations are presented in Fig. S5.

S4. Reproducing Details

For a fair comparison with other re-ranking methods, we conduct experiments by reproducing other re-ranking methods based on the global backbone network. We reproduce two re-ranking methods: geometric verification (GV) and Reranking Transformer [14]. Because both methods are based on the local features of DELG, we attach the local branch of DELG [1] to our global backbone (R50-CVNet-Global) to learn the local features of DELG. All local-feature-related settings are identical to those in the DELG [1]. During testing, we extract a maximum of 1000 local features (500 for RRT) and use them for the re-ranking process.

Geometric Verification (GV). We reproduce the GV based on the DELG. Official code of DELG uses RANSAC [4], which belongs to the scikit-learn [10] package; however, we could not improve the re-ranking performance with

this version. Finally, we implement RANSAC using pydegensac [7], which exhibits performance superior to that of scikit-learn. Additionally, as mentioned in DELG [1] paper, we set a minimum number of inliers to improve re-ranking performance. We tune the minimum number of inliers over the range: [10,300], and finally set it to be 100.

Reranking Transformers (RRT). We train the RRT model with official code provided by [14], and use all the same settings are as the provided one. The only difference is that we input the global descriptor extracted from CVNet-Global and local features extracted from the added local branch, instead of the features extracted by the pre-trained DELG model.

S5. Additional Qualitative Results

Additional qualitative results on $\mathcal{R}$Oxford5k-Hard+1M, $\mathcal{R}$Paris6k-Hard+1M, and the GLDv2-retrieval-test are shown in Fig. S6, Fig. S7, and Fig. S8, respectively. The results show that the proposed re-ranking method performs re-ranking robustly, even if the global descriptor matching results in misjudgment in situations involving challenges such as viewpoint change, occlusion, and truncation.

References

[1] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Proc. European Conference on Computer Vision (ECCV)*, pages 726–743. Springer, 2020. 4, 5, 6, 7

[2] Cheng Chang, Guangwei Yu, Chundi Liu, and Maksims Volkovs. Explore-exploit graph traversal for image retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9423–9431, 2019. 5

[3] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007. 5

[4] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6

[5] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2077–2086, 2017. 5

[6] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 5

[7] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 7

[8] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. Solar: second-order loss and attention for image retrieval. In *Proc. European Conference on Computer Vision (ECCV)*, pages 253–270. Springer, 2020. 5

[9] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 3456–3465, 2017. 1, 4

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 6

[11] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(7):1655–1668, 2018. 5

[12] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 5107–5116, 2019. 5

[13] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017. 3

[14] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 6, 7

[15] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5109–5118, 2019. 5

[16] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2575–2584, 2020. 1

[17] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 11772–11781, 2021. 4

[18] Shuhei Yokoo, Kohei Ozaki, Edgar Simo-Serra, and Satoshi Iizuka. Two-stage discriminative re-ranking for large-scale landmark retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1012–1013, 2020. 1

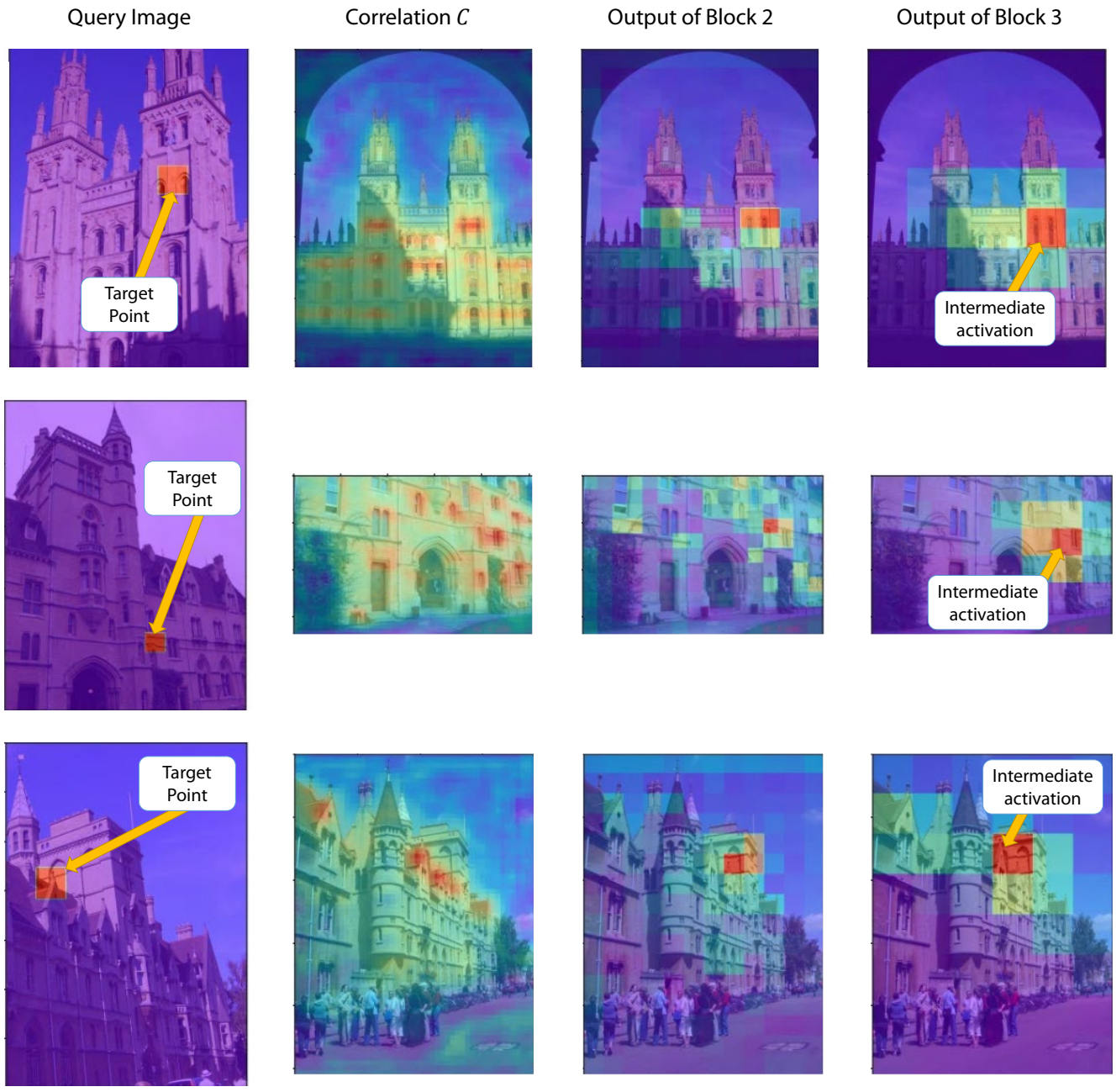


Figure S5. **Additional Intermediate Feature Visualization.**

Figure S6. **Additional qualitative results on $\mathcal{R}$Oxford5k-Hard+1M with R50-CVNet.** The upper line is the global descriptor matching result and the lower line is the re-ranking result. Correct/incorrect results are marked with green/red borders, respectively. The query used as an input is generated by cropping only the part bounded by a green square. Our purpose is to visualize the difference between global descriptor matching and re-ranking, so we skip the results of the ranks that are correct in both the global descriptor matching and re-ranking processes.

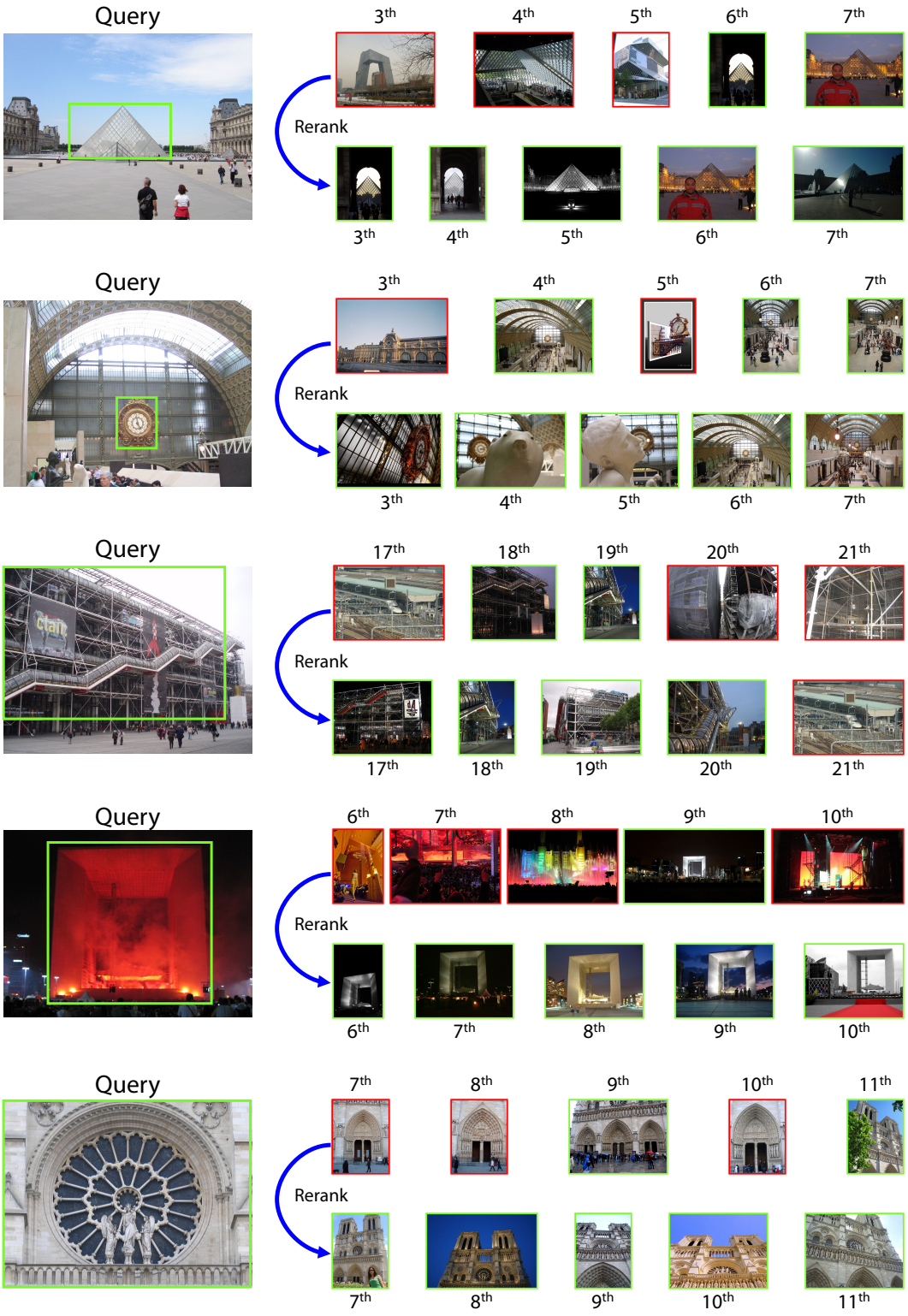


Figure S7. **Additional qualitative results on $\mathcal{R}$Paris6k-Hard+1M with R50-CVNet.** The upper line is the global descriptor matching result and the lower line is the re-ranking result. Correct/incorrect results are marked with green/red borders, respectively. The query used as an input is generated by cropping only the part bounded by a green square. Our purpose is to visualize the difference between global descriptor matching and re-ranking, so we skip the results of the ranks that are correct in both the global descriptor matching and re-ranking processes.

Figure S8. **Qualitative results on GLDv2-retrieval-test with R50-CVNet.** The upper line is the global descriptor matching result and the lower line is the re-ranking result. Correct/incorrect results are marked with green/red borders, respectively. The last two queries each have only one positive sample, so we skip the results after the correct answer comes out.