Abstract

Instance-level image retrieval is the task of searching in a large database for images that match an object in a query image. To address this task, systems usually rely on a retrieval step that uses global image descriptors, and a subsequent step that performs domain-specific refinements or reranking by leveraging operations such as geometric verification based on local features. In this work, we propose Reranking Transformers (RRTs) as a general model to incorporate both local and global features to rerank the matching images in a supervised fashion and thus replace the relatively expensive process of geometric verification. RRTs are lightweight and can be easily parallelized so that reranking a set of top matching results can be performed in a single forward-pass. We perform extensive experiments on the Revisited Oxford and Paris datasets, and the Google Landmarks v2 dataset, showing that RRTs outperform previous reranking approaches while using much fewer local descriptors. Moreover, we demonstrate that, unlike existing approaches, RRTs can be optimized jointly with the feature extractor, which can lead to feature representations tailored to downstream tasks and further accuracy improvements. The code and trained models are publicly available at https://github.com/uvavision/RerankingTransformer.

1. Introduction

Instance recognition is a challenging task that aims to visually recognize an object instance. This is distinct from category-level recognition that identifies only the object class. Instance recognition is important in e-commerce where it is often desired to find a specific product in a large image collection, or in place identification where the objective is to infer the identity of a public landmark. As the number of instances is much larger than the number of object categories, instance recognition is typically cast as image retrieval instead of classification, and usually involves both metric learning and local feature based reranking.

Over the last decade, instance recognition continues to be a major focus of research. Pioneering systems lever-
ASMK require large amounts of local descriptors to ensure retrieval performance.

In this work, we propose Reranking Transformers (RRTs), which learn to predict the similarity of an image pair directly. Our method is general and can be used as a drop-in replacement for other reranking approaches such as geometric verification. We conduct detailed experiments showing that as either a drop-in replacement or trained together with a global retrieval approach, the proposed method is the top-performing across the standard benchmarks for instance recognition. RRTs leverage the transformer architecture [59] which has led to significant improvements in natural language processing [16, 30] and vision-and-language tasks [28, 12, 33]. Most recently, it has also been used for purely vision tasks, notably for image recognition [18] and object detection [10]. To the best of our knowledge, our work is the first to adapt transformers for a visual task involving the analysis of image pairs in the context of reranking image search results.

Reranking Transformers are lightweight. Compared with typical feature extractors which have over 20 million parameters (e.g., 25 million in ResNet50 [24]), the proposed model only has 2.2 million parameters. It can also be easily parallelized such that re-ranking the top 100 neighbors requires a single pass. As shown in Fig. 1, our method directly predicts a similarity score for the matching images, instead of estimating a homography, which may be challenging under large viewpoint changes or infeasible for deformable objects. Our method requires much fewer descriptors but achieves superior performance, especially for challenging cases. In current state-of-the-art models, the feature extraction and matching are optimized separately, which may lead to suboptimal feature representations. In this work, we first perform experiments using pretrained feature extractors, then demonstrate the benefit of jointly optimizing the feature extractor and our model in a unified framework.

**Contributions.** (1) We propose Reranking Transformers (RRTs), a small and effective model which learns to predict the similarity of an image pair based on global and local descriptors; (2) Compared with existing methods, RRTs require fewer local descriptors and can be parallelized so that reranking the top neighbors only requires a single forward-pass; (3) We perform extensive experiments on Revisited Oxford/Paris [44], Google Landmarks v2 [61], and Stanford Online Products, and show that RRTs outperform prior reranking methods across a variety of settings.

2. Related Work

**Feature learning for instance recognition.** Hand-crafted local descriptors [34, 32] were widely used in earlier instance retrieval work [52, 36]. Recently, local features extracted from convolution neural networks (CNN) are shown to be more effective [37, 54, 51, 19, 38, 39]. Some of these works learn feature detection and representation jointly by non-local maximum suppression [19, 56], or attention [37, 54, 9]. The detected local descriptors are usually used for geometric verification [42] or ASMK [55]. Compared to local features, global descriptors provide a compact representation of an image for large-scale search. Current global descriptors are typically extracted from CNN models [3, 58, 43, 21] by spatial pooling [2, 26, 58, 43, 35], which may not be ideal for modeling region-wise relations across images. Recent systems either use global descriptors to reduce the solution space and then local descriptors to re-rank the nearest neighbors, or encode local descriptors using a large visual codebook, followed by image matching with an aggregated selective match kernel [55, 54, 56]. This work mainly follows the retrieve-and-rerank paradigm.

**Reranking for instance recognition/retrieval.** Geometric verification is the dominant image reranking approach and widely used in both traditional [42] and more recent works [51, 37, 9]. Inspired by text retrieval, query expansion techniques have also been introduced for image retrieval [14, 13, 57, 22]. These methods differ from geometric verification and our work as they rely on analysing the local nearest neighbor graph for each query during testing. Diffusion based approaches [17, 63, 25, 5, 4] aim to learn the structure of the data manifold by similarity propagation over the global affinity graph built on a query and all the gallery images, which is nontrivial to scale. Overall, the motivation of image reranking is to make better use of test-time knowledge to refine retrieval results. Our work shares the same vision with this line of research but focuses more on learning the similarity of an image pair directly.

**Transformers for visual tasks.** Transformers have become the dominant architecture for representing text [16, 30]. Recently, it has also been introduced to vision-and-language [28, 33] and pure vision tasks [40, 10, 27]. As the key ingredient of the transformer architecture, the self-attention mechanism has also been studied for visual recognition [6, 45, 64]. These works apply transformers for single image predictions while we leverage transformers to learn the visual relation of an image pair. Our work is also closely related to SuperGlue [50], which while not a Transformer, also relies on self-attention. SuperGlue aims to learn local correspondences between images with pixel-level supervision. Our work differs from SuperGlue in that it learns the similarity of an image pair with image-level supervision. We provide a best-effort comparison with this approach.

3. Methodology

3.1. Attention Modules in Transformers

First, we briefly review the key ingredients in the Transformer architecture: Single-Head Attention (SHA) and Multi-Head Attention (MHA).
Single-Head Attention (SHA): The input of a SHA layer comprises three sets of variables: the queries $Q := \{q_i \in \mathbb{R}^{d_q}\}_{i=1}^N$, the keys $K := \{k_j \in \mathbb{R}^{d_k}\}_{j=1}^M$ and the values $V := \{v_j \in \mathbb{R}^{d_v}\}_{j=1}^M$. Here, $d_q$, $d_k$, $d_v$ are the dimensions of the corresponding feature vectors, $N$ and $M$ are the sequence lengths. SHA produces a new feature sequence where each vector is a linear combination of $\{v_j\}$.

In doing this, $Q$, $K$, $V$ are first linearly projected as $Q = W^Q Q$, $K = W^K K$, $V = W^V V$, using parameter tensors: $W^Q \in \mathbb{R}^{d_q \times d_h}$, $W^K \in \mathbb{R}^{d_k \times d_h}$, $W^V \in \mathbb{R}^{d_v \times d_h}$, where $d_h$ is the new feature dimension. The output of a SHA layer is computed as: $SHA(Q, K, V) := \text{SOFTMAX}(\frac{K^T V}{\sqrt{d_h}}) V$.

Multi-Head Attention (MHA): Like SHA, MHA takes $Q$, $K$, $V$ as input and comprises multiple SHA modules: $\text{MHA}(Q, K, V) := [\text{HEAD}_1; \ldots; \text{HEAD}_h] W^O$, $\text{HEAD}_i := \text{SHA}_{Ax}(Q, K, V)$. Here $[;]$ denotes the concatenation operator, $h$ is the number of the SHA heads. $W^O \in \mathbb{R}^{(hd_h) \times d_v}$ is a linear projection with an output dimension of $d_v$.

3.2. Model

With the fundamental building blocks defined above, we introduce the detailed formulation of our model:

Image representations: An image $I$ is represented by a global descriptor of a dimension $d_g$: $x_g \in \mathbb{R}^{d_g}$ and a set of $L$ local descriptors: $x_l = \{x_{l,i} \in \mathbb{R}^{d_l}\}_{l=1}^L$, each of a dimension $d_l$. Both $x_g$ and $x_l$ are extracted from a CNN backbone (to be discussed in Sec. 4.2). Optionally, each $x_{l,i}$ is associated with a coordinate tuple $p_{l,i} = (u, v) \in \mathbb{R}^2$ and a scale factor $s_{l,i} \in \mathbb{R}$, indicating the pixel location and image scale where $x_{l,i}$ is extracted from. In this work, $s_{l,i}$ is an integer, indexing a set of pre-defined image scales.

Input: As a sequence transduction model [16, 30], Transformers take as input a list of “tokens” (e.g. Q,K,V in Sec. 3.1). In image retrieval, these “tokens” can be derived from the features of an image pair $(I, \bar{I})$. Following the BERT transformer encoder [16], we define the input as:

$$X(I, \bar{I}) := [\langle \text{CLS} \rangle; f_g(x_g); f_l(x_{l,1}); \ldots; f_l(x_{l,L}); \langle \text{SEP} \rangle; f_g(\bar{x}_g); f_l(\bar{x}_{l,1}); \ldots; f_l(\bar{x}_{l,L})],$$

where:

$$f_g(x_g) := x_g + \alpha;$$
$$f_l(x_{l,i}) := x_{l,i} + \varphi(p_{l,i}) + \psi(s_{l,i}) + \beta;$$
$$\bar{f}_g(\bar{x}_g) := \bar{x}_g + \alpha;$$
$$\bar{f}_l(\bar{x}_{l,i}) := \bar{x}_{l,i} + \varphi(\bar{p}_{l,i}) + \psi(\bar{s}_{l,i}) + \bar{\beta}.$$
4. Experiments

Next, we describe the datasets we use to evaluate our approach, and details about our implementation.

4.1. Datasets

We perform experiments on three datasets, Google Landmarks v2 (GLDv2) [61], Revisited Oxford/Paris [44], and Stanford Online Products (SOP) [53].

GLDv2: Google Landmarks v2 (GLDv2) [61] is a new benchmark for instance recognition that includes over five million images from 200k natural landmarks. As the Reranking Transformer has limited parameters (e.g. 2.2 million), we sample a small subset of the images from the “v2-clean” split of GLDv2 for training. We randomly sample 12,000 landmarks where each landmark has at least 10 images. For each landmark, we randomly sample at most 500 images. This results in 322,008 images, which is 20% of the “v2-clean” split and 8% of the original training set. The names of the sampled images are included in the supplementary material. For testing, we evaluate on the standard test set for the retrieval task, which contains 1,129 query images and 761,757 gallery images.

R Oxf and R Par: Revisited Oxford (R Oxf) and Paris (R Par) [44] are standard benchmarks for instance recognition, which have 4,993 and 6,322 gallery images respectively. They both have 70 query images, each with a bounding box depicting the location and span of the prominent landmark. An extra distractor set (R1M) with 1,001,001 images is included for large-scale experiments. We follow the standard evaluation protocol [44, 9] and crop the query image using the provided bounding box. We report mean Average Precision (mAP) on the Medium and Hard setups.

SOP: To investigate the benefit of jointly optimizing the feature representation and our Reranking Transformer, we perform experiments on a dataset of product images: Stanford Online Products (SOP) [53]. SOP is a commonly used benchmark for metric learning [62, 49, 47, 8, 48, 60, 7], which includes 120,053 images, 59,551 for training, 60,502 for testing. We follow the evaluation protocol for metric learning and report the R@K scores.

4.2. Implementation

Experiments on the pretrained descriptors: We first perform experiments with descriptors obtained from a pretrained feature extractor, DELG [9]. Our main experiments leverage ResNet50 [24] as the CNN backbone, but we also include experiments with ResNet101 in the supplementary material. DELG provides a unified framework for global/local feature extraction. The local descriptors, each with a dimension of 128, are extracted at 7 image scales ranging from 0.25 to 2.0. The global descriptor with a dimension of 2048 is extracted at 3 scales: \(\frac{1}{\sqrt{2}}, 1, \sqrt{2}\). We use an extra linear projection to reduce the global descriptor to a dimension of 128. In the original DELG model, the top 1000 local descriptors with the highest attention scores are selected for image reranking. We observe that RRT does not require this amount of descriptors, and the retrieval performance saturates at 500 local descriptors. Thus, in our experiments we choose the top 500 local descriptors and set \(L = 500, d_g = d_l = 128\). For images with fewer descriptors, we pad the feature sequence with empty vectors and use a binary attention mask, as in BERT [16], to indicate the padding locations. Both the global and local features are L2 normalized to unit norm. During training, the positive image is randomly sampled from the images sharing the same label as the query. The negative image is randomly sampled from the top 100 neighbors returned by the global retrieval, which have a different label from the query. DELG is pretrained on both Google Landmarks (GLD) v1 [37] and v2-clean [61]. Thus, we perform experiments on two sets of descriptors extracted from these two models. For the architecture, we use 4 SHA heads \((h = 4)\) and 6 transformer layers \((C = 6)\). \(d_g, d_k, d_v, d_e\) in SHA are set to 128, \(d_s\) is set to 32, \(d_e\) in MLP (Eq. 3) is set to 1024. The number of learnable parameters is 2,243,201, which is 9% of the amount in ResNet50. The model is trained with AdamW [31] for 15 epochs, using a learning rate of 0.0001 and a weight decay of 0.0004.

Experiments on SOP: We perform experiments on SOP [53] using a single image scale, following the protocol for metric learning [60]. During training, each image is randomly cropped to 224 × 224, followed by a random flip. During testing, each image is first resized to 256 × 256 then cropped at the center to 224 × 224. We use ResNet50 and extract features from the last convolutional layer, which leads to 49 \((7 × 7)\) local descriptors for each image. The global descriptor is obtained by spatially averaging the local responses. Both the global and local descriptors are linearly projected a dimension of 128. The RRT architecture and most of the training details remain the same as in the DELG experiments. Here we only describe the main differences. The global model is trained with a contrastive loss, as in [60]. Different from [60], we do not rely on a cross batch memory but simply use a batch size of 800. As all the local features are used, we do not incorporate the global descriptor term \((f_g(x_q), f_g(\bar{x}_q))\) in Eq. 1. We also drop the scale embedding \((\psi)\) as only one image scale is used. The global model is trained using SGD with Nesterov momentum for 100 epochs, using a learning rate of 0.001, a weight decay of 0.0004 and a momentum of 0.9. The learning rate drops by a factor of 10 after 60 and 80 epochs. We train an RRT model on top of the pretrained global model, either freezing or finetuning the CNN backbone. Both models are trained with AdamW [31] for 100 epochs, using a learning rate of 0.0001. The learning rate drops by a factor of 10 after 60 and 80 epochs. We implement RRTs in PyTorch [41].
Position embedding: For the experiments on DELG, where the keypoints are sparsely sampled, we observe no benefit in applying the position embedding and do not use the $\varphi$ term in Eq. 2. For the experiments on SOP, we find the position embedding is helpful, we posit it is because all the positions are used in this experiment.

Latency and memory: For each query, when using an NVIDIA P100 GPU, RRT reranks the top-100 retrieved images in a single forward-pass, which takes 0.36/0.013 seconds on average in the DELG [9]/SOP [53] experiments. In the DELG experiments, we use the same global descriptor but only half (500 out of 1000) of the local descriptors for each image. In other words, the memory footprint is approximately half of that in DELG [9]. We agree that this is still a high cost for large-scale systems. In the future, we’d like to explore techniques that can potentially reduce the memory footprint, e.g. quantization.

5. Results

Here, we demonstrate the effectiveness of the Reranking Transformers (RRTs) across different settings, benchmarks and use cases.

5.1. Comparison with Geometric Verification

We consider geometry verification (GV) as the main baseline. We compare GV and RRT using the same pre-trained DELG [9] descriptors. Following the protocol in [9], given a query, we use its global descriptor to retrieve a set of top-ranked images. The top-100 neighbors are reranked by GV and RRT. We present results on two sets of descriptors: DELG pretrained on GLD v1 [37] and v2-clean [61].

On $R_{Oxf}$ and $R_{Par}$, both GV and RRT outperform global-only retrieval, as shown in Table 1. RRT shows further advantages over GV, with much fewer local descriptors. On $R_{Oxf}$ (+$R_{1M}$), RRT performs on par with GV on the Medium setup and consistently better on the Hard setup. On $R_{Par}$ (+$R_{1M}$), RRT consistently outperforms GV. The largest performance gap is achieved on the Hard setup. RRT obtains 2.2 (3.7) absolute improvements over GV on $R_{Oxf}$ ($R_{Par}$), when using the “v1” descriptors. We posit that, while GV is effective for sufficiently similar images, it has difficulty handling challenging cases, e.g. large variations in viewpoint. To verify this, we reranked more images (e.g. top-200), resulting in a larger performance gap. RRT obtains 3.4 (8.4) absolute improvement over GV on $R_{Oxf}$ ($R_{Par}$), when using the “v1” descriptors.

We present results on the GLDv2 retrieval task [61] in Table 2. Following [9], we report the mAP@100 scores on the public and private test sets.

5.2. Comparison with Query Expansion

Query expansion (QE) [14, 13, 57] is another popular reranking technique for image retrieval. Different from
We tune these parameters on one of the most widely used query expansion methods: the nearest ones as in GV and RRT. We compare RRT with query expansion and RRT considerably improves over using \( \alpha \)QE only, showing that RRT and \( \alpha \)QE are complementary to each other.

GV and RRT, QE aggregates the query image and a number of top-ranked neighbors into a new query. This new query is used to rerank all the gallery images rather than the nearest ones as in GV and RRT. We compare RRT with one of the most widely used query expansion methods: \( \alpha \)-weighted query expansion (\( \alpha \)QE) proposed in [43]. We use the public implementation of \( \alpha \)QE released by [46]. \( \alpha \)QE has two hyper-parameters: (1) \( n \)QE, the number of top-ranked neighbors to aggregate; (2) \( \alpha \), the exponential weight. In [46], they are set as \((n \text{QE}, \alpha) = (10, 2.0)\). Our experiment shows that these values do not work out of the box for the DELG descriptors. We tune these parameters on \( \mathcal{R}\text{Oxf} \) over the ranges: \( n \text{QE} \in [2, 15], \alpha \in [0.1, 3.0] \), and eventually set them as \((n \text{QE}, \alpha) = (2, 0.3)\).

Table 3 shows the results on \( \mathcal{R}\text{Oxf} \) and \( \mathcal{R}\text{Par} \). When reranking the top 100 neighbors, the performance of RRT is superior to \( \alpha \)QE on five of the eight settings, except for \( \mathcal{R}\text{Par}+\text{Medium} \), \( \mathcal{R}\text{Par}+\text{R1M}+\text{Medium} \), \( \mathcal{R}\text{Par}+\text{R1M}+\text{Hard} \) (underlined numbers). We believe it is because \( \alpha \)QE reranks all the gallery images while RRT reranks only 100 neighbors and keeps the ranks of all the other images unchanged. By reranking more neighbors, e.g. 200, 400, we observe that the performance of RRT progressively improves and eventually surpasses \( \alpha \)QE by significant margins across all settings. On the Hard setup with the “v1” descriptors, the absolute gains of RRT over \( \alpha \)QE on \( \mathcal{R}\text{Oxf} \), \( \mathcal{R}\text{Oxf}+\text{R1M} \), \( \mathcal{R}\text{Par} \), \( \mathcal{R}\text{Par}+\text{R1M} \) are \((11.1, 9.0, 8.0, 3.5)\).

We also perform experiments on combining \( \alpha \)QE and RRT by reranking the top neighbors produced by \( \alpha \)QE. As shown in Table 3, combining \( \alpha \)QE and RRT considerably improves over using \( \alpha \)QE only, with improvements of \((10.4, 8.5, 6.7, 5.0)\) on the Hard setup of \( \mathcal{R}\text{Oxf}, \mathcal{R}\text{Oxf}+\text{R1M}, \mathcal{R}\text{Par}, \mathcal{R}\text{Par}+\text{R1M} \) for the “v1” descriptors. We consider query expansion and RRT are thus complementary.

### 5.3. Comparison with Aggregated Selective Match Kernel (ASMK)

Aggregated Selective Match Kernel (ASMK) [55] also leverages local descriptors for image retrieval. The key idea is to create a large visual codebook (i.e. filter banks) by clustering the local descriptors. This visual codebook is used to encode the query and gallery images into global descriptors. The clustering and encoding procedures are typically performed offline as they’re relatively time-consuming. Previously, ASMK was mainly considered as a global retrieval technique. In this paper, we treat ASMK as both a global retrieval baseline and a reranking baseline. We use the public implementation of ASMK released by [54]. Following the common practice proposed in [54], we train a codebook of 65,536 visual words on \( \mathcal{R}\text{Oxf} \) for

| Method          | # local Desc. version | Medium \( \mathcal{R}\text{Oxf} \) | Medium \( \mathcal{R}\text{Par} \) | Hard \( \mathcal{R}\text{Oxf} \) | Hard \( \mathcal{R}\text{Par} \) |
|-----------------|-----------------------|-----------------|-----------------|-----------------|-----------------|
| DELG global     | 0                     | 69.7            | 81.6            | 45.1            | 63.4            |
| ASMK global     | 1000                  | 71.2            | 80.8            | 47.1            | 61.6            |
| ASMK rerank     | 1000                  | 71.3            | 82.6            | 47.5            | 66.2            |
| RRT (ours)      | 500                   | 75.5            | 82.7            | 56.4            | 68.6            |
| DELG global     | 0                     | 73.6            | 85.7            | 51.0            | 71.5            |
| ASMK global     | 1000                  | 70.4            | 80.9            | 45.8            | 62.0            |
| ASMK rerank     | 1000                  | 73.1            | 86.3            | 49.3            | 71.9            |
| RRT (ours)      | 500                   | 78.1            | 86.7            | 60.2            | 75.1            |
b) ASMK rerank: using ASMK for image reranking, e.g. global retrieval, as in all the previous literature [55, 54, 56]; we conduct two experiments: a) ASMK global: using ASMK for retrieval experiments on Rset [53] are reported.

To further demonstrate the possibility of jointly optimizing the feature representations and RRTs, we perform experiments on the Stanford Online Products (SOP) dataset [53]. We study three models: (1) CO: A global retrieval model trained with a contrastive loss [60], using the metric learning protocol, i.e. the global descriptor has a dimension of 128 [48, 8, 49, 47, 60]; (2) CO + RRT (frozen): an RRT model trained on top of CO. The pretrained CO remains frozen and an extra linear layer is used to reduce the dimension of the local descriptors to 128; (3) CO + RRT (finetune): a model with the same architecture as CO + RRT (frozen) but the backbone is also finetuned. It is also initialized by CO + RRT (frozen). During testing, we perform global retrieval using the global descriptor from CO. The top-100 neighbors for each query are reranked by either CO + RRT (frozen) or CO + RRT (finetune). We present our results along with the results of the most recent metric learning approaches [48, 8, 60, 7] to provide an overview in the context of the state-of-the-art on SOP.

As shown in Table 6, the global CO model, which is trained with a contrastive loss using a relatively large batch size, performs surprisingly well. It achieves the same level of accuracy as well-established works on metric learning. This aligns with the recent research on self-supervised learning [11, 23] showing that contrastive loss is very effective for feature learning. CO + RRT (frozen) further improves the performance, demonstrating the effectiveness of reranking. Note that, as only the top-100 images are reranked, the R@100 and R@1k scores remain unchanged. CO + RRT (finetuned) achieves the best reranking performance, with an absolute improvement of 3.8 over the global-only retrieval on R@1. We believe it is because jointly optimizing the backbone and our model leads to better local features that are tailored to the reranking tasks.

### 5.4. Feature Learning & RRT: Joint Optimization

To further demonstrate the possibility of jointly optimizing the feature representations and RRTs, we perform experiments on the Stanford Online Products (SOP) dataset [53]. We study three models: (1) CO: A global retrieval model trained with a contrastive loss [60], using the metric learning protocol, i.e. the global descriptor has a dimension of 128 [48, 8, 49, 47, 60]; (2) CO + RRT (frozen): an RRT model trained on top of CO. The pretrained CO remains frozen and an extra linear layer is used to reduce the dimension of the local descriptors to 128; (3) CO + RRT (finetune): a model with the same architecture as CO + RRT (frozen) but the backbone is also finetuned. It is also initialized by CO + RRT (frozen). During testing, we perform global retrieval using the global descriptor from CO. The top-100 neighbors for each query are reranked by either CO + RRT (frozen) or CO + RRT (finetune). We present our results along with the results of the most recent metric learning approaches [48, 8, 60, 7] to provide an overview in the context of the state-of-the-art on SOP.

As shown in Table 6, the global CO model, which is trained with a contrastive loss using a relatively large batch size, performs surprisingly well. It achieves the same level of accuracy as well-established works on metric learning. This aligns with the recent research on self-supervised learning [11, 23] showing that contrastive loss is very effective for feature learning. CO + RRT (frozen) further improves the performance, demonstrating the effectiveness of reranking. Note that, as only the top-100 images are reranked, the R@100 and R@1k scores remain unchanged. CO + RRT (finetuned) achieves the best reranking performance, with an absolute improvement of 3.8 over the global-only retrieval on R@1. We believe it is because jointly optimizing the backbone and our model leads to better local features that are tailored to the reranking tasks.

### Table 5. Comparison to the state-of-the-art on Revisited Oxford/Paris [44]. The mAP scores on the Medium and Hard setups are reported.

| Method              | Training set | Net | # local desc. | Medium  | Hard  |
|---------------------|--------------|-----|---------------|---------|-------|
| R-MAC [21]          | Landmarks    | R101 | 0             | @1 60.9 | R1M 78.9 | +R1M 54.8 | @1 32.4 | R1M 59.4 | +R1M 28.0 |
| GeM [43]            | SIM-120k     | R101 | 0             | @1 64.7 | R1M 52.2 | +R1M 53.2 | @1 38.5 | R1M 56.3 | +R1M 24.7 |
| GeM-AP [46]         | SIM-120k     | R101 | 0             | @1 67.5 | R1M 52.5 | +R1M 42.8 | @1 42.8 | R1M 60.5 | +R1M 25.1 |
| DELG [9]            | GLDv1        | R50  | 0             | @1 69.7 | R1M 59.7 | +R1M 45.1 | @1 27.8 | R1M 63.4 | +R1M 34.1 |

### Table 6. Results on jointly optimizing the feature extractor and RRT. The R@K (K =1, 10, 100, 1000) scores on the SOP test set [53] are reported.

| Method | Desc. dim. | SOP | CO | CO + RRT (frozen) | CO + RRT (finetuned) |
|--------|-------------|-----|----|-------------------|----------------------|
| Margin [62, 48] | 128 | 76.1 | 95.1 | 98.3 | 98.3 |
| FastAP [8]      | 128 | 73.8 | 94.9 | 96.2 | 98.7 |
| XBM [60]        | 128 | 80.6 | 91.6 | 96.3 | 96.3 |
| CE [7]          | 2048 | 81.1 | 91.7 | 96.3 | 98.8 |
| CO              | 128 | 80.7 | 91.9 | 96.6 | 99.0 |
| CO + RRT (frozen) | 128 | 81.8 | 92.4 | 96.6 | 99.0 |
| CO + RRT (finetuned) | 128 | 84.5 | 93.2 | 96.6 | 99.0 |
reranked by RRT are presented. Correct/incorrect neighbors are marked with green/red borders.

Figure 3. Qualitative examples from Revisited Oxford/Paris [44]. For each query, the top-3 neighbors ranked by the global retrieval and the performance over the global-only retrieval. RRT gains

gains by both SuperGlue and our work significantly improve point numbers as our work. As shown in Table 7, reranking backbone, and uses different feature dimensions and key-ferent data distributions than the test sets (Revisited Ox-

ford/Paris). SuperGlue leverages SuperPoint [15] as theferent data, the CNN backbones, and the number of local features,etc. For context we provide as much information about eachmethod regarding these differences.

In Table 5, we compare the proposed method with thestate-of-the-art on the \( R \)Oxf (+\( R \)1M) and \( R \)Par (+\( R \)1M) benchmarks. We include the most recent instance recognition/retrieval models in three different groups: (A) Retrieval by global features only; (B) Retrieval by local feature aggregation; (C) Retrieval by combining global features with reranking. While our method performs favorably on most ofthe settings (except for \( R \)Oxf, \( R \)Oxf+\( R \)1M), these results include comparisons to methods that differ on the training data, the CNN backbones, and the number of local features, etc. For context we provide as much information about eachmethod regarding these differences.

In Fig. 3, we present qualitative examples on image re-


triever when using only global features and when usingour full reranking approach. While global-only retrievalcan return highly similar images in general, reranking byglobal/local descriptors captures a more fine-grained match-
ing between images, leading to better recognition accuracy.

Finally, Table 7 shows a comparison with Super-

Glue [50]. Similar as in geometry verification, the num-
er of inlier correspondences predicted by SuperGlue isused as the similarity score. As SuperGlue is not designedfor global retrieval, we use the pretrained DELG v1/v2-clean descriptors for global retrieval, so that SuperGlueand our method are evaluated on the same initial rankinglists. We use the SuperGlue pretrained on MegaDepth [29],which contains 130K images with dense annotations. Notethat both SuperGlue and our work are trained on datasets(MegaDepth vs a subset of GLDv2-clean) that have dif-

ferent data distributions than the test sets (Revisited Ox-

ford/Paris). SuperGlue leverages SuperPoint [15] as thebackbone, and uses different feature dimensions and key-

point numbers as our work. As shown in Table 7, rerank-
ing by both SuperGlue and our work significantly improve the performance over the global-only retrieval. RRT gains

5.5. Comparison with the State-of-the-Art

In Table 5, we compare the proposed method with thestate-of-the-art on the \( R \)Oxf (+\( R \)1M) and \( R \)Par (+\( R \)1M) benchmarks. We include the most recent instance recognition/retrieval models in three different groups: (A) Retrieval by global features only; (B) Retrieval by local feature aggregation; (C) Retrieval by combining global features with reranking. While our method performs favorably on most ofthe settings (except for \( R \)Oxf, \( R \)Oxf+\( R \)1M), these results include comparisons to methods that differ on the training data, the CNN backbones, and the number of local features, etc. For context we provide as much information about eachmethod regarding these differences.

In Fig. 3, we present qualitative examples on image re-


triever when using only global features and when usingour full reranking approach. While global-only retrievalcan return highly similar images in general, reranking byglobal/local descriptors captures a more fine-grained match-
ing between images, leading to better recognition accuracy.

Finally, Table 7 shows a comparison with Super-

Glue [50]. Similar as in geometry verification, the num-
er of inlier correspondences predicted by SuperGlue isused as the similarity score. As SuperGlue is not designedfor global retrieval, we use the pretrained DELG v1/v2-clean descriptors for global retrieval, so that SuperGlueand our method are evaluated on the same initial rankinglists. We use the SuperGlue pretrained on MegaDepth [29],which contains 130K images with dense annotations. Notethat both SuperGlue and our work are trained on datasets(MegaDepth vs a subset of GLDv2-clean) that have dif-

ferent data distributions than the test sets (Revisited Ox-

ford/Paris). SuperGlue leverages SuperPoint [15] as thebackbone, and uses different feature dimensions and key-

point numbers as our work. As shown in Table 7, rerank-
ing by both SuperGlue and our work significantly improve the performance over the global-only retrieval. RRT gains

| Method          | Desc. version | # local desc. | Desc. dim. | Medium ROxf | Hard ROxf | RPar  |
|-----------------|---------------|---------------|------------|-------------|-----------|-------|
| DELG global v1  | 500           | 128           | 75.5       | 62.7        | 56.4      | 68.6  |
| SuperGlue [50]  | SuperPoint [15] | 1000         | 256       | 74.4        | 55.2      | 64.6  |
| RRT (ours)      | 500           | 128           | 75.5       | 62.7        | 56.4      | 68.6  |
| SmartSearch     |               |               |            |             |           |       |

Table 7. Comparison to the pretrained SuperGlue model [50] on Revisited Oxford/Paris [44]. The SuperGlue model is pretrained on MegaDepth [29] with SuperPoint [15] as the backbone. The mAP scores on the Medium and Hard setups are reported.

larger improvements on most of the settings, especially for challenge cases. We provide more experiments on Super-

Glue in the supplemental material.

6. Conclusion

We introduce Reranking Transformers (RRTs) for in-

stance image retrieval. We show that RRTs outperform prior reranking approaches across a variety of settings. Com-

pared to geometric verification [42] and other local feature based methods [55], RRT’s use fewer descriptors and can be parallelized such that reranking requires a single forward pass. We also demonstrate that, unlike previous reranking approaches, RRTs can be optimized jointly with the feature extractor, leading to further gains.

Acknowledgements: This project was funded by a gift from eBay Inc to the University of Virginia. We also thank Robinson Piramuthu for his encouragement in starting this collaboration, Dmytro Mishkin for suggesting comparisons with SuperGlue and anonymous reviewers for their feedback. V.O. was also supported by a Facebook Research Award, a gift from Leidos Inc, and NSF Award IIS-2045773.
References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalisation. arXiv preprint arXiv:1607.06450, 2016.

[2] Artem Babenko and Victor S. Lempitsky. Aggregating deep convolutional features for image retrieval. In Int. Conf. Comput. Vis., 2015.

[3] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. In Eur. Conf. Comput. Vis., 2014.

[4] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. Regularized diffusion process on bidirectional context for object retrieval. IEEE Trans. Pattern Anal. Mach. Intell., 41(5):1213–1226, 2019.

[5] Song Bai, Peng Tang, Philip H.S. Torr, and Longin Jan Latecki. Re-ranking via metric fusion for object retrieval and person re-identification. In IEEE Conf. Comput. Vis. Pat- tern Recog., 2019.

[6] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In Int. Conf. Comput. Vis., 2019.

[7] Malik Boudiaf, J’erome Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. Metric learning: cross-entropy vs. pairwise losses. In Eur. Conf. Comput. Vis., 2020.

[8] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In Eur. Conf. Comput. Vis., 2020.

[9] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, Mihai Dusmanu, Ignacio Rocco, Tomás Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable CNN for joint detection and description of local features. In IEEE Conf. Comput. Vis. Pattern Recog., 2019.

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

[11] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. Int. J. Comput. Vis., 2017.

[12] Albert Gordo, Filip Radenovic, and Tamara Berg. In Eur. Conf. Comput. Vis., 2020.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In IEEE Conf. Comput. Vis. Pattern Recog., 2020.

[14] A simple framework for contrastive learning of visual representations. In Int. Conf. Mach. Learn., 2020.

[15] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In Eur. Conf. Comput. Vis., 2020.

[16] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Everest Hinton. A simple framework for contrastive learning of visual representations. In Int. Conf. Mach. Learn., 2020.

[17] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16478–16488, 2021.

[18] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In Eur. Conf. Comput. Vis. Worksh., 2015.

[19] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16478–16488, 2021.

[20] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Omer Levy. Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557, 2019.

[21] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In IEEE Conf. Comput. Vis. Pattern Recog., 2018.

[22] Yannin Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In Eur. Conf. Comput. Vis. Worksh., 2015.
[62] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In Int. Conf. Comput. Vis., Oct 2017. 4, 7

[63] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific rank fusion for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell., 37(4):803–815, 2015. 2

[64] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In IEEE Conf. Comput. Vis. Pattern Recog., 2020. 2