Improving Spatial and Semantic Context of Local Descriptors for Image Retrieval

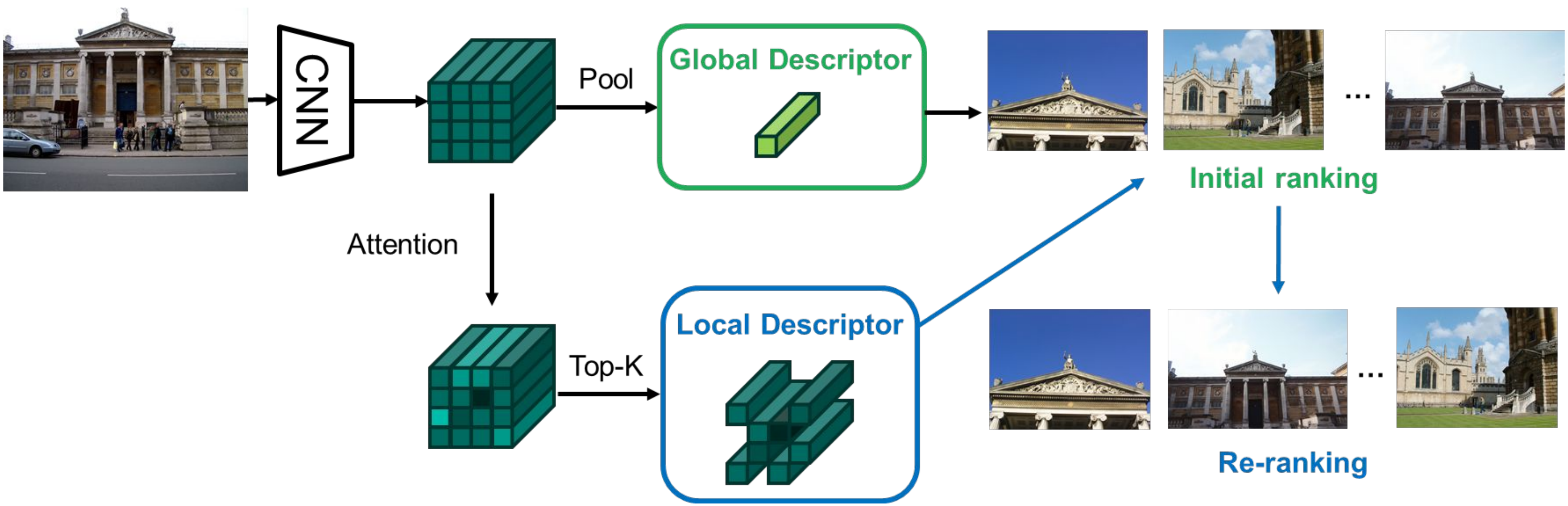
Jinhwan Seo, Kyu Beom Han
(서진환, 한규범)

2024/06/03

KAIST

Image Retrieval

- Global descriptors for initial ranking
- Local descriptors for re-ranking based on attention score



Framework of Image Retrieval

1. Global: Initial ranking (speed ↑ acc ↓)
2. Global + local: Initial + re-ranking (speed - acc -)
3. Local + matching: Matching (speed ↓ acc ↑)

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 (Tolias 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)**

KAIST

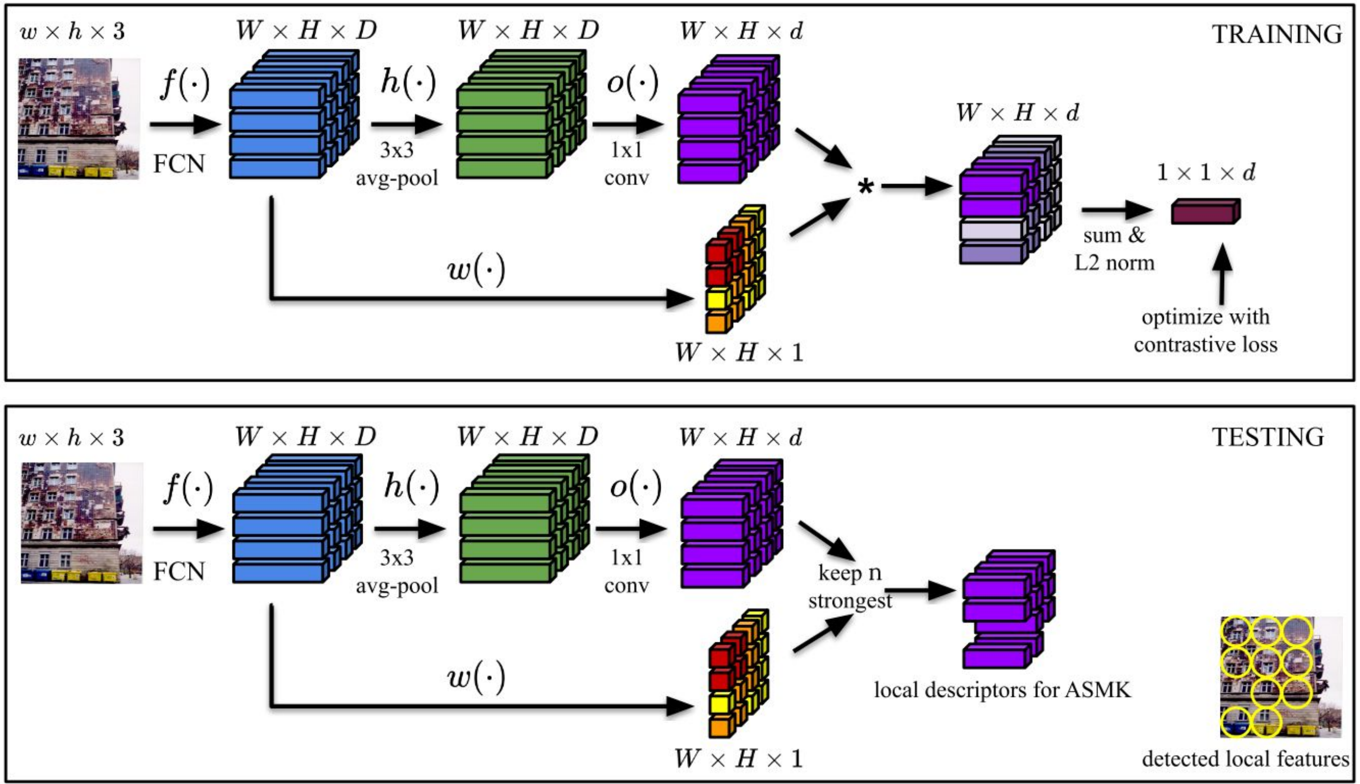
Framework of Image Retrieval

1. Global: Initial ranking (speed ↑ acc ↓)
2. Global + local: Initial + re-ranking (speed - acc -)
3. Local + matching: Matching (speed ↓ acc ↑)

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
1 *Global descriptors*										
RMAC (Tolias 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
2 *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*
3 *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	*64.0*	38.1	80.2	58.6	59.7	29.4
HOW (Tolias et al., 2020)	R50⁻	7.9	78.3	55.8	63.6	36.8	80.1	60.1	58.4	30.7
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**)

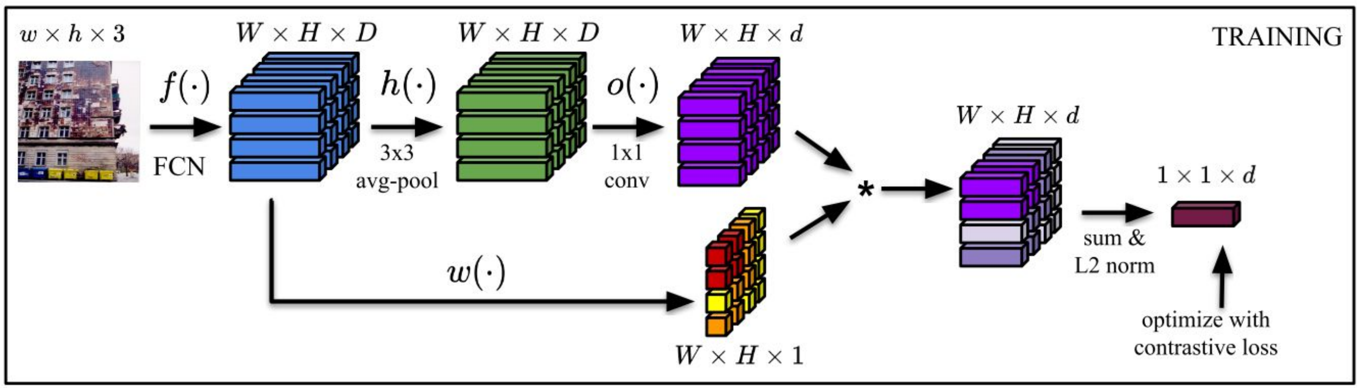
Gathering Local Descriptors

- Train with aggregate of local features from CNN backbone
- Use top-K local features as descriptors based on attention map



Motivation - Limited Local Context

- Relationships between local descriptors (i.e., local context) are discarded during training & matching



Training is done by simply taking the weighted-sum of local features

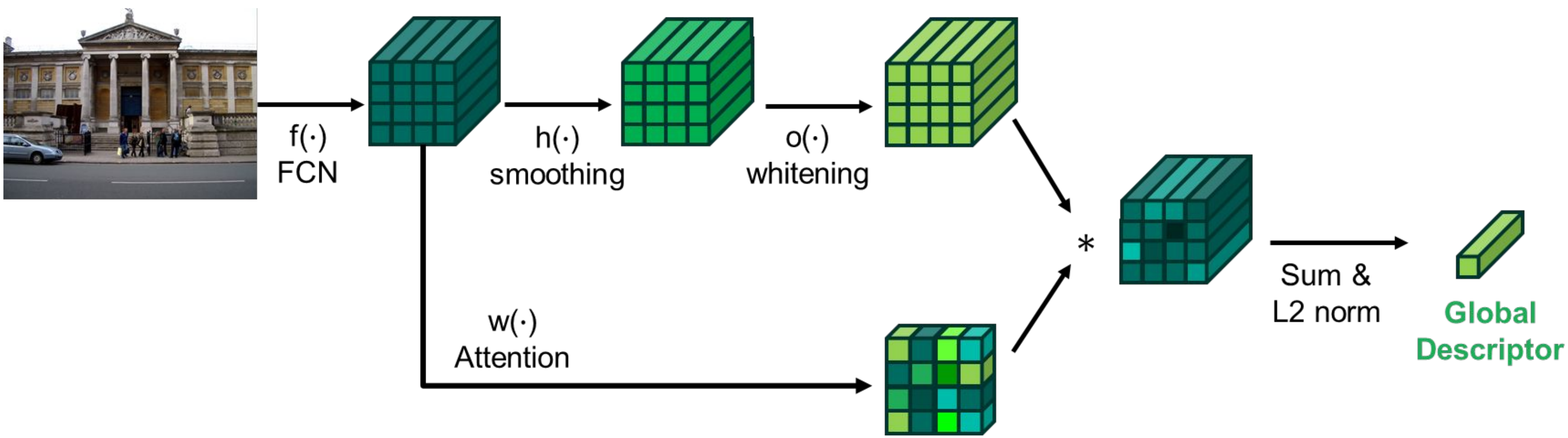
$\alpha = 3,\ \tau = 0.25$

Matching is done by taking the sum of similarity between matched local descriptors (ASMK)

$$\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} (x^T \cdot y)^{\alpha}$$

Previous work

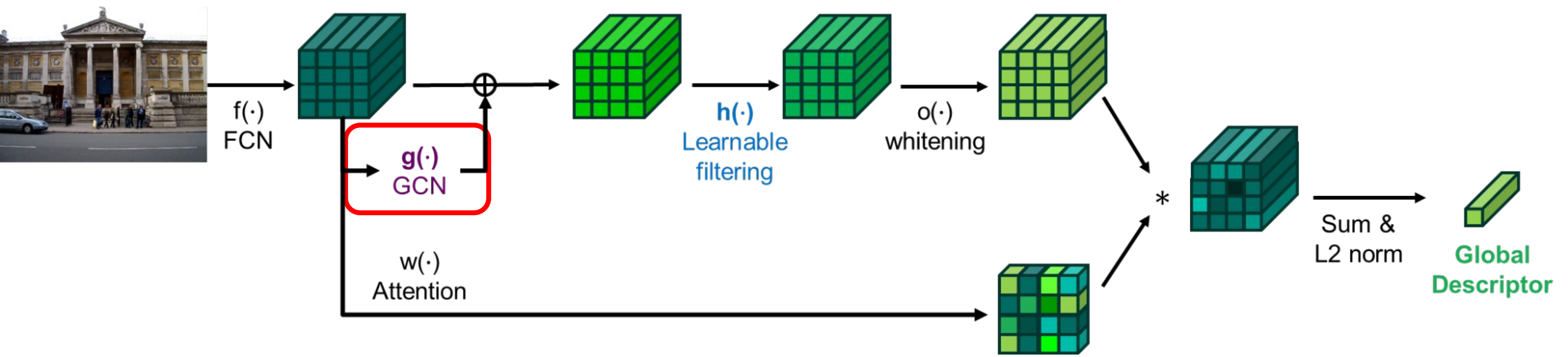
- f(·): Feature extractor - R18 / 50
- h(·): Local smoothing - Average pooling
- o(·): Whitening - 1x1 Conv
- w(·): Attention - L2 norm



KAIST

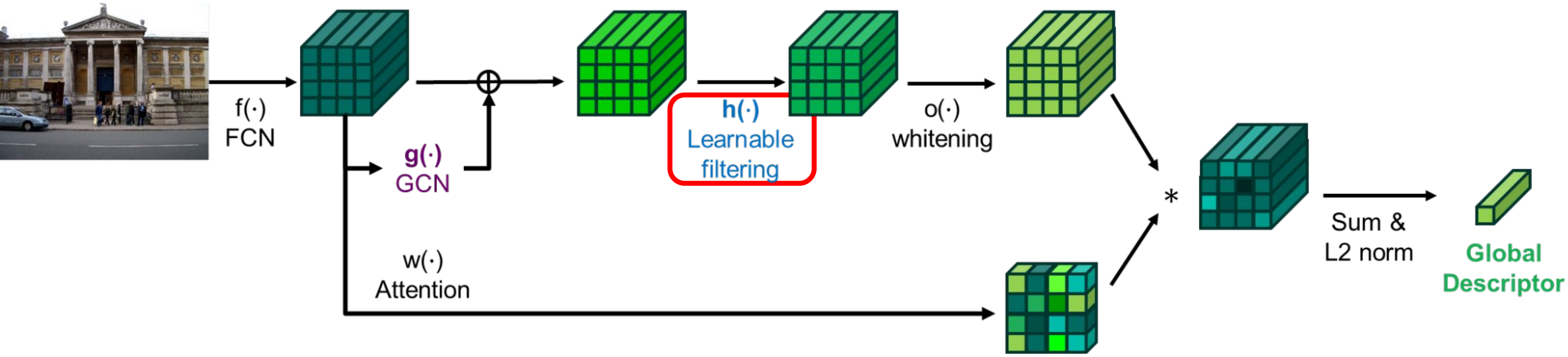
Our Goal

- Provide rich context information for local descriptors
  - Provide semantic information of local descriptors via **graph convolution**
  - Increase the effective region via **learnable smoothing** for local descriptors



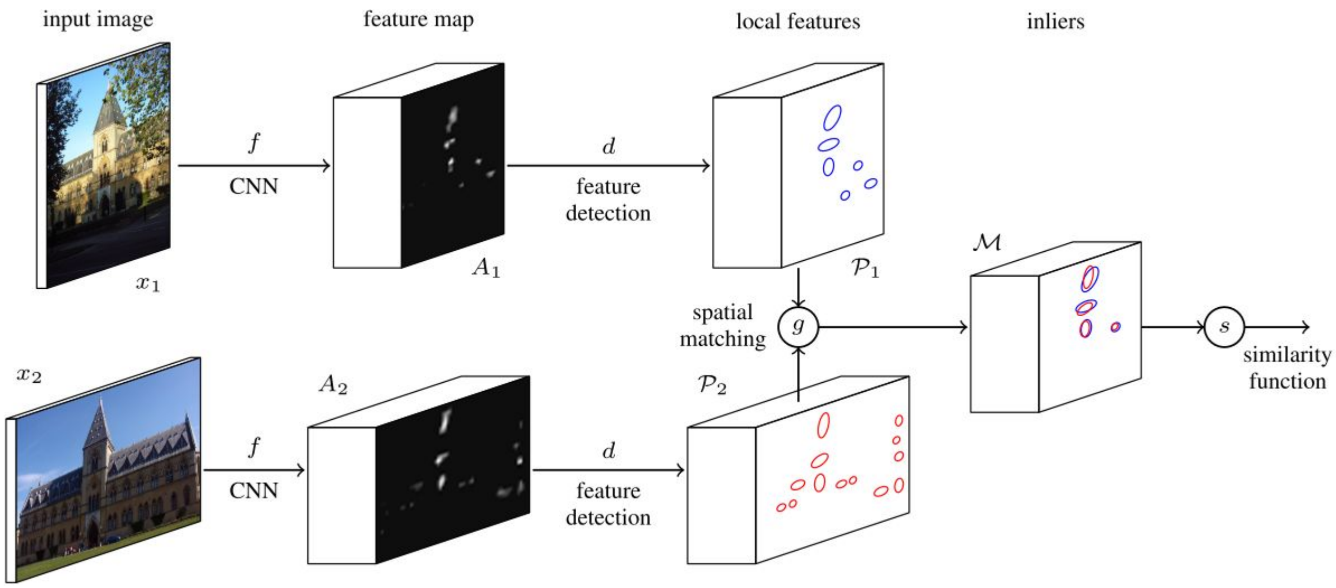
Our Goal

- Provide rich context information for local descriptors
  - Provide semantic information of local descriptors via **graph convolution**
  - Increase the effective region via **learnable smoothing** for local descriptors



Related Work: Fitting Ellipses

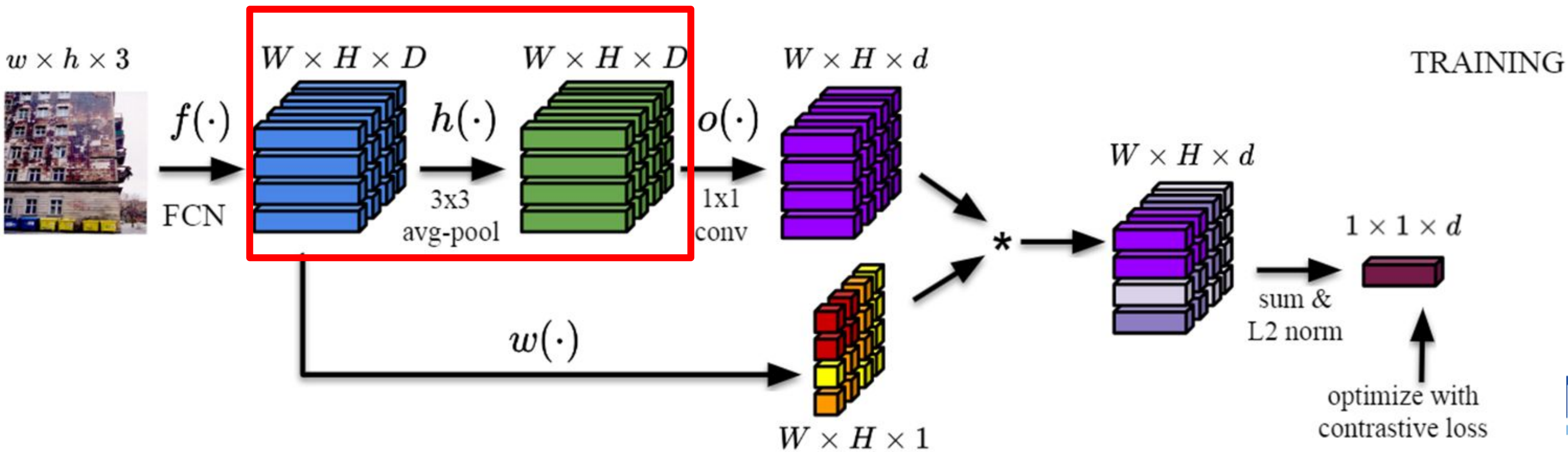
- Simeoni et al. represent a set of local feature as ellipse for matching
- Ellipses are aggregated as global descriptor for matching or used for re-ranking, but doesn't directly use them for retrieval



Related Work: Local Smoothing

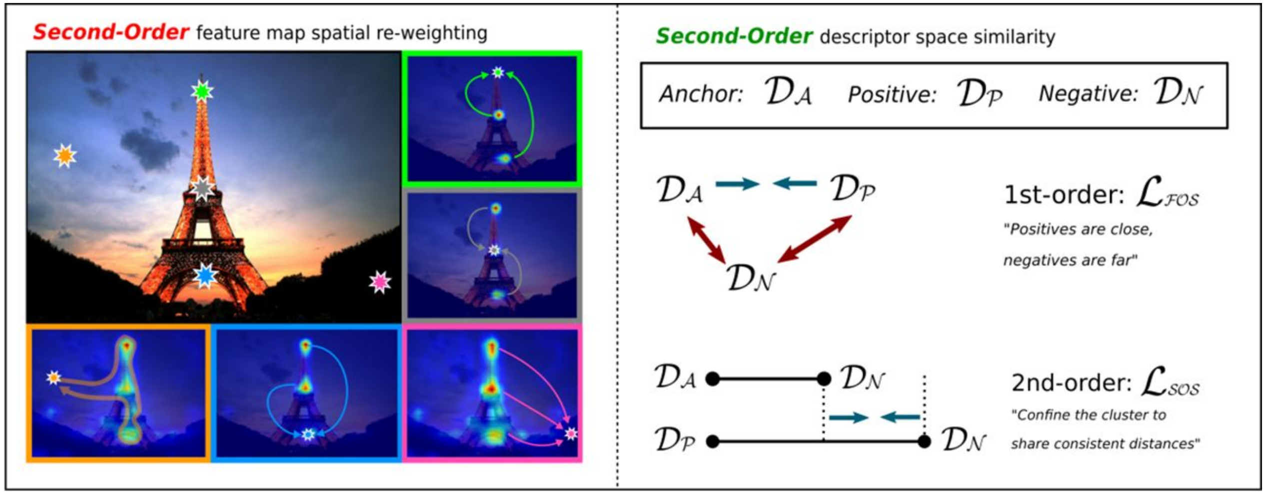
- Tolias et al. apply spatial avg. pooling to diffuse sparse local features to neighbors
- **Limitation:**
  - Unwanted smoothing may happened, reducing the importance of local descriptor

Method	Loss	Validation mAP	Validation $\lvert\mathcal{C}_\mathcal{X}\rvert$	$\mathcal{R}$Oxford mAP	$\mathcal{R}$Oxford $\lvert\mathcal{C}_\mathcal{X}\rvert$	Tiny-GLD$_2$ μAP	Tiny-GLD$_2$ $\lvert\mathcal{C}_\mathcal{X}\rvert$	
5: R18$_{\hat{h}\hat{o}\hat{w}}$	CE	75.5 ± 1.3	391.0 ± 8.2	63.7 ± 1.6	442.3 ± 9.7	64.0 ± 1.8	427.5 ± 15.6	w/o smoothing
8: R18$_{h\hat{o}\hat{w}}$	CE	77.0 ± 0.9	279.6 ± 5.6	65.4 ± 0.5	320.6 ± 6.8	68.6 ± 1.8	300.9 ± 11.4	w/ smoothing



Related Works - SOLAR

- Re-weighting <u>local</u> descriptor
- Confine clusters with second-order loss
- **Limitation:**
  - Attention map requires expensive computational cost

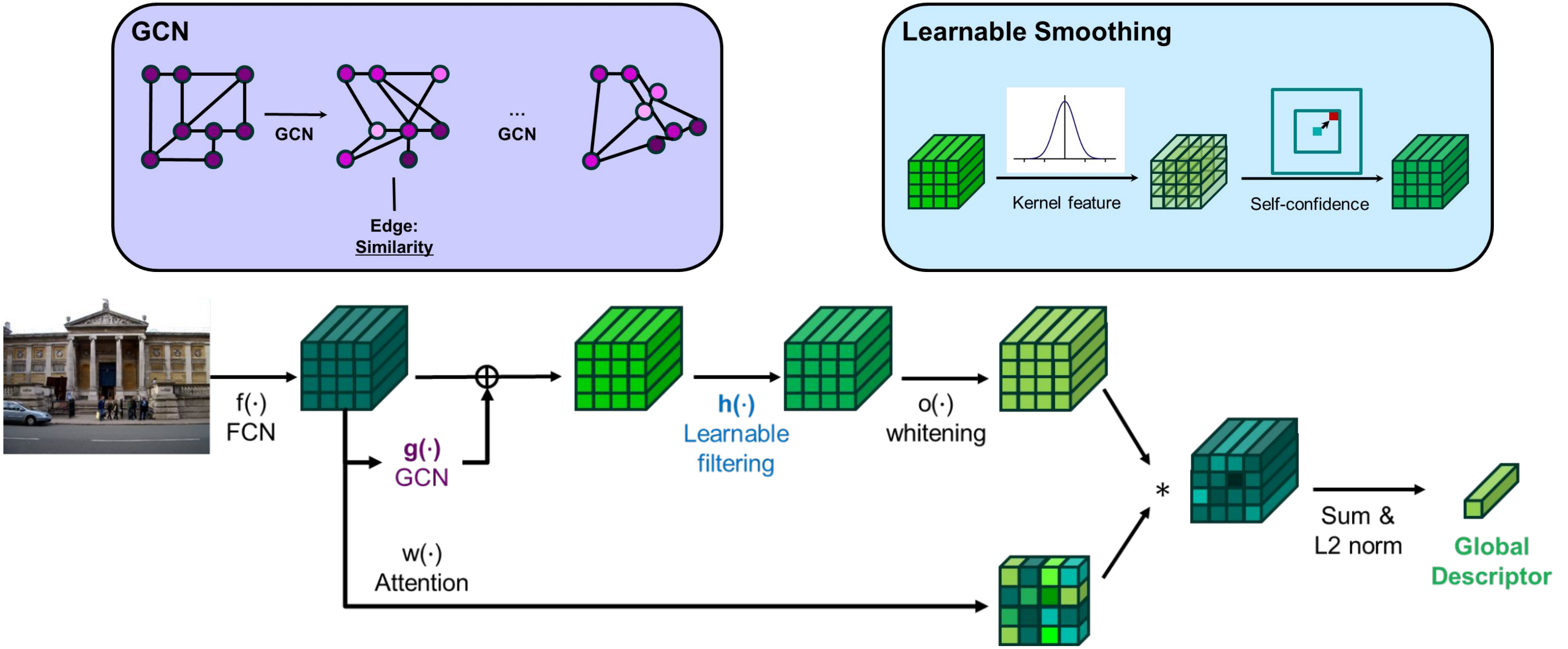


Re-weighting <u>local descriptors</u> prior to GeM

Distribute <u>global descriptors</u> in descriptor space

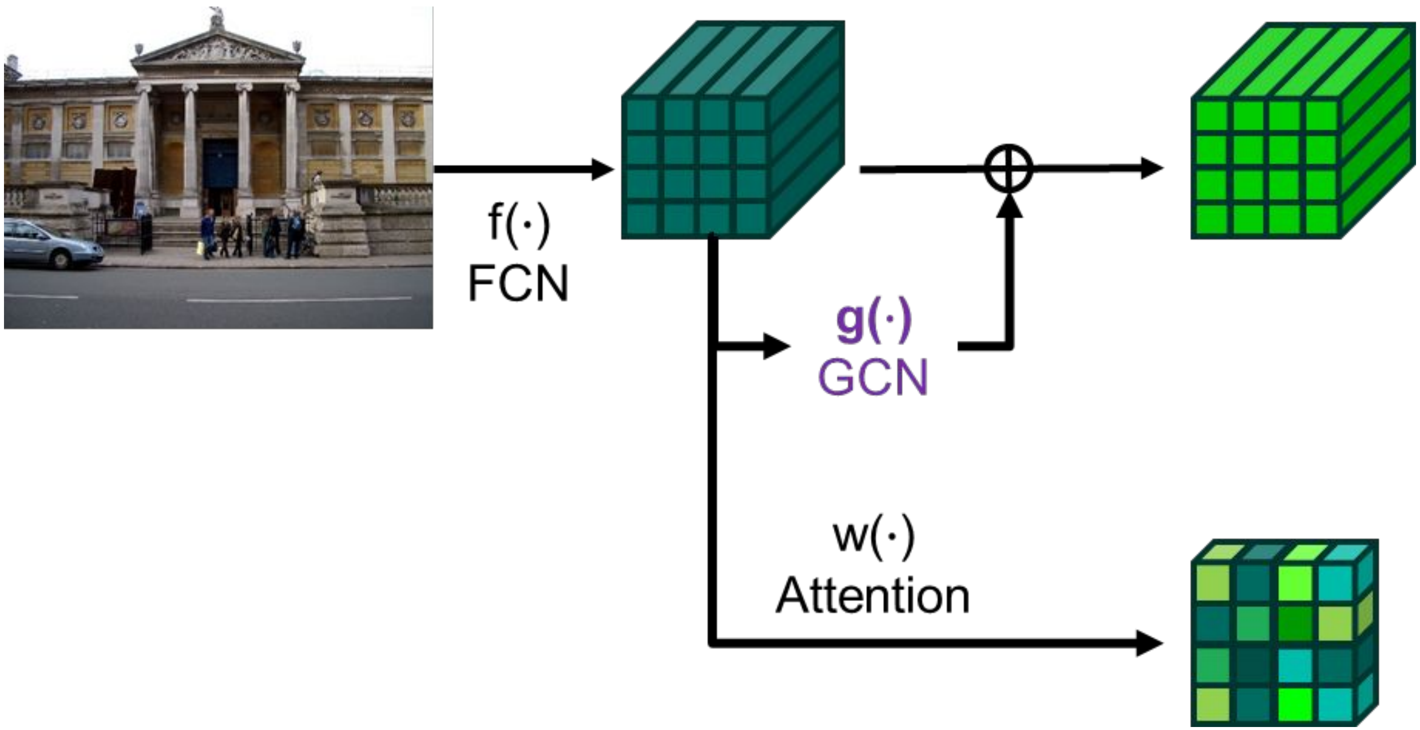
Overall Method

- Semantic context re-weighting : <u>Semantic context</u>
- Learnable smoothing : <u>Local spatial context</u>



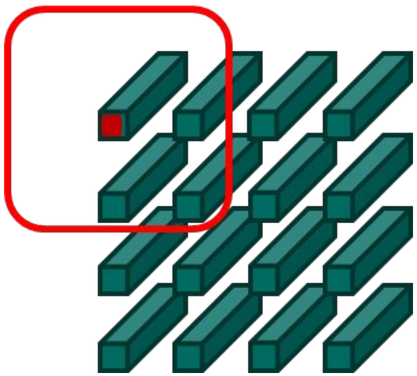
Method 1. Graph-based Refinement

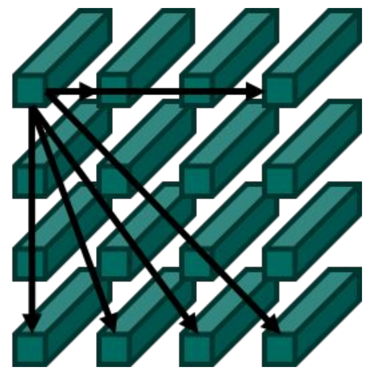
- Learnable smoothing focus on limited region of locality
- Re-weight local features via GCN
- GCN captures semantic information

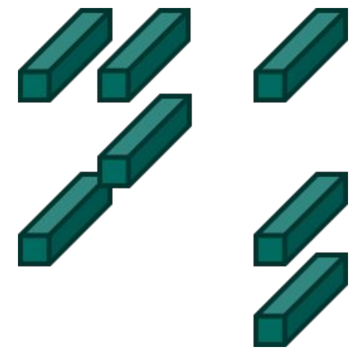


Method 1.1 Graph Structure

- Nodes are connected with edges (similarity)
- Each GCN propagates messages to next block
- Can consider global semantic context



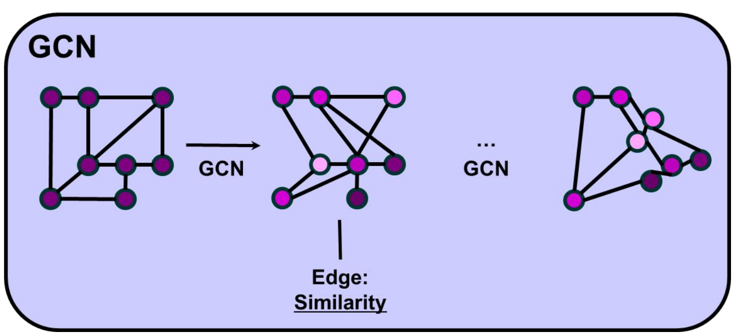




Learnable smoothing

Calculate similarity for all nodes

Connected nodes can consider global context



Attention Map Visualization

HOW

Ours

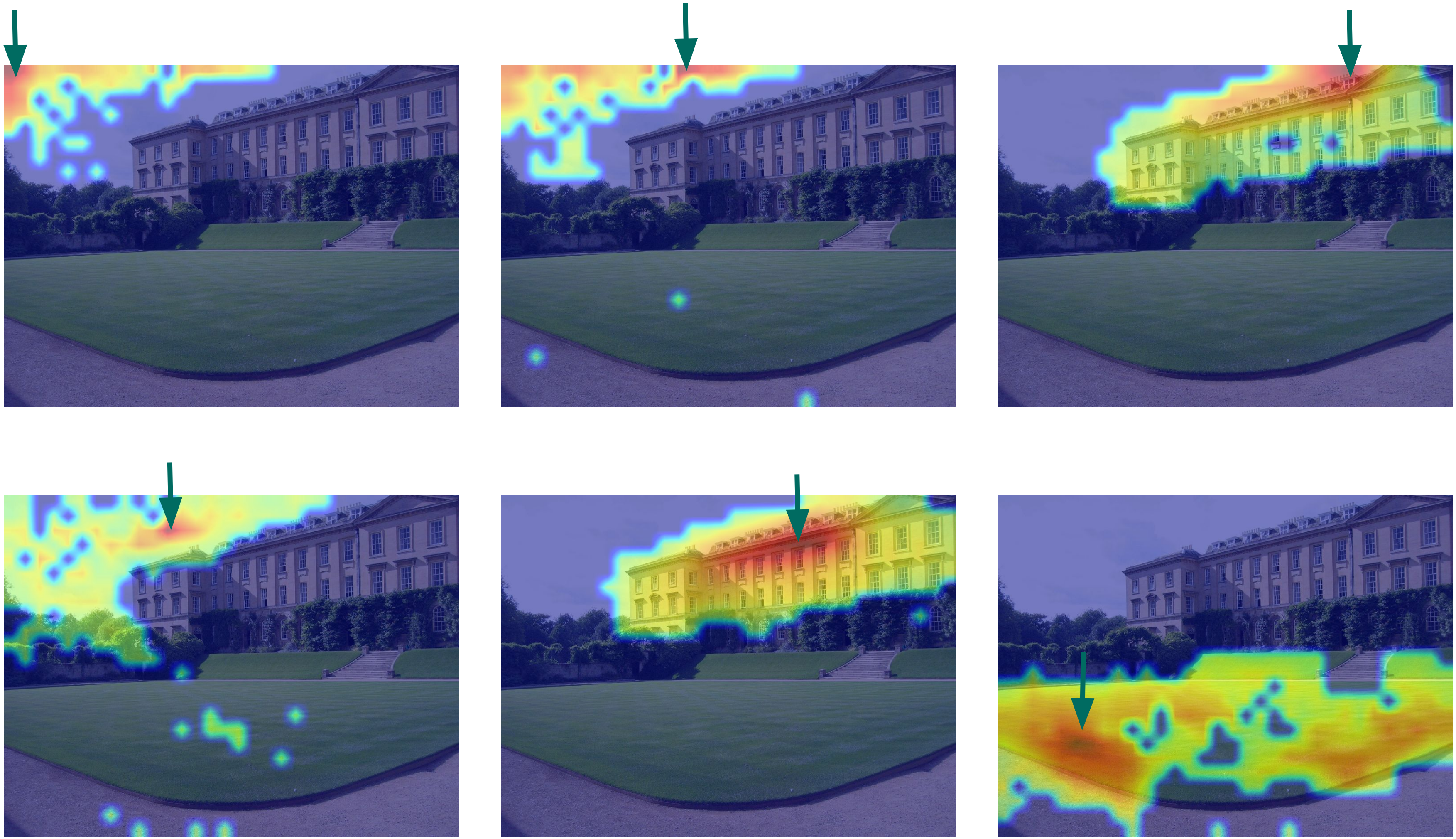
Attention Map Visualization

HOW

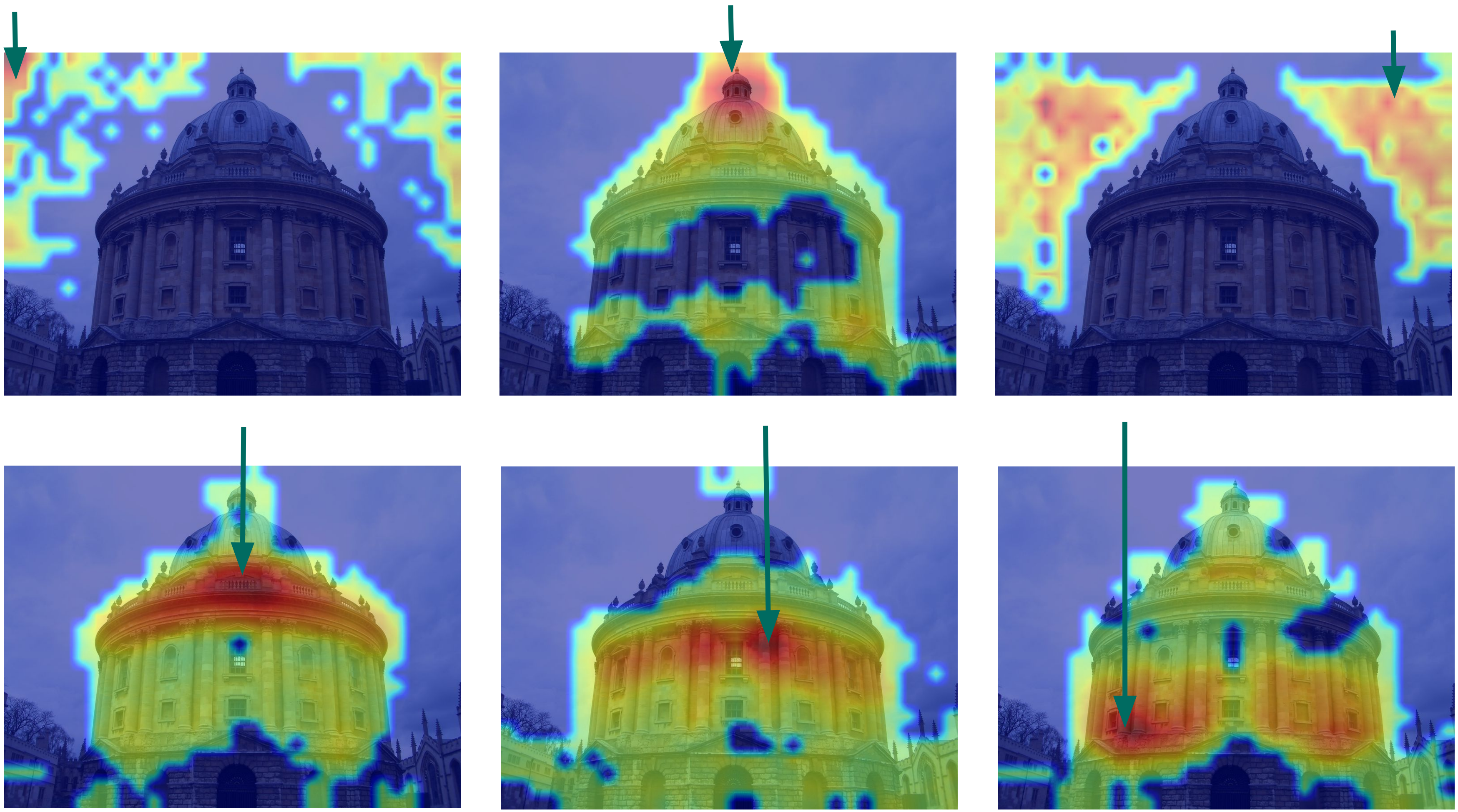
Ours

KAIST

Edge connection map

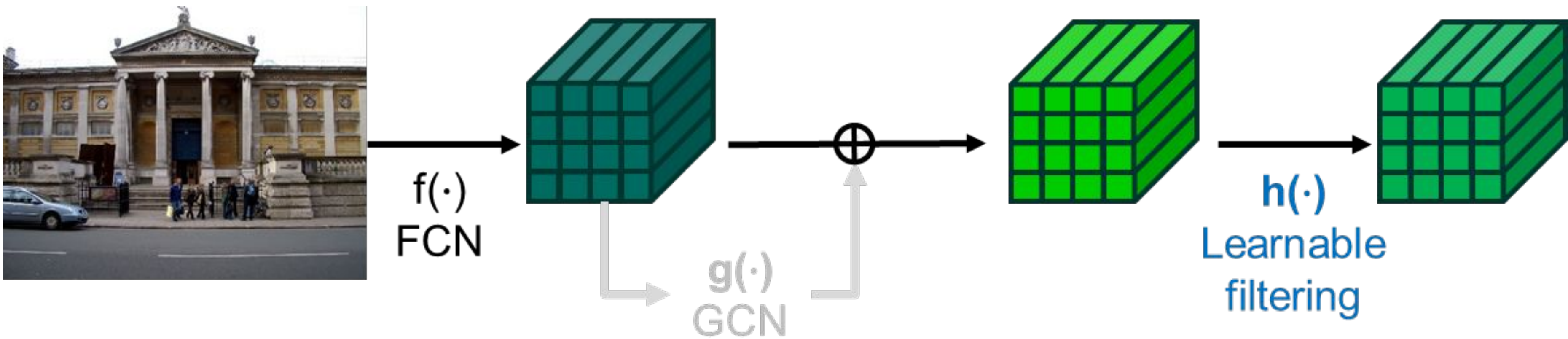


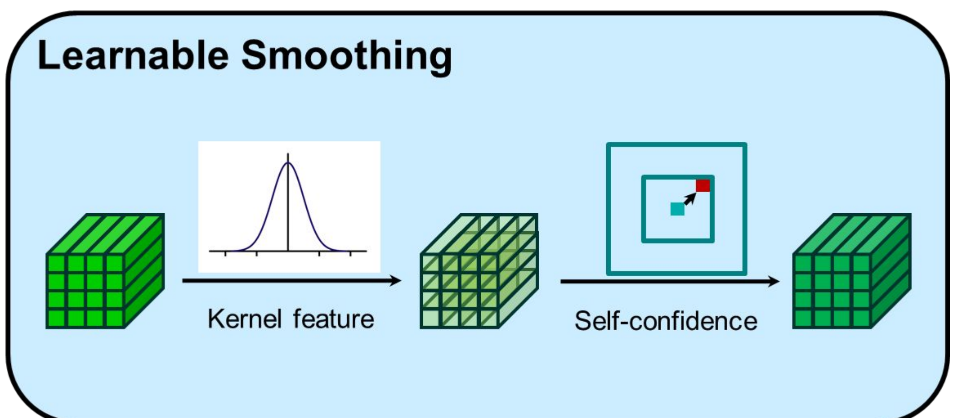
Edge connection map



Method 2. Learnable Smoothing

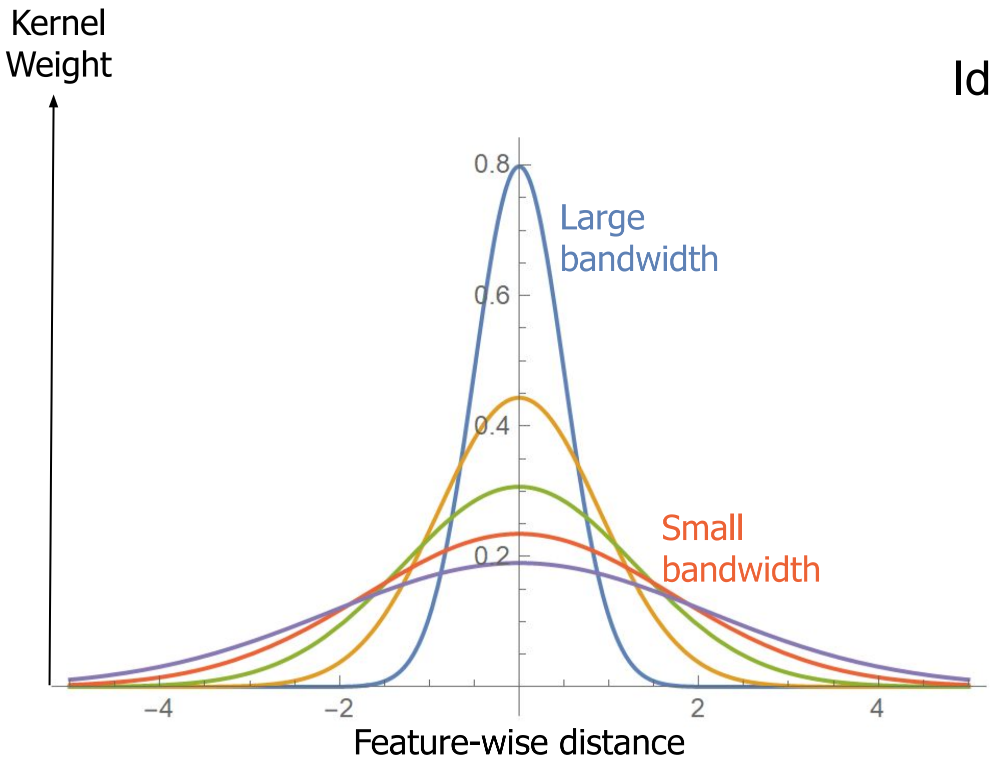
- Gather neighbor local feature based on learned weight
- Learnable kernel bandwidth (receptive field) for smoothing
- Estimate self-confidence to reduce burstiness



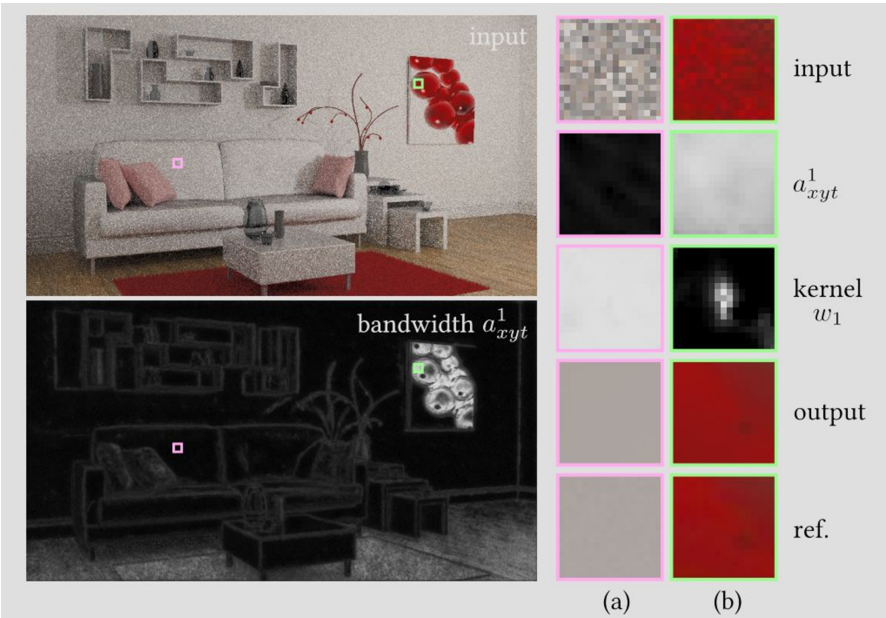


Method 2.1 Learnable Bandwidth

- Gaussian kernel with learnable bandwidth
  - Larger bandwidth (Narrow): Spatially non-correlated info.
  - Smaller bandwidth (Wide): Spatially correlated info.



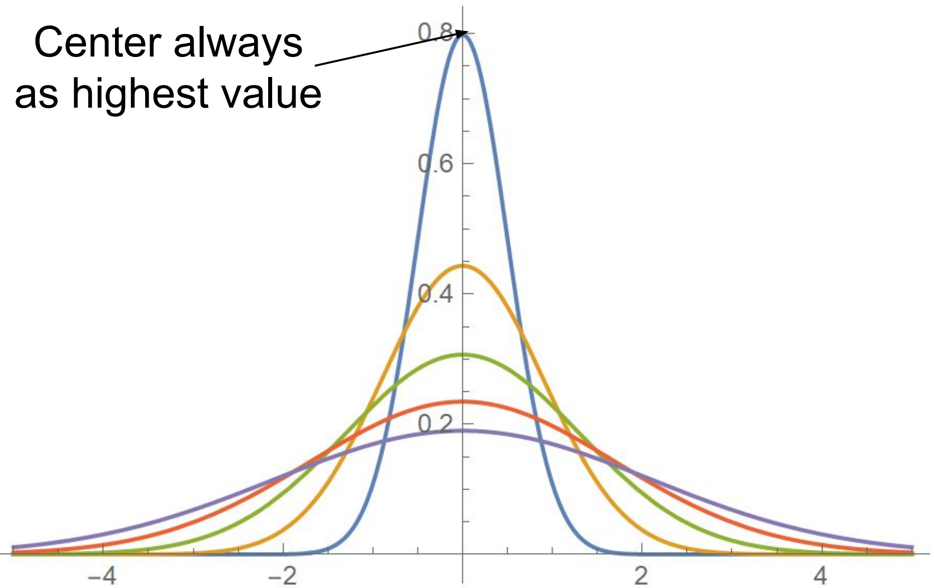
Identify high-frequency region for denoising



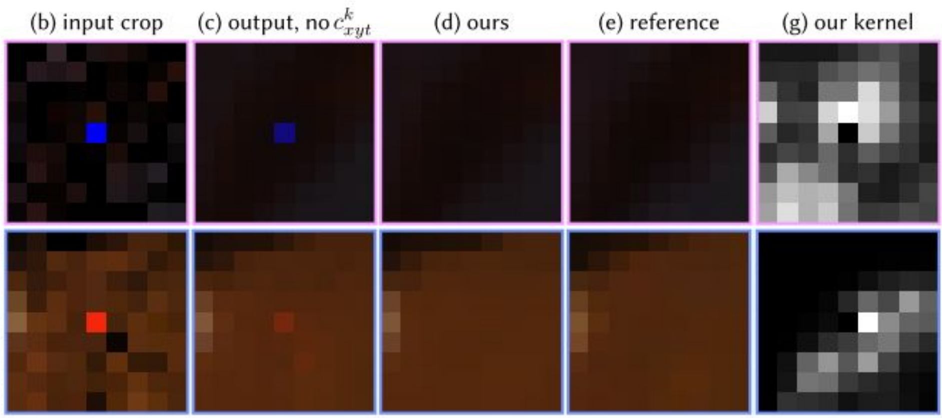
Method 2.2 Learnable Self-Confidence

- Gaussian filter always show highest value on center
- Center (itself) may contain irrelevant feature
- Self-confidence allows to reject itself when it is not relevant for image retrieval

Center always as highest value



Helps to reject when the center pixel is an outlier for denoising



KAIST

Kernel Visualization

- Refines local features to highlight all important region

Local feature attention

Average Pooling

Learnable Pooling

Kernel Visualization

- Refines local features to highlight all important region

Local feature attention

Average Pooling

Learnable Pooling

Kernel Visualization

- Refines local features to highlight all important region

Local feature attention

Average Pooling

Learnable Pooling

KAIST

Kernel Visualization

- Self-confidence highlights less highlighted region
- Large bandwidth for edge details

Local feature attention

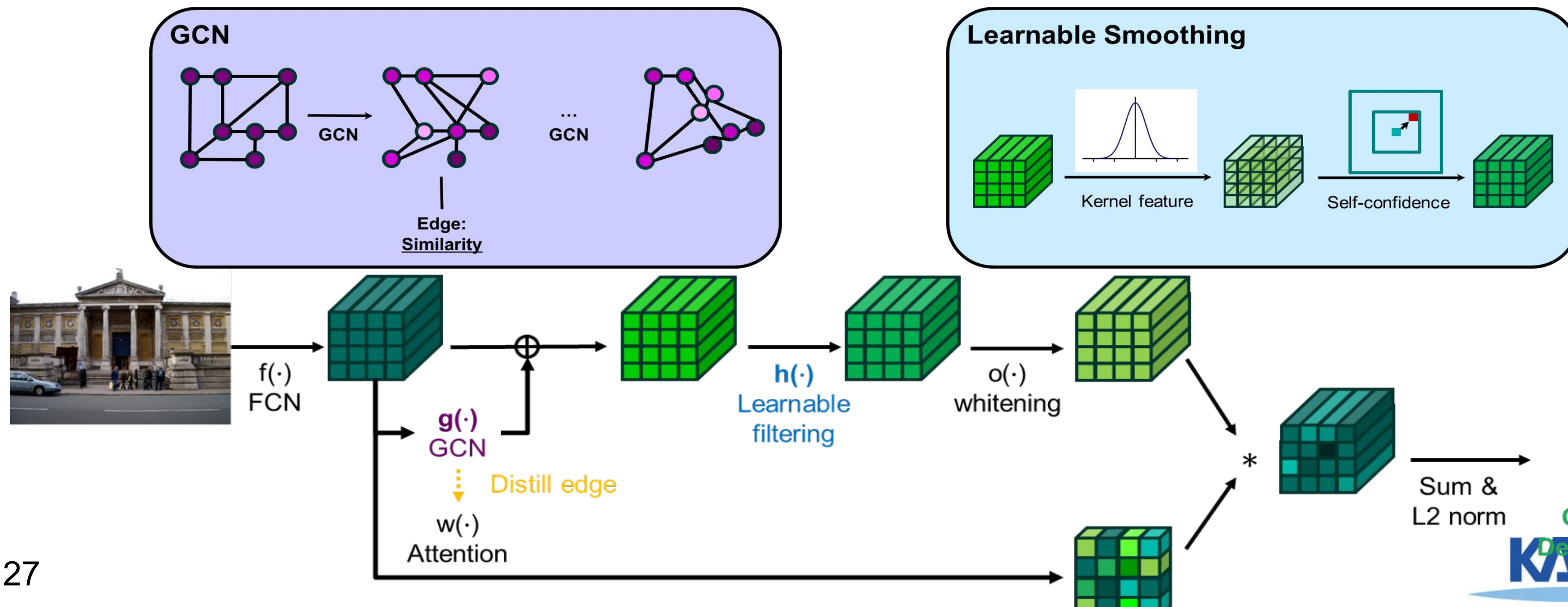
Self-confidence

Bandwidth

Learnable Pooling

Experiment Details

- Use ImageNet pretrained ResNet-18 as backbone
- Finetune backbone with small learning rate, while training GCN and smoothing with large learning rate on SfM dataset
    - 1:10 ratio



Numerical Results

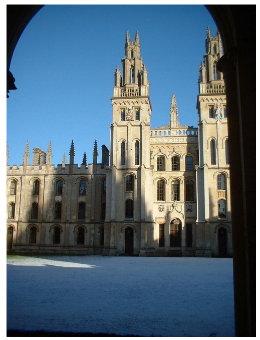
- Slight increase in performance
- Could not fully merge the advantages of both methods at the end

Method	SFM_val	R_Oxford		R_Paris	
		M	H	M	H
HOW	85.2	74.2	52.1	80.0	59.3
+ Learnable filter	85.1	75.0	51.8	**80.9**	**61.3**
+ GCN	**86.1**	**75.3**	**53.3**	80.7	60.9
+ Both	84.9	**75.3**	53.2	80.5	60.2

Visual Results

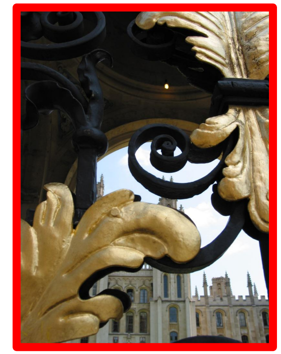
HOW

Rank #1



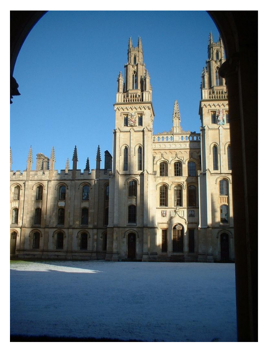
Rank #2

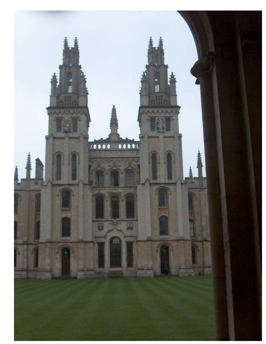
Rank #3



Rank #4

Ours

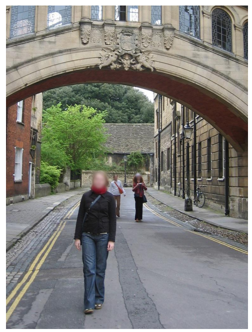




Visual Results

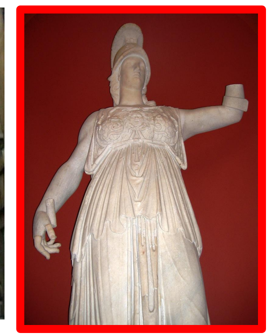
HOW

Rank #1



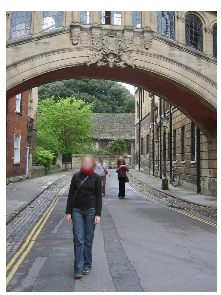
Rank #2

Rank #6



Rank #7

Ours



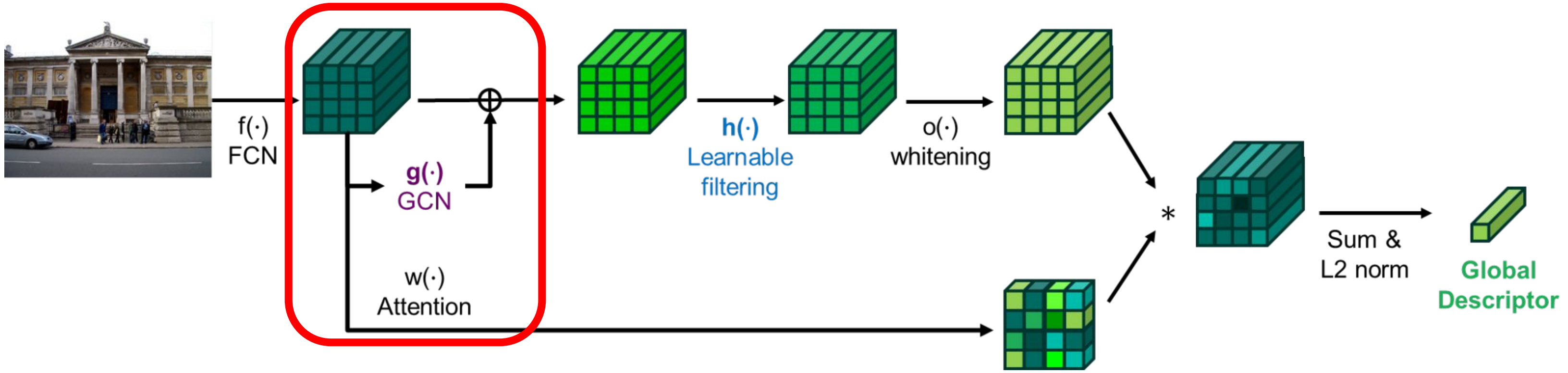
Visual Results

HOW

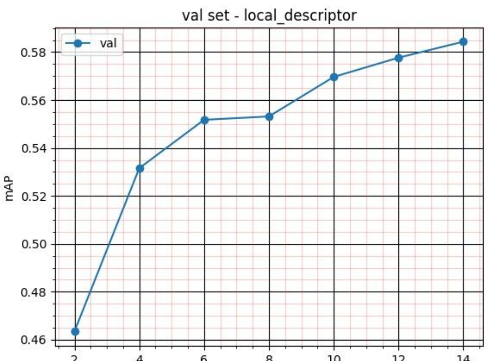
Ours

Limitation

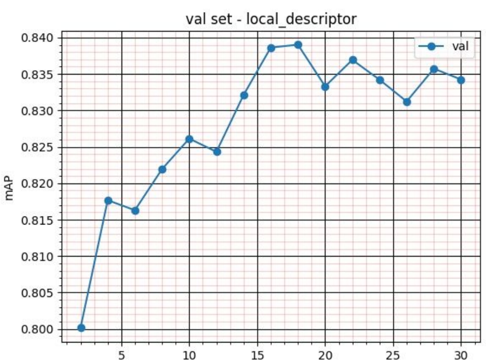
- Hard to compare with general method (i.e., self attention)
  - Could not converge the training with self attention
- Could not apply our work on various backbones



GCN could be generalized to self-attention, but find it hard to optimize



Val. plot w/ self attention



Val. plot w/ GCN

Summary

- Provide local context to the local descriptors for better matching
  - Graph-based Refinement
  - Learnable Smoothing
- Found substantial increase in performance for retrieval
- Shown that our method could be also used for better localization

<Contributions>
KB: Local descriptor matching baseline + Learnable Filter
JH: Graph-base Refinement + Localization + Visualizations

Appx. Kernel Size

- Smaller kernel gives better performance
- Extensive smoothing on local features may harm the performance

Filter size	SFM_val	R_Oxford		R_Paris	
		M	H	M	H
3	**85.1**	**75.0**	**51.8**	**80.9**	**61.3**
5	83.9	74.0	51.0	79.8	59.0
7	84.0	73.4	50.9	80.0	59.1