LEARNING SUPER-FEATURES FOR IMAGE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Methods that combine local and global features have recently shown excellent performance on multiple challenging deep image retrieval benchmarks, but their use of local features raises at least two issues. First, these local features simply boil down to the localized map activations of a neural network, and hence can be extremely redundant. Second, they are typically trained with a global loss that only acts on top of an aggregation of local features; by contrast, testing is based on local feature matching, which creates a discrepancy between training and testing. In this paper, we propose a novel architecture for deep image retrieval, based solely on mid-level features that we call *Super-features*. These Super-features are constructed by an iterative attention module and constitute an *ordered set* in which each element focuses on a localized yet discriminant image pattern. For training, only image labels suffice. A contrastive loss operates directly at the level of Super-features and focuses on Super-features that match across images. It is combined with a second complementary loss which encourages diversity. Experiments on common landmark retrieval benchmarks validate that Super-features substantially outperform state-of-the-art methods when using the same number of features, and only require a significantly smaller memory footprint to match their performance.

1 INTRODUCTION

Image retrieval is a task that models exemplar-based recognition, *i.e*. a class-agnostic, fine-grained understanding task which requires to retrieve all images matching a query image over an (often very large) image collection. It requires learning features that are discriminative enough for a highly detailed visual understanding but also robust enough to extreme viewpoint/pose or illumination changes. A popular image retrieval task is landmark retrieval, whose goal is to single out images of the exact same landmark out of millions of images, possibly containing a different landmark from the exact same fine-grained class (*e.g*. 'gothic-era churches with twin bell towers').

While early approaches relied on handcrafted local descriptors, recent methods use image-level (global) or local Convolutional Neural Networks (CNN) features, see Csurka & Humenberger (2018) for a review. The current state of the art performs matching or re-ranking using CNN-based local features (Noh et al., 2017; Cao et al., 2020; Tolias et al., 2020) and only learns with global (*i.e*. image-level) annotations and losses. This is done by aggregating all local features into a global representation on which the loss is applied, creating a discrepancy between training and inference.

It has been shown that attention maps from modules like the ones in (Vaswani et al., 2017) are able to capture intermediate scene-level information, which makes them fundamentally similar to mid-level features (Xiao et al., 2015; Chen et al., 2019). Unlike individual neurons in CNNs which are highly localized, attention maps may span the full input tensor and focus on more global or semantic patterns. Yet, the applicability of mid-level features for instance-level recognition and image retrieval is currently underwhelming; we argue that this is due to the following reasons: generic attention maps are not localized and may fire on multiple unrelated locations; at the same time, object-centric attentions such as the one presented in Locatello et al. (2020) produce too few attentional features and there is no mechanism to supervise them individually. In both cases, methods apply supervision at the global level, and the produced attentional features are simply not discriminative enough.

In this paper, we present a novel image representation and training framework based solely on attentional mid-level features we call *Super-features*. We introduce an iterative Local feature Integration Transformer (LIT), which tailors existing attention modules to the task of image retrieval. Compared

Figure 1: **Super-features attention maps** produced by our iterative attention module (LIT) for three images (left), with the first two that match, for five Super-features. Note that they consistently fire on some semantic patterns, *e.g*. circular shapes, windows, building tops (second to fourth columns).

to the slot attention (Locatello et al., 2020), for example, it is able to output an ordered and much larger set of features, as it is based on learned templates, and has a simplified recurrence mechanism. For learning, we devise a loss that is applied directly to Super-features, yet only requires image-level annotations. It pairs a contrastive loss on a set of matching Super-features across matching images, with a decorrelation loss on the attention maps of each image, to encourage Super-feature diversity. In the end, for each image our network extracts a fixed-size set of Super-features that are semantically ordered, *i.e*., each answering to different types of patterns; see Figure 1 for some examples.

At test time, we follow the protocol of the best performing recent retrieval methods, except that we aggregate and match Super-features instead of local features. Our experiments show that the proposed method significantly outperforms the state of the art on common benchmarks like $\mathcal{R}$Oxford and $\mathcal{R}$Paris (Radenović et al., 2018a), while requiring less memory. We further show that performance gains persist in the larger scale, *i.e*. after adding 1M distractor images, while exhaustive ablations suggest that Super-features are less redundant and more discriminative than local features.

Contributions. Our contribution is threefold: (a) an image representation based on mid-level features (*Super-features*) and an iterative module to extract them; (b) a framework to learn such representations, based on a loss applied directly on Super-features yet only requiring image-level labels; (c) extensive evaluations that show *significant* performance gains over the state of the art for landmark image retrieval. We call our method *Feature Integration-based Retrieval* or FIRe for short.

2 Background: Learning local features with a global loss

Let function $f : \mathcal{I} \rightarrow \mathbb{R}^{W \times H \times D}$ denote a convolutional neural network (CNN) backbone that encodes an input image $\boldsymbol{x} \in \mathcal{I}$ into a $(W \times H \times D)$-sized tensor of D-dimensional local activations over a $(W \times H)$ spatial grid. After flattening the spatial dimensions, the output of f can also be seen as set of $L = W \cdot H$ feature vectors denoted by $\mathcal{U} = \{\boldsymbol{u}_l \in \mathbb{R}^D : l \in 1 .. L\}$; note that the size of this set varies with the resolution of the input image. These local features are then typically whitened and their dimension reduced, a process that we represent by function $o(\cdot)$ in this paper. Global representations, *i.e*. image-level feature vectors, are commonly produced by averaging all local features, *e.g*. via global average pooling (Babenko & Lempitsky, 2015).

A global contrastive loss for training. Tolias et al. (2020) argue that optimizing global representations is a good surrogate for learning local features to be used together with efficient match kernels for image retrieval. When building their global representation $g(\mathcal{U})$, they weight the contribution of

each local feature to the aggregated vector using its l_2 norm:

$$g(\mathcal{U}) = \frac{\hat{g}(\mathcal{U})}{\|\hat{g}(\mathcal{U})\|_2}, \qquad \hat{g}(\mathcal{U}) = \sum_{l=1}^{L} w_l \cdot o(\boldsymbol{u}_l), \quad w_l = \|o(\boldsymbol{u}_l)\|_2 \,, \tag{1}$$

where $\|\cdot\|_2$ denotes the l_2 norm. Given a database where each image pair is annotated as matching with each other or not, they propose minimizing a contrastive loss over tuples of global representations. Intuitively, this loss encourages the global representations of matching images to be similar and those of non-matching images to be dissimilar. Let tuple $(\mathcal{U}, \mathcal{U}^+, \mathcal{V}_1^-, \ldots, \mathcal{V}_n^-)$ represent the sets of local features of images $(\boldsymbol{x}, \boldsymbol{x}^+, \boldsymbol{y}_1^-, \ldots, \boldsymbol{y}_n^-)$, where $\boldsymbol{x}, \boldsymbol{x}^+$ are matching images (*i.e.* a positive pair) and $\boldsymbol{y}_1^-, \ldots, \boldsymbol{y}_n^-$ are not matching image $\boldsymbol{x}$ (*i.e.* are negatives). Let $[\cdot]^+$ denote the positive part and μ a margin hyper-parameter, they define a contrastive loss over global representations as:

$$\mathcal{L}_{global} = \|g(\mathcal{U}) - g(\mathcal{U}^+)\|_2^2 + \sum_{j=1}^{n} \left[\mu - \|g(\mathcal{U}) - g(\mathcal{V}_j^-)\|_2^2\right]^+. \tag{2}$$

In *HOW*, Tolias et al. (2020) employ the global contrastive loss of Eq.(2) to learn a model whose local features are then used with match kernels such as ASMK (Tolias et al., 2013) for image retrieval. ASMK is a matching process defined over selective matching kernels of local features; it is a much stricter and more precise matching function than comparing global representations, and is crucial for achieving good performance. By learning solely using a loss defined over global representations and directly using ASMK over local features $\mathcal{U}$, HOW achieves excellent image retrieval performance.

3 Learning with Super-features

Although highly successful, methods that learn with a global loss and then use local features $\mathcal{U}$ for matching have a number of disadvantages. First, using local network activations as local features leads to high redundancy, as they exhaustively cover highly overlapping patches of the input image. Second, using ASMK on features from a model trained with a global loss introduces a mismatch between training and testing: the local features used for ASMK are only trained *implicitly* via a loss over aggregated features, but are expected to individually match in the matching kernel.

To obtain less redundant feature sets, we propose to learn mid-level features we call *Super-features* using a loss function that operates directly on those Super-features, and also use the latter during retrieval; the train/testing discrepancy of the pipeline presented in Section 2 is thus eliminated. In this section, we first introduce the Local feature Integration Transformer (*LIT*), an iterative attention module which produces an ordered set of Super-features (Section 3.1). We then present a framework for effectively learning such features (Section 3.2) that consists of two complementary losses: A contrastive loss that matches individual Super-features across positive image pairs, and a decorrelation loss on the Super-feature attention maps that encourages them to be as diverse as possible. An overview of the full pipeline is depicted in Figure 2. We refer to the proposed image retrieval framework based on Super-features as *Feature Integration-based Retrieval* or FIRe for short, an homage to the seminal feature integration theory of Treisman & Gelade (1980).

3.1 Local feature Integration Transformer (LIT)

Inspired by the recent success of attention mechanisms for encoding semantics from global context in sequences (Vaswani et al., 2017) or images (Caron et al., 2021), we rely on attention to design our Local feature Integration Transformer (*LIT*), a module that outputs an ordered set of *Super-features*.

Let LIT be represented by function $\Phi(\mathcal{U}) : \mathbb{R}^{L \times D} \rightarrow \mathbb{R}^{N \times d}$ that takes as input the set of local features $\mathcal{U}$ and outputs N Super-features. We define LIT as an *iterative* module:

$$\Phi(\mathcal{U}) = \mathcal{Q}^T, \qquad \mathcal{Q}^t = \phi(\mathcal{U}; \mathcal{Q}^{t-1}), \tag{3}$$

where ϕ denotes the core function of the module applied T times, and $\mathcal{Q}^0 \in \mathbb{R}^{N \times d}$ denotes a set of learnable *templates*, *i.e.* a matrix of learnable parameters. Super-features are progressively formed by iterative refinement of the templates, conditioned on the local features from the CNN.

The architecture of the core function ϕ is inspired by the Transformer architecture (Vaswani et al., 2017) and is composed of a dot-product attention function ψ, followed by a multi-layer perceptron

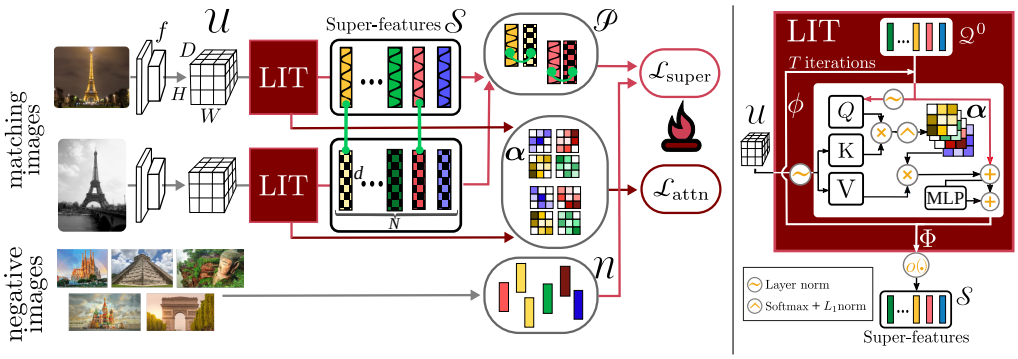


Figure 2: **An overview of FIRe.** Given a pair of matching images encoded by a CNN encoder, the iterative attention module LIT (Section 3.1) outputs an ordered set of Super-features. A filtering process keeps only reliable Super-feature pairs across matching images (Section 3.2§1), which are fed into a Super-feature-level contrastive loss (Section 3.2§2), while a decorrelation loss reduces the spatial redundancy of the Super-features attention masks for each image (Section 3.2§3).

(MLP). The dot-product attention function ψ receives three inputs, the *key*, the *value* and the *query*[1] which are first passed through layer normalization and then fed to linear projection functions K, V and Q that project them to dimensions d_k, d_v and d_q, respectively.

The key and value inputs are set as the local features $\boldsymbol{u}_l \in \mathcal{U}$ across all iterations. The query input is template $\mathcal{Q}^t = \{\boldsymbol{q}_j^t \in \mathbb{R}^d, j = 1 .. N\}$. It is initialized as the learnable templates $\mathcal{Q}^0$ for iteration 0, and is set as the previous output of function ϕ for the following iterations. After projecting with the corresponding linear projection functions, the key and the query are multiplied to construct a set of N *attention maps* over the local features, given by:[2]

$$\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1; \dots; \boldsymbol{\alpha}_L] \in \mathbb{R}^{L \times N}, \quad \boldsymbol{\alpha}_l = \frac{\hat{\boldsymbol{\alpha}}_l}{\sum_{i=1}^{L} \hat{\boldsymbol{\alpha}}_i}, \quad \hat{\boldsymbol{\alpha}}_l = \frac{e^{M_l}}{\sum_{j=1}^{N} e^{M_{lj}}}, \quad M_{lj} = \frac{K(\boldsymbol{u}_l) \cdot Q(\boldsymbol{q}_j^t)}{\sqrt{d}}, \tag{4}$$

where $[\boldsymbol{\alpha}_l]$ denotes stacking $\boldsymbol{\alpha}_l$ into a matrix. The dot product between keys and templates can be interpreted as a tensor of compatibility scores between local features and templates. These scores are normalized across templates via a softmax function, and are further turned into attention maps by l_1 normalization across all L spatial locations. The input value is first projected with V, re-weighted with the attention maps and then residually fed to a $\texttt{MLP}$[3] to produce the output of function ϕ:

$$\phi(\mathcal{U}; \mathcal{Q}) = \texttt{MLP}(\psi(\mathcal{U}; \mathcal{Q})) + \psi(\mathcal{U}; \mathcal{Q}), \qquad \psi(\mathcal{U}; \mathcal{Q}) = \boldsymbol{\alpha} \cdot V(\mathcal{U}) + \mathcal{Q}. \tag{5}$$

Following standard image retrieval practice, we further whiten and l_2-normalize the output of $\Phi(\mathcal{U})$ to get the final set of Super-features. Specifically, let $\Phi(\mathcal{U}) = [\hat{\boldsymbol{s}}_1; \dots; \hat{\boldsymbol{s}}_N] \in \mathbb{R}^{N \times d}$ be the raw output of our iterative attention module, we define the ordered set of Super-features as:

$$\mathcal{S} = \left\{ \boldsymbol{s}_i : \boldsymbol{s}_i = \frac{o(\hat{\boldsymbol{s}}_i)}{\|o(\hat{\boldsymbol{s}}_i)\|_2}, \quad i = 1, .., N \right\}, \tag{6}$$

where, as in Section 2, $o(\cdot)$ denotes dimensionality reduction and whitening. Figure 2 (right) illustrates the architecture of LIT. Note that all learnable parameters of ϕ are shared across all iterations.

What do Super-features attend to? Each Super-feature in $\mathcal{S}$ is a function of all local features in $\mathcal{U}$, invariant to permutation of its elements, and can thus attend arbitrary regions of the image. To visualize the patterns captured by Super-features, Figure 1 shows the attention maps of the same five Super-features for three images, including two of the same landmark (first two rows). The type of patterns depends on the learned initialization templates $\mathcal{Q}^0$; this explains why the Super-features form an ordered set and allows to directly compare Super-features with the same ID. We observe that the attention maps to be similar across images of the same landmark and to contain some mid-level patterns (such as a half-circle on the second column, or windows on the third one).

[1] The term 'query' has a precise meaning for retrieval; yet, for this subsection only, we overload the term to refer to one of the inputs of the dot-product attention, consistently with the terminology from seminal works on attention like Vaswani et al. (2017).

[2] The attention maps presented in Eq.(4) are technically taken at iteration t, but we omit iteration superscripts for clarity. For the rest of the paper and visualizations, we use *attention maps* to refer to the attention maps of Eq.(4) after the final (T-th) iteration of the iterative module.

[3] The $\texttt{MLP}$ function consists of a layer-norm, a fully-connected layer with half the dimensions of the features, a ReLU activation and a fully-connected layer that projects features back to their initial dimension.

3.2 LEARNING WITH SUPER-FEATURES

We jointly fine-tune the parameters of the CNN and of the LIT module using contrastive learning. However, unlike recent approaches like HOW (Tolias et al., 2020) or DELG (Cao et al., 2020) that use a global contrastive loss similar to the one presented in Eq. (2) for learning, we introduce a contrastive loss that operates directly on Super-features, *the representations we use at test time*, and yet only requires image-level labels. Given a positive pair (*i.e.* matching images of the same landmark) and a set of negatives, we select promising Super-feature pairs by exploiting their ordered nature and without requiring any extra supervision signal. We then minimize their pairwise distance, while simultaneously reducing the spatial redundancy of Super-features within an image.

Selecting matching Super-features. Since we are only provided with pairs of matching images, *i.e.* image-level labels, defining correspondences at the Super-feature level is not trivial. Instead of approximating sophisticated metrics as in (Liu et al., 2020; Mialon et al., 2021)–see Section 5 for a discussion–, we leverage the fact that Super-features are ordered and we select promising matches and filter out erroneous ones only relying on simple, nearest neighbor-based constraints.

For any $\boldsymbol{s} \in \mathcal{S}$, let $i(\boldsymbol{s})$ be the function that returns the position/order or *Super-feature ID*, *i.e.* $i(\boldsymbol{s}_i) = i, \ \forall \boldsymbol{s}_i \in \mathcal{S}$. Further let function $n(\boldsymbol{s}, \mathcal{S}) = \arg\min_{\boldsymbol{s}_i \in \mathcal{S}} \|\boldsymbol{s} - \boldsymbol{s}_i\|_2$ be the function that returns the nearest neighbor of $\boldsymbol{s}$ from set $\mathcal{S}$. Now, given a positive pair of images $\boldsymbol{x}, \boldsymbol{x}^+$, let $\boldsymbol{s} \in \mathcal{S}, \boldsymbol{s}' \in \mathcal{S}'$ be two Super-features from their Super-feature sets $\mathcal{S}, \mathcal{S}'$, respectively. In order for Super-feature pair $(\boldsymbol{s}, \boldsymbol{s}')$ to be *eligible*, all of the following criteria must be met: a) $\boldsymbol{s}, \boldsymbol{s}'$ have to be reciprocal nearest neighbors, b) they need to pass Lowe's first-to-second neighbor ratio test (Lowe, 2004) and c) they need to have the same Super-feature ID. Let $\mathcal{P}$ be the set of eligible pairs and τ the hyper-parameter that controls the ratio test; the conditions above can formally be written as:

$$(\boldsymbol{s}, \boldsymbol{s}') \in \mathcal{P} \iff \begin{cases} s = n(\boldsymbol{s}', \mathcal{S}) \\ s' = n(\boldsymbol{s}, \mathcal{S}') \end{cases} \quad \text{and} \quad \begin{cases} i(\boldsymbol{s}) = i(\boldsymbol{s}') \\ \|\boldsymbol{s} - \boldsymbol{s}'\|_2 \ / \ \|\boldsymbol{s} - n(\boldsymbol{s}, \mathcal{S}' \backslash \{\boldsymbol{s}'\})\|_2 \geqslant \tau \end{cases} . \tag{7}$$

We set $\tau = 0.9$. Our ablations (Section 4.1) show that all four criteria are important. Note that the pair selection step is non-differentiable and no gradients are computed for Super-features not in $\mathcal{P}$. We further discuss this in Appendix C.4 and empirically show that all Super-features are trained.

A contrastive loss on Super-features. Once the set $\mathcal{P}$ of all eligible Super-feature pairs has been constructed, we define a contrastive margin loss on these matches. Let pair $p = (\boldsymbol{s}, \boldsymbol{s}^+) \in \mathcal{P}$ be a pair of Super-features, selected from a pair of matching images $\boldsymbol{x}, \boldsymbol{x}^+$, and let $\mathcal{N}(j) = \{\boldsymbol{n}_j^k : k = 1 \,..\, n\}$ for $j = 1 \,..\, N$ be the set of Super-features with Super-feature ID j extracted from negative images $(\boldsymbol{y}_1^-, \ldots, \boldsymbol{y}_n^-)$. The contrastive Super-feature loss can be written as:

$$\mathcal{L}_{super} = \sum_{(\boldsymbol{s}, \boldsymbol{s}^+) \in \mathcal{P}} \Big[\|\boldsymbol{s} - \boldsymbol{s}^+\|_2^2 + \sum_{\boldsymbol{n} \in \mathcal{N}(i(\boldsymbol{s}))} [\mu' - \|\boldsymbol{s} - \boldsymbol{n}\|_2^2]^+ \Big], \tag{8}$$

where μ' is a margin hyper-parameter and the negatives for each $\boldsymbol{s}$ are the Super-features from all n negative images of the training tuple with Super-feature ID equal to $i(\boldsymbol{s})$.

Reducing the spatial correlation between attention maps. To obtain Super-features that are as complementary as possible, we encourage them to attend to different local features, *i.e.* different locations of the image. To this end, we minimize the cosine similarly between the attention maps of all Super-features of every image. Specifically, let $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_j], \boldsymbol{\alpha}_j \in \mathbb{R}^L, j = 1 \,..\, N$ be the attention maps after the last iteration of LIT, see Eq.(4). The *attention decorrelation loss* is given by:

$$\mathcal{L}_{attn}(\boldsymbol{x}) = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{\boldsymbol{\alpha}_i^\intercal \cdot \boldsymbol{\alpha}_j}{\|\boldsymbol{\alpha}_i\|_2 \|\boldsymbol{\alpha}_j\|_2}, \quad i, j \in \{1, .., N\}. \tag{9}$$

In other words, this loss minimizes the off-diagonal elements of the $N \times N$ self-correlation matrix $\boldsymbol{\alpha}^\intercal \boldsymbol{\alpha}$. We ablate the gains from this loss and all others presented in this section in Section 4.1.

Image retrieval with Super-features. Our full pipeline, FIRe, is composed of a model that outputs Super-features, trained with the contrastive Super-feature loss of Eq.(8) and the attention decorrelation loss of Eq.(9). As our ablations show (see Section 4), the combination of these two losses is required for achieving state-of-the-art performance, while adding a third loss on the aggregated global features as in Eq.(2) does not bring any gain.

reci-proc.	ratio test	same ID	SfM-120k val	$\mathcal{R}$Oxford med	$\mathcal{R}$Oxford hard	$\mathcal{R}$Paris med	$\mathcal{R}$Paris hard
			68.3	64.3	39.8	74.1	52.4
✓			79.6	69.2	44.3	79.2	60.9
✓	✓		80.8	70.7	45.1	80.3	61.9
		✓	75.9	63.8	35.1	77.3	56.5
✓	✓	✓	**89.7**	**81.8**	**61.2**	**85.3**	**70.0**

Table 1: **Ablation on matching constraints**: impact of removing constraints on reciprocity, Lowe's ratio test and the Super-feature ID.

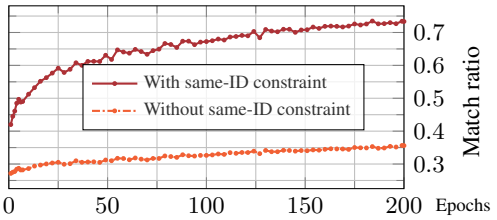


Figure 3: **Evolution of the matching quality** measured as the ratio of matches coming from the positive pair over all pairs with the query in each training tuple.

4 Experiments

This section validates our proposed FIRe approach on standard landmark retrieval tasks. We use the SfM-120k dataset (Radenović et al., 2018b) following the 551/162 3D model train/val split from Tolias et al. (2020). For testing, we evaluate instance-level search on the $\mathcal{R}$Oxford (Philbin et al., 2007) and the $\mathcal{R}$Paris (Philbin et al., 2008) datasets in their revisited version (Radenović et al., 2018a), with and without the 1 million distractor set called $\mathcal{R}$1M. They both contain 70 queries, with 4993 and 6322 images respectively. We report mAP on the Medium (med) and Hard (hard) setups, or the average over these two setups (avg).

Image search with Super-features. At test time, we follow the exact procedure described in Tolias et al. (2020) and extract Super-features from each image at 7 resolutions/scales $\{2.0, 1.414, 1.0, 0.707, 0.5, 0.353, 0.25\}$. We then keep the top Super-features (the top 1000 unless otherwise stated) according to their L2 norm and use the binary version of ASMK with a codebook of 65536 clusters. Note that one can measure the memory footprint of a database image $\boldsymbol{x}$ via the number of non-empty ASMK clusters, *i.e.* clusters with at least one assignment, denoted as $|\mathcal{C}(\boldsymbol{x})|$.

Implementation details. We build our codebase on HOW (Tolias et al., 2020)[4] and sample tuples composed of one query image, one positive image and 5 hard negatives. Each epoch is composed of 400 batches of 5 tuples each, while hard negatives are updated at each epoch using the global features of Eq.(1). We train our model for 200 epochs on SfM-120k using an initial learning rate of 3.10^{-5} and random flipping as data augmentation. We multiply the learning rate by a factor of 0.99 at each epoch, and use an Adam optimizer with a weight decay of 10^{-4}. We use a ResNet50 (He et al., 2016) without the last convolutional block as backbone (R50$^-$). For LIT, we use $D = d_k = d_q = d_v = d = 1024$. Following HOW, we reduce the dimensionality of features to 128 and initialize $o(\cdot)$ using PCA and whitening before training and keep it frozen. We pretrain the LIT module with the backbone on ImageNet-1K for image classification, see details in Appendix C.6. We use $\mu' = 1.1$ in Eq.(8) and weight $\mathcal{L}_{super}$ and $\mathcal{L}_{attn}$ with 0.02 and 0.1, respectively.

4.1 Analysis and ablations

In this section, we analyze the proposed FIRe framework and perform exhaustive ablations on the training losses and the matching constraints. The impact of the number of iterations (T) and templates (N) in LIT are studied in Appendix B.1. For the rest of the paper, we set $T = 6$ and $N = 256$.

Matching constraints ablation. Table 1 reports the impact of the different matching constraints: nearest neighbor, reciprocity, Lowe ratio test and constraint on Super-features ID. Adding reciprocity and Lowe's ratio test significantly improves the performance, which indicates that it reduces the number of incorrect matches. Keeping all pairs with the same ID yields lower performance, so the ID constraint alone is clearly not enough. One possible explanation is that two images from the same landmark may differ significantly, *e.g.* when some parts are visible in only one of the two images. In that case, the other constraints allow features attending non-overlapping regions of a positive image pair to be excluded from $\mathcal{P}$. Finally, combining the selective ID constraint with the others leads to the best performance.

In order to better understand the impact of the Super-feature ID constraint, a constraint only applicable for Super-features due to their ordered nature, we measure the quality of selected matches during

[4] https://github.com/gtolias/how

$\mathcal{L}_{global}$	$\mathcal{L}_{attn}$	$\mathcal{L}_{super}$	SfM-120k val	$\mathcal{R}$Oxford med	$\mathcal{R}$Oxford hard	$\mathcal{R}$Paris med	$\mathcal{R}$Paris hard
✓			79.0	64.3	38.0	75.4	51.7
✓	✓		87.7	75.8	51.2	79.0	57.0
✓	✓	✓	88.4	79.0	57.2	83.0	65.6
		✓	61.7	59.3	32.8	69.9	47.3
	✓	✓	**89.7**	**81.9**	**61.5**	**85.3**	**70.1**

Table 2: **Ablation on loss components**: Impact of removing $\mathcal{L}_{\text{attn}}$, using either a global loss $\mathcal{L}_{\text{global}}$ or a loss directly on Super-features $\mathcal{L}_{\text{super}}$ or a combination of both.

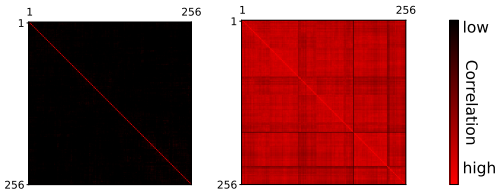


Figure 4: **Impact of $\mathcal{L}_{\text{attn}}$ on the correlation matrix between attention maps** (the darker, the lower is the correlation) at the last iteration of LIT when training with (left) and without (right). This is averaged over the 70 queries of $\mathcal{R}$Oxford.

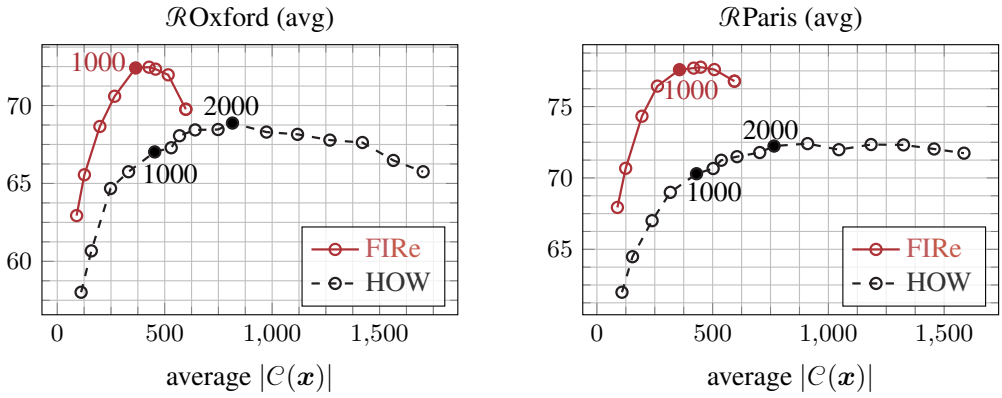


Figure 5: **Performance versus memory when varying the number of selected features** for HOW and FIRe. The x axis represents the average number of clusters used per image in ASMK – proportional to memory – when varying the number of selected features in $\{200, \ldots, 2000, 2500, \ldots, 5000\}$. FIRe has at most 1,792 extracted features (256 per scale).

training with and without it. Since there is no ground-truth for such localized matches on landmark retrieval datasets, we measure instead the ratio of matches coming from the positive pair, over all matches (from the positive and all negatives). Ideally, a minimal number of matches should come from the negatives, hence this ratio should be close to 1. In Figure 3 we plot this match ratio for all epochs; we observe that it is significantly higher when using the Super-features ID constraint.

Training losses ablation. We study the impact of the different training losses in Table 2. We start from a global loss similar to HOW (first row). We then add the decorrelation loss (second row) and observe a clear gain. It can be explained by the fact that without this loss, Super-features tend to be redundant (see the correlation matrices in Figure 4). Adding the loss operating directly on the Super-features further improves performance (third row). Next, we remove the global loss and keep only the loss on Super-features alone (fourth row) or with the decorrelation loss (last row). The latter performs best (fourth vs last row). Figure 4 displays the correlation matrix of Super-features attention maps with and without $\mathcal{L}_{\text{attn}}$. Without it, we observe that most Super-features have correlated attentions. In contrast, training with $\mathcal{L}_{\text{attn}}$ leads to uncorrelated attention maps. This is illustrated in Figure 1 which shows several attention maps focusing on different areas.

Varying the number of features at test time. Figure 5 compares HOW (Tolias et al., 2020) with our approach, as we vary the number of local features / Super-features. The x axis shows the average number of clusters used in ASMK for the database images, *i.e.*, which is proportional to the average memory footprint of an image. We observe that our approach does not only significantly improve accuracy compared to HOW, but also requires overall less memory. Notably, FIRe matches the best performance of HOW with a memory footprint reduced by a factor of 4. For both methods, performance goes down once feature selection no longer discards background features. The gain in terms of performance is even larger when considering a single scale at test time, as shown in Appendix C.1.

Statistics on the number of selected features per scale. The left plot of Figure 6 shows the percentage of the 1000 selected features that comes from each scale, for HOW and FIRe. For HOW

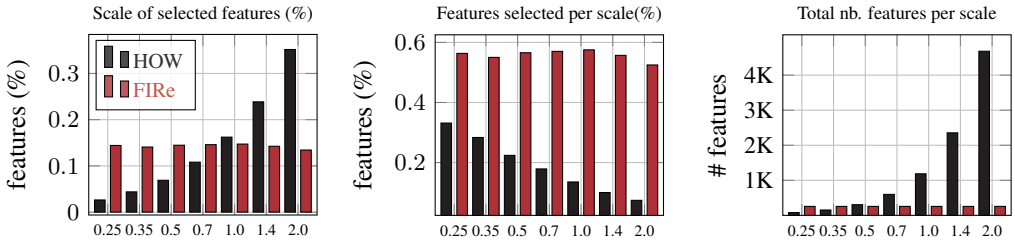


Figure 6: **Statistics on the feature selected across scales** for HOW and FIRe, averaged over the 70 queries from $\mathcal{R}$Oxford. *Left:* Among the 1000 selected features, we show the percentage coming from each scale. *Middle:* For each scale, we show the ratio of features that are selected. *Right:* Total number of features per scale; we extract $N = 256$ Super-features regardless of scale.

most selected features come from the higher resolution inputs; by contrast, selected Super-features are almost uniformly distributed across scales. Interestingly, Figure 6 (middle) shows that HOW keeps a higher percentage of the features coming from coarser scales. Yet, the final feature selection for HOW is still dominated by features from higher scales, due to the fact that the number of local features significantly increases with the input resolution (see Figure 6 right).

4.2 Comparison to the state of the art

We compare our method to the state of the art in Table 3. All reported methods are trained on SfM-120k for a fair comparison[5]. First, we observe that methods based on global descriptors (Gordo et al., 2016; Revaud et al., 2019a) compared with a L2 distance tend to be less robust than methods based on local features (Noh et al., 2017; Tolias et al., 2020). DELG (Cao et al., 2020) shows better performance, owing to a re-ranking step based on local features, at the cost of re-extracting local features for the top-ranked images of a given query, given that local features would take too much memory to store. Our FIRe method outperforms DELF (Noh et al., 2017) as well as HOW (Tolias et al., 2020) by a significant margin when extracting 1000 features per image. Importantly, our approach also requires less memory, as it uses fewer ASMK clusters per image, as shown in Figure 5: for the whole $\mathcal{R}$1M set, HOW uses 469 clusters per image while FIRe uses only 383, on average.

5 Related Work

Image retrieval. The first approaches for image retrieval were based on handcrafted local descriptors and bag-of-words representations borrowed from text retrieval (Sivic & Zisserman, 2003; Csurka et al., 2004), or other aggregation techniques like Fisher Vectors (Perronnin et al., 2010), VLAD (Jégou et al., 2010) or ASMK (Tolias et al., 2013). First deep learning techniques were extracting a global vector per image either directly or by aggregating local activations, and have shown to highly outperform handcrafted local features, see Csurka & Humenberger (2018) for a review. Methods that perform matching or re-ranking using CNN-based local features are currently the state of the art in the area (Noh et al., 2017; Teichmann et al., 2019; Cao et al., 2020; Tolias et al., 2020). They are able to learn with global (*i.e.* image-level) annotations. Most of them use simple variants of attention mechanisms (Noh et al., 2017; Ng et al., 2020) or simply the feature norm (Tolias et al., 2020) to weight local activations.

Low-level features aggregation. Aggregating local features into richer regional features has a long research history; crucial to the success of traditional descriptor based approaches, popular methods include selecting discriminative patches in the input image (Singh et al., 2012; Gordo, 2015), regressing saliency map scores at different resolutions (Jiang et al., 2013), aggregating SIFT descriptors with coding and pooling schemes (Boureau et al., 2010), or mining frequent patterns in sets of SIFT features (Singh et al., 2012). More recently, predicting mid-level representations of input images, such as depth maps or segmentations, has been used as auxiliary objectives to improve training in reinforcement learning (Sax et al., 2019). Biological studies have shown that similar phenomena are at play in the human brain (Long et al., 2018).

[5]The much larger GLDv2-clean dataset (1.5M images) (Yokoo et al., 2020) is sometimes used for training; after manual inspection we have found significant overlap between the training set classes of GLD-v2-clean and the $\mathcal{R}$Oxford and $\mathcal{R}$Paris query landmarks, which departs from usual evaluation protocols used in image retrieval. See Appendix C.7 for more details.

method	FCN	Mem (GB)	$\mathcal{R}$Oxford med	$\mathcal{R}$Oxford hard	$\mathcal{R}$Oxford +$\mathcal{R}$1M med	$\mathcal{R}$Oxford +$\mathcal{R}$1M hard	$\mathcal{R}$Paris med	$\mathcal{R}$Paris hard	$\mathcal{R}$Paris +$\mathcal{R}$1M med	$\mathcal{R}$Paris +$\mathcal{R}$1M hard
Global descriptors										
RMAC (Gordo et al., 2016)	R101	7.6	60.9	32.4	39.3	12.5	78.9	59.4	54.8	28.0
AP-GeM‡ (Revaud et al., 2019a)	R101	7.6	67.1	42.3	47.8	22.5	80.3	60.9	51.9	24.6
GeM+SOLAR (Ng et al., 2020)	R101	7.6	69.9	47.9	53.5	29.9	81.6	64.5	59.2	33.4
Global descriptors + reranking with local features										
DELG (Cao et al., 2020)	R50	7.6	75.1	54.2	61.1	36.8	82.3	64.9	60.5	34.8
DELG (Cao et al., 2020)	R101	7.6	*78.5*	*59.3*	*62.7*	*39.3*	*82.9*	*65.5*	*62.6*	*37.0*
Local features + ASMK matching (max. 1000 features per image)										
DELF (Noh et al., 2017)	R50^{-}	9.2	67.8	43.1	53.8	31.2	76.9	55.4	57.3	26.4
DELF-R-ASMK (Teichmann et al., 2019)	R50^{-}	27.4	76.0	52.4	<u>64.0</u>	<u>38.1</u>	<u>80.2</u>	58.6	<u>59.7</u>	29.4
HOW (Tolias et al., 2020)	R50^{-}	7.9	<u>78.3</u>	<u>55.8</u>	63.6	36.8	80.1	<u>60.1</u>	58.4	<u>30.7</u>
FIRe (ours)	R50^{-}	**6.4**	**81.8**	**61.2**	**66.5**	**40.1**	**85.3**	**70.0**	**67.6**	**42.9**
(standard deviation over 5 runs)			(±0.6)	(±1.0)	(±0.8)	(±1.1)	(±0.4)	(±0.6)	(±0.7)	(±0.8)
(mAP gains over HOW)			**(↑ 3.5)**	**(↑ 5.4)**	**(↑ 2.9)**	**(↑ 3.3)**	**(↑ 5.2)**	**(↑ 9.9)**	**(↑ 9.2)**	**(↑ 12.2)**

Table 3: **Comparison to the state of the art.** All models are trained on SfM-120k. FCN denotes the fully-convolutional network backbone, with R50^{-} denoting a ResNet-50 without the last block. Memory is reported for the image representation of the full $\mathcal{R}$1M set (without counting local features for the global descriptors + reranking methods). ‡ result from (Tolias et al., 2020). **Bold** denotes best performance, <u>underlined</u> second best among methods using ASMK, *italics* second best overall.

Supervision for local features. Several works focus on providing supervision at the level of local features in the context of contrastive learning. Xie et al. (2021) and Chen et al. (2021) obtain several views of an input image using data augmentations with known pixel displacements, such as crops and offsets. Similarly, Liu et al. (2020) train a model to predict the probability for a pair of local features to match, evaluated using known displacements. Wang et al. (2020) and Zhou et al. (2021) obtain local supervision by relying on epipolar coordinates and relative camera poses between shots. Positive pairs in image retrieval depart from these setups, as pixel-level correspondences cannot be known. To build matches without such signal, Wang et al. (2021a) use a standard nearest neighbor algorithm to build pairs of features, similarly to our approach, but without the use of filtering which is critical to our final performance. Using the current model predictions to define targets is reminiscent of modern self-supervised learning approaches which learn without any label (Caron et al., 2021; Grill et al., 2020). The additional filtering step can be seen as a way to keep only the most reliable model predictions, similar to *self-distillation* as in *e.g.* (Sohn et al., 2020). Other recent approaches for supervising local features include exploring set-level optimization strategies to define set-level alignment (Wang et al., 2021b) or approximating optimal transport kernels (Mialon et al., 2021).

Attention modules. Our LIT module is an iterative variant of standard attention (Bahdanau et al., 2015; Vaswani et al., 2017), modified to map a variable number of input features to an ordered set of N output features. The Perceiver model (Jaegle et al., 2021) has demonstrated the flexibility of such constructions by using it to scale attention-based deep networks to large inputs. Our design was heavily inspired by *slot-attention* (Locatello et al., 2020), but has some key differences that enable us to achieve high performance in more complex visual environments: a) unlike the slot attention which initializes its slots with i.i.d sampling, we *learn* the initial templates and therefore define an ordering on the output set, a crucial property for selecting promising matches; b) we replace the recurrent network gates with a residual connection across iterations. These modifications, together with the attention decorrelation loss enable our module to go from a handful of object-oriented slots to a much larger set of mid-level features. For object detection, Carion et al. (2020) rely on a set of learned *object queries* to initialize a stack of transformer layers. Unlike ours, their module is not recurrent; Appendix B.1 experimentally shows substantial benefits from applying LIT T times, with weight sharing, to increase model flexibility without adding extra parameters.

6 Conclusions

We present an approach for encoding mid-level features for image retrieval, a task that has up to now been dominated by approaches that work at the local feature level. We design an attention mechanism that outputs an ordered set of such features that are more discriminative and expressive than local features. Exploiting their ordered nature and without any extra supervision, we present a loss working directly on these mid-level features Our method not only significantly improves performance, but also requires less memory, a crucial requirement for scalable image retrieval.

REFERENCES

Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In *Proc. ICCV*, 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, 2015.

Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *Proc. CVPR*, 2010.

Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Proc. ECCV*, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *Proc. ECCV*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021.

Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *Proc. ICCV*, 2021.

Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proc. CVPR*, 2019.

Gabriela Csurka and Martin Humenberger. From handcrafted to deep local invariant features. *arXiv preprint arXiv:1807.10254*, 2018.

Gabriela Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Proc. ECCV Worshops*, 2004.

Albert Gordo. Supervised mid-level features for word image representation. In *Proc. CVPR*, 2015.

Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Proc. ECCV*, 2016.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Kavukcuoglu Koray, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Proc. NeurIPS*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proc. ICML*, 2021.

Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.

Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proc. CVPR*, 2013.

Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Proc. NeurIPS*, 2020.

Bria Long, Chen-Ping Yu, and Talia Konkle. Mid-level visual features underlie the high-level categorical organization of the ventral stream. *Proc. National Academy of Sciences*, 2018.

David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

Grégoire Mialon, Dexiong Chen, Alexandre d'Aspremont, and Julien Mairal. A trainable optimal transport embedding for feature aggregation and its relationship to attention. In *Proc. ICLR*, 2021.

Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. Solar: second-order loss and attention for image retrieval. In *Proc. ECCV*, 2020.

Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proc. ICCV*, 2017.

Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proc. CVPR*, 2010.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, 2008.

Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proc. CVPR*, 2018a.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Trans PAMI*, 2018b.

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. ICCV*, 2019a.

Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: Reliable and repeatable detector and descriptor. In *Proc. NeurIPS*, 2019b.

Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proc. CVPR*, 2018.

Alexander Sax, Jeffrey O Zhang, Bradley Emi, Amir Zamir, Silvio Savarese, Leonidas Guibas, and Jitendra Malik. Learning to navigate using mid-level visual priors. In *Proc. CoRL*, 2019.

Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *Proc. ECCV*, 2012.

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Proc. NeurIPS*, 2020.

Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proc. CVPR*, 2019.

Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *Proc. ICCV*, 2013.

Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *Proc. ECCV*, 2020.

Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 1980.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017.

Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Proc. ECCV*, 2020.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proc. CVPR*, 2021a.

Zhaoqing Wang, Qiang Li, Guoxin Zhang, Pengfei Wan, Wen Zheng, Nannan Wang, Mingming Gong, and Tongliang Liu. Exploring set similarity for dense self-supervised representation learning. *arXiv preprint arXiv:2107.08712*, 2021b.

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. CVPR*, 2020.

Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proc. CVPR*, 2015.

Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proc. CVPR*, 2021.

Shuhei Yokoo, Kohei Ozaki, Edgar Simo-Serra, and Satoshi Iizuka. Two-stage discriminative re-ranking for large-scale landmark retrieval. In *Proc. CVPR Worshops*, 2020.

Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proc. CVPR*, 2021.

Appendix

Table of Contents

A Appendix summary **13**

B Additional ablations **14**
B.1 Impact of the iteration and the number of templates hyper-parameters 14
B.2 Impact of hard negatives 14
B.3 Impact of the update function 15

C Analysis and discussions **15**
C.1 Single-scale results 15
C.2 Consistency of attention maps across scales 16
C.3 Redundancy in Super-features 16
C.4 Are all Super-features trained? 16
C.5 Using the Super-feature loss with local features 17
C.6 Pretraining the backbone together with LIT 17
C.7 Is Google Landmarks v2 clean an appropriate training dataset for testing on $\mathcal{R}$Oxford and $\mathcal{R}$Paris? 17

D Application to visual localization **18**

A Appendix summary

In this appendix, we present additional ablations (Appendix B), a deeper analysis of our framework (Appendix C) as well as results of the application of our FIRe framework for image retrieval to the task of visual localization (Appendix D). We briefly summarize the findings in the following paragraphs.

Ablations. We study the impact of some hyper-parameters of our model, namely the size N of the set of Super-features, and the number of iterations T in the LIT module, in Appendix B.1. We show that 256 Super-features and 6 iterations offer the best trade-off between performance and computational cost. We also show in Appendix B.2 that we obtain further performance gains of over 1% by increasing the number of negatives to 10. Finally, we study the impact of replacing the residual connection inside the LIT module by a recurrent network (Appendix B.3).

Properties of Super-features. We show single-scale results in Appendix C.1. In Appendix C.2, we display the attention maps of Super-features, at different scales for a fixed Super-feature ID. We further study the amount of redundancy in Super-features, compared to local features, in Appendix C.3. Next, we verify in Appendix C.4 that all Super-features receive training signal, as a sanity check. We discuss the case of applying a loss directly on local features in Appendix C.5 and give details about the pretraining on ImageNet in Appendix C.6.
Our main conclusions are:

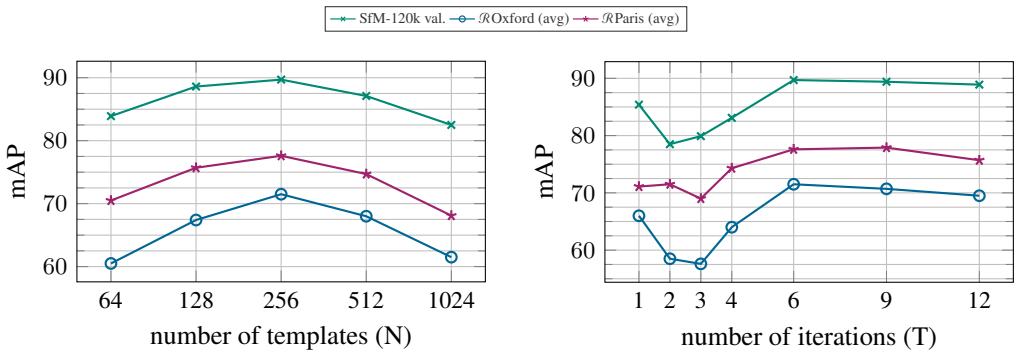
- FIRe still outperforms HOW when using a single scale at test time.
- Attention maps are *highly consistent* across scales.
- Super-features are *less redundant* than local ones.

Localization. We further evaluate our model using a visual localization setting on the Aachen Day-Night v1.1 dataset (Sattler et al., 2018) in Appendix D. To do so, we leverage a retrieval + local feature matching pipeline, and show that it is beneficial to use our method, especially in hard settings.

B ADDITIONAL ABLATIONS

B.1 IMPACT OF THE ITERATION AND THE NUMBER OF TEMPLATES HYPER-PARAMETERS

In this section, we perform ablations on the number of Super-features N and the number of iterations T in the Local feature Integration Transformer. We first study the impact of the number N of Super-features extracted for each scale of each image in Figure 7a. We observe that the best value is 256 on both the validation and test sets ($\mathcal{R}$Oxford and $\mathcal{R}$Paris). We then study the impact of the number of iterations T in Figure 7b. While the performance decreases when doing 2 or 3 iterations compared to just 1, a better performance is reached for 6 iterations, after which the performance saturates while requiring more computations. We thus use $T = 6$.



(a) Impact of the number of Super-features

(b) Impact of the number of iterations

Figure 7: **Varying the number of Super-features N and of iterations T** in the Local feature Integration Transformer.

B.2 IMPACT OF HARD NEGATIVES

Each training tuple is composed of one image pair depicting the same landmark, and 5 negative images (*i.e.* from different landmarks). We plot the performance when varying this number in Figure 8. We observe that adding more negatives improves overall the performance on all datasets, *i.e.* by more than 1% when increasing it to 10 negatives, but at the cost of a longer training time as more images need to be processed at each iteration. This is why we use 5 negatives in the rest of the paper as it offers a good compromise between performance and cost.

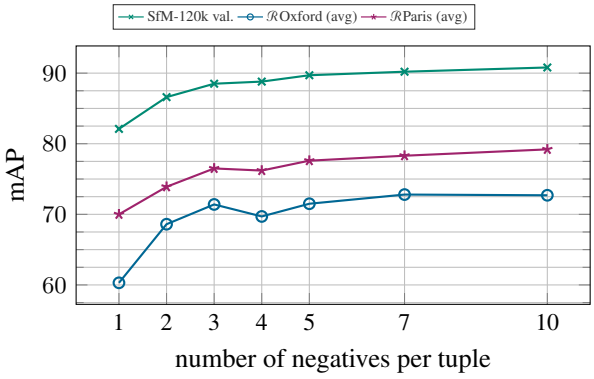


Figure 8: **Varying the number of hard negatives per training tuple.**

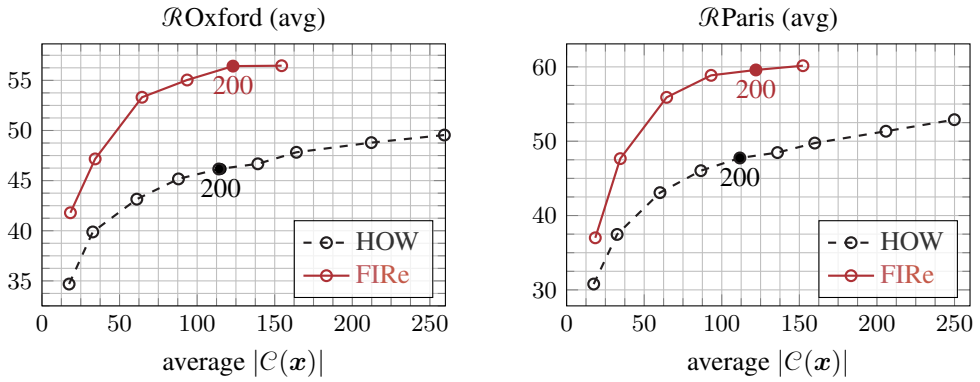


Figure 9: **Performance versus memory when varying the number of selected features at a single scale** for HOW and FIRe. The x axis represents the average number of vectors per image in ASMK, which is proportional to the memory, when varying the number of selected features in $(25, 50, 100, 150, 200, 300, 400, 500, 600)$, FIRe is limited to 256 features.

B.3 Impact of the update function

The formula for ψ in Equation (5) sums the previous $\mathcal{Q}$ with the output of the attention component $\boldsymbol{\alpha} \cdot V(\mathcal{U})$, *i.e.*, with a residual connection. This is a different choice than the one made in the object-centric slot attention of Locatello et al. (2020), which proposes to use a Gated Recurrent Unit: $\psi(\mathcal{U}; \mathcal{Q}) = \texttt{GRU}(\boldsymbol{\alpha} \cdot V(\mathcal{U}), \mathcal{Q})$. We thus compare the residual connection we use to a GRU function and report results in Table 4 with $T = 3$. We observe that the residual connection reaches a better performance in all datasets while having the interest of not adding extra parameters.

update function	SfM-120k val	$\mathcal{R}$Oxford med	$\mathcal{R}$Oxford hard	$\mathcal{R}$Paris med	$\mathcal{R}$Paris hard
residual	**79.9**	**71.3**	**43.9**	**78.7**	**59.4**
GRU	74.7	66.7	41.4	77.2	57.8

Table 4: **Impact of the update function** in the LIT module. We compare the performance of the residual combination of the previous template value with the cross-attention tensor compared to a GRU as used in slot attention (Locatello et al., 2020). In this experiment, we use $T = 3$.

In summary, we propose the LIT module to obtain a few hundred mid-level features for image retrieval while slot attention handles a handful of object attentions. Technical differences to the slot attention include: a) we use a learned initialization of the templates instead of i.i.d. sampling, which allows to obtain an ordered set of features, and thus to apply the constraint on the ID for the matching, leading to a clear gain, see Table 1 and Figure 3, b) to handle a larger number of templates, we also add a decorrelation loss on the attention maps, which has clear benefit, see Table 2 and Figure 4, c) we use a residual connection with 6 iterations instead of a GRU which performs better for our task (see Table 4).

C Analysis and discussions

C.1 Single-scale results

Similar to Figure 5 of the main paper, we perform the same ablation in Figure 9 when while extracting features at a single scale (1.0). We observe that FIRe significantly improves the mAP on the two datasets compared to HOW. The average number of clusters, *i.e.* the memory footprint of images, remains similar for both methods at a same number of selected features. This stands in contrast to the multi-scale case where our approach allows to save memory (about 20%), which we hypothesize is due to the correlation of our features across scales that we discuss below.

C.2 CONSISTENCY OF ATTENTION MAPS ACROSS SCALES

At test time, we extract Super-features at different image resolutions. We propose here to study if the attention maps of the Super-features across different image scales are correlated. We show in Figure 10 the attention maps at the last iteration of LIT for the different image scales. We observe that they fire at the same image location at the different scales. Note that the attention maps are larger/smoother at small scale (right columns), as for visualization, we resize lower resolution attention maps to the original image size.

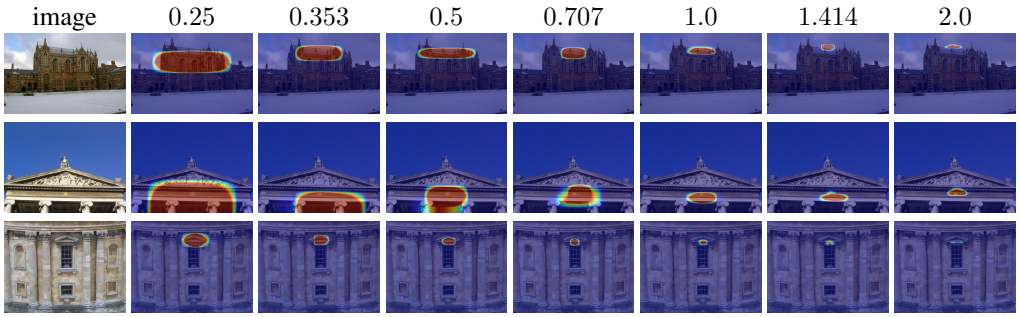


Figure 10: **Attention consistency across scales.** For three images, we select one Super-feature per image and show the attention maps of the latest iteration of LIT for different image scales, with from left to right $0.25, 0.353, 0.5, 0.707, 1.0, 1.414$ and 2.0. We clearly observe that attention maps are correlated. They show larger regions at small scale as for visualization, we resize the lower resolution attention to the original image size.

C.3 REDUNDANCY IN SUPER-FEATURES

To evaluate the redundancy of Super-features versus local features, Figure 11 displays the average cosine similarity between every local features / Super-feature and its K nearest local features / Super-features from the same image, for different values of K. We observe that Super-features are significantly less correlated to the most similar other ones, compared to local features used in HOW.

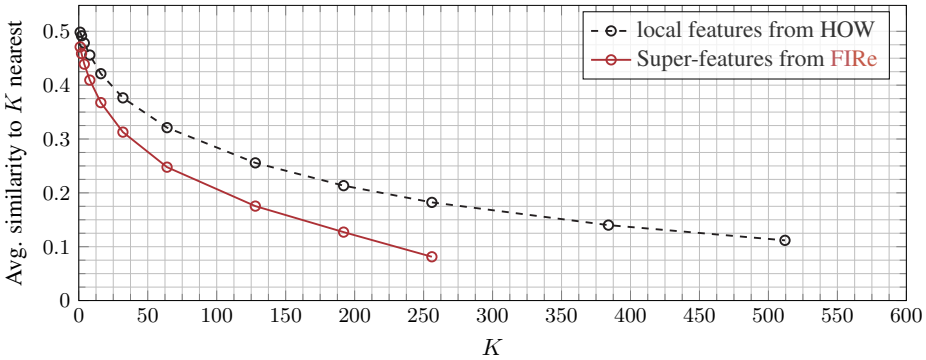


Figure 11: **Measuring Super-feature redundancy.** We compute the cosine similarity between every pair of features (local features from HOW or Super-features from FIRe) coming from a given image, and report their average cosine similarity to the top-K most similar features from the same image. Results are averaged over the 70 query images of the $\mathcal{R}$Oxford dataset. Only 256 Super-features can be extracted per image, explaining its maximum x value.

C.4 ARE ALL SUPER-FEATURES TRAINED?

The loss in Equation (8) only operates on a subset of the Super-feature pairs that pass the criteria of Equation (7); if a Super-feature ID is not matched, it receives no training signal. To investigate

if all of the IDs contribute to the loss, we monitored how many times each ID is matched at each epoch and report the percentage over all 2000 training tuples per epoch. In Figure 12 we report mean, standard deviation and minimum number of matches for the N Super-feature ID. We clearly observe that all Super-features receive training signals regularly and each Super-feature is matched about one quarter of the time on average.

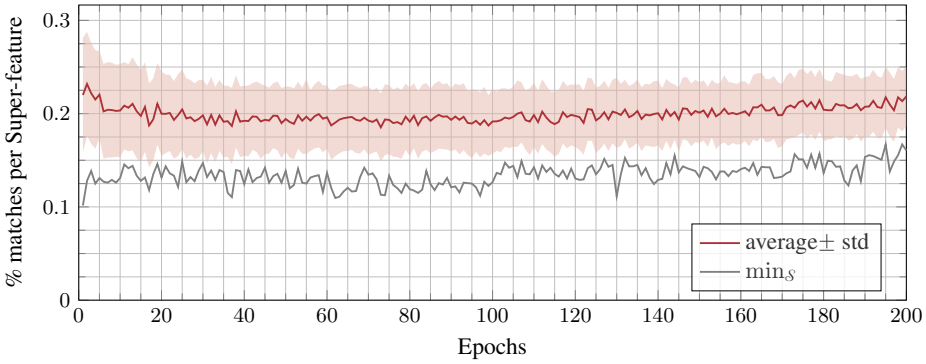


Figure 12: **Are all Super-features trained?** As a sanity check, this plot shows the average number (in percentage) of times (with standard deviation) that the Super-features receive training signal, across training epochs. We also plot the minimum value, which is always significantly positive, showing that each Super-feature ID receives training signal. Thus our loss $\mathcal{L}_{\text{super}}$ proposed in Equation (8) does not lead to degenerate solutions where some IDs are never selected and never trained.

C.5 Using the Super-feature loss with local features

It is worth noting that the loss in Equation (8) could also theoretically be used over local feature activations. It could be computed on pairs of local features that pass the conditions described in Equation (7), after removing the constraint on having the same Super-feature ID. Unfortunately, we were unable to get any gains over HOW (Tolias et al., 2020) when appending such a loss side-by-side with the global loss over local features. Empirically, our ablations (see Section 4.1) show that adding the constraint on the Super-features ID, which is only possible with ordered feature sets, is key to the success of our approach, and significantly improves the quality of the matching.

C.6 Pretraining the backbone together with LIT

Before training on landmark retrieval, we pretrain the backbone network, with the proposed LIT module appended after the last layer, for image classification on ImageNet-1K. Given that we remove the last convolutional block similar to HOW, to avoid overfitting we further append a classification head composed of one fully-connected layer, batch norm, leaky ReLU, dropout and another fully-connected layer at the end of the network and train it for 80 epochs using a standard cross-entropy loss and in addition the decorrelation loss $\mathcal{L}_{\text{attn}}$ weighted by 0.1, the same weight as during finetuning. We start with a learning rate of 0.01 and divide it by 10 every 20 epochs. This new architecture reaches a top-1 (resp. top-5) accuracy of 73.48% (resp. 91.54%) on the ImageNet validation set. Note that this performance is a bit lower than a standard ResNet-50: this is because we have removed the last convolutional block which contains most parameters in the ResNet architecture.

C.7 Is Google Landmarks v2 clean an appropriate training dataset for testing on $\mathcal{R}$Oxford and $\mathcal{R}$Paris?

In the comparison to the state of the art (Table 3), all reported methods are trained on the SfM-120k dataset. Several recent works have also released a model trained on the Google Landmarks v2 dataset (Weyand et al., 2020) or its clean version (Yokoo et al., 2020) with excellent performance on $\mathcal{R}$Paris. However, we find out that several of the query landmarks from $\mathcal{R}$Paris and $\mathcal{R}$Oxford were present in the training set of the cleaned version of Google Landmarks v2, such as 'La Defense', 'Eiffel Tower', 'Sacré Coeur' or 'Hotel les Invalides' for $\mathcal{R}$Paris or such as 'Mary Madgalen', 'Bodleian Library', 'All Soulds College' or 'Radcliffe' for $\mathcal{R}$Oxford.

D APPLICATION TO VISUAL LOCALIZATION

In this section, we evaluate FIRe for the task of visual localization, where retrieval is used as a first-stage filtering and before more precise, local feature-based geometric matching. To this end, we follow the pipeline proposed by Kapture[6] (Humenberger et al., 2020) on the Aachen Day-Night v1.1 dataset (Sattler et al., 2018). In this scenario, a global Structure-from-Motion map is built from the training images using R2D2 local descriptors (Revaud et al., 2019b). At test time, given a query image to localize, image retrieval is used to retrieve the top-50 nearest images. On these retrieved images, R2D2 local features are extracted and matched with the ones on the query image, and this is used to estimate the position of the camera. The percentage of successfully localized images within three levels of thresholds is then reported on the day or night images, following the visual localization benchmark protocol[7], the latter ones being more challenging as training images are taken during daytime.

We compare our retrieval method to AP-GeM (Revaud et al., 2019a) and HOW (Tolias et al., 2020) and report results in Table 5. AP-GeM is the default method used in Kapture (Humenberger et al., 2020). We observe that using FIRe leads to better visual localization, specially in the most challenging scenario of night image localization and the strictest localization threshold: Performance improves by 2% compared to AP-GeM and by 1.5% compared to HOW on night images at a threshold of 0.25m and 2°. Overall, for either day or night matching, FIRe either improves or performs on par to both methods compared.

Retrieval method	Day images			Night images		
	0.25m, 2°	0.5m, 5°	5m, 10°	0.25m, 2°	0.5m, 5°	5m, 10°
AP-GeM (Revaud et al., 2019a)	88.8	**96.6**	**99.6**	72.3	86.9	97.9
HOW (Tolias et al., 2020)	**90.8**	96.2	**99.6**	72.8	**90.1**	97.9
FIRe (ours)	<u>90.7</u>	<u>96.5</u>	<u>99.5</u>	**74.3**	**90.1**	**98.4**

Table 5: **Visual localization results.** Percentage of successfully localized images on the Aachen Day-Night v1.1 dataset when changing the retrieval method in the Kapture pipeline from Humenberger et al. (2020). In the most challenging scenario, *i.e.* night images at strictest localization threshold, using FIRe yields a 2% improvement compared to AP-GeM and a 1.5% improvement compared to HOW. **Bold** number denotes the best performance, <u>underlined</u> indicates performance within a 0.1 margin to the best one.

[6] https://github.com/naver/kapture-localization
[7] https://www.visuallocalization.net/