CVPR

SuperGlue: Learning Feature Matching with Graph Neural Networks

Paul-Edouard Sarlin[1]
Daniel DeTone[2]
Tomasz Malisiewicz[2]
Andrew Rabinovich[2]

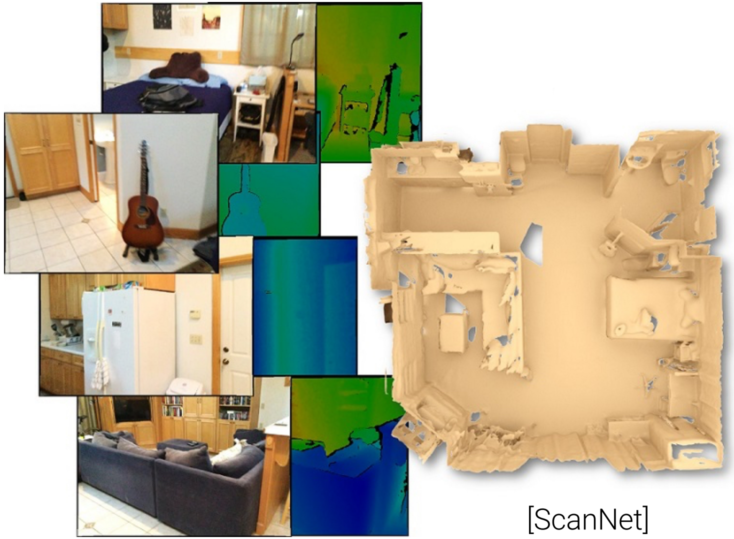
ETH zürich
magic leap

Feature matching is ubiquitous

- 3D reconstruction
- Visual localization
- SLAM
- Place recognition

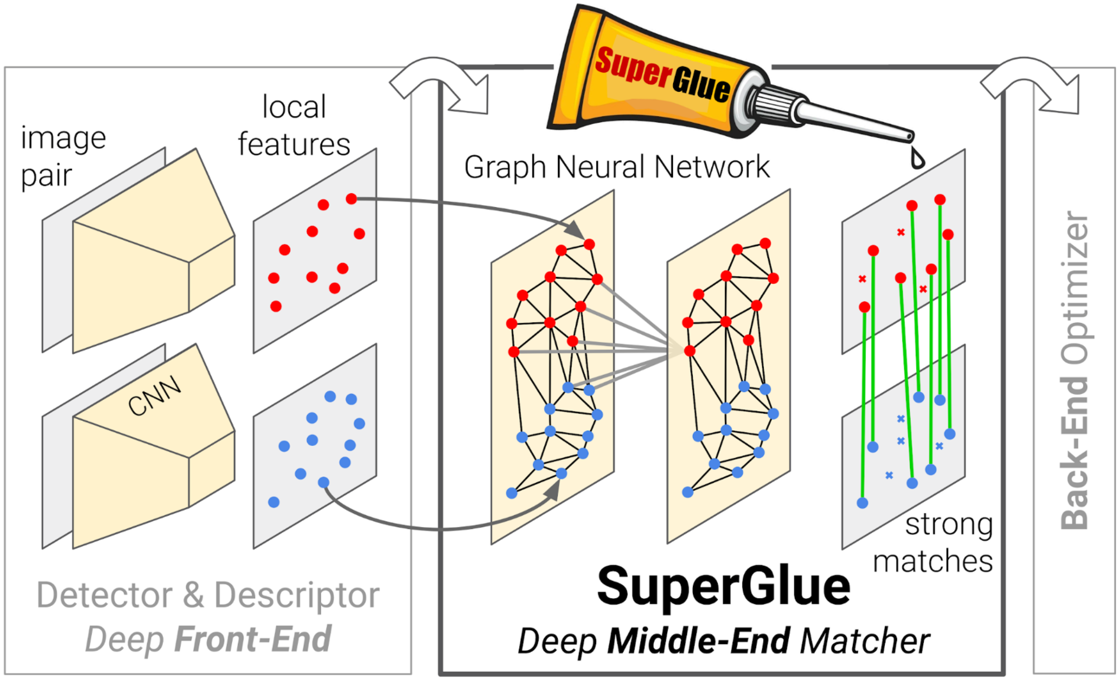
[Image Matching Workshop 2020]

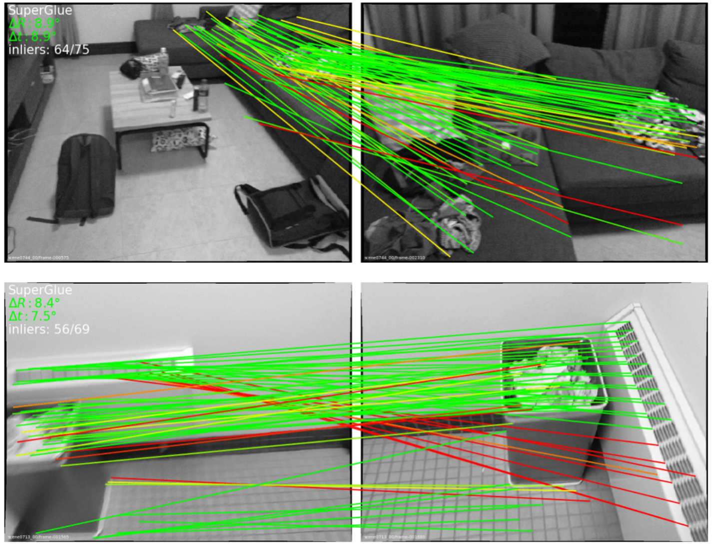
[Google VPS]



[ScanNet]

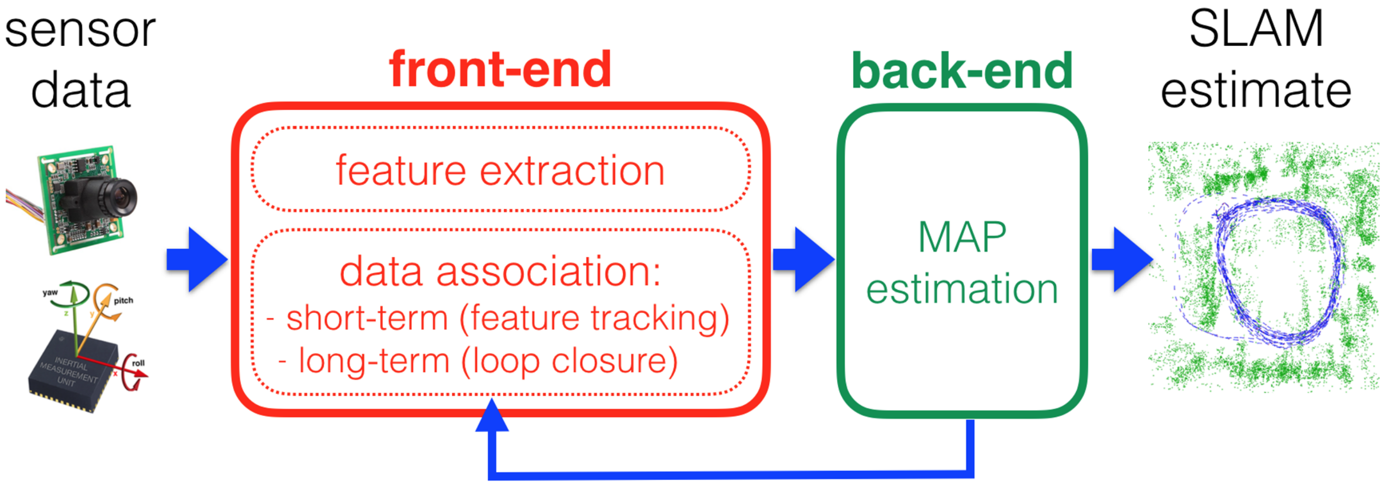
SuperGlue = Graph Neural Nets + Optimal Transport





- Extreme **wide-baseline** image pairs in **real-time on GPU**
- State-of-the-art **indoor+outdoor** matching with **SIFT** & **SuperPoint**

Visual SLAM

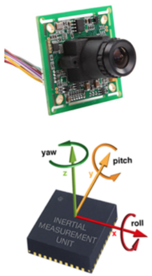


- **Front-end**: images to constraints
  - Recent works: **deep learning for feature extraction** → Convolutional Nets!
- **Back-end**: optimize pose and 3D structure

[Cadena et al, 2016]

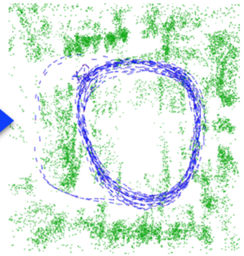
A middle-end

sensor data



front-end: feature extraction → **middle-end**: data association → **back-end**: MAP estimation →

SLAM estimate



- Our position: **learn** the data association!
- We propose a new **middle-end**: **SuperGlue**
- 2D-to-2D feature matching

A minimal matching pipeline

SuperGlue: context aggregation + matching + filtering

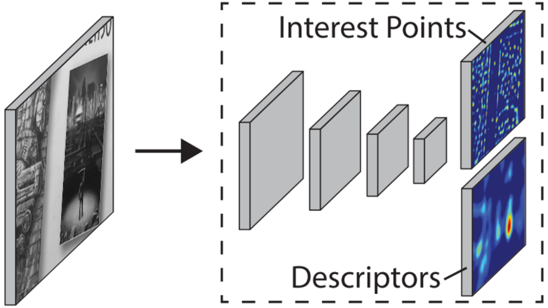
image pair

detection → description → feature matching → outlier filtering → pose estimation

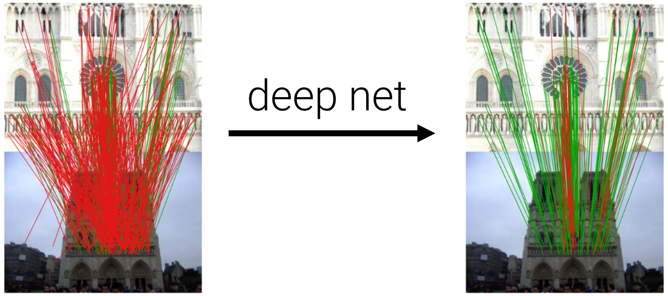
> Classical: SIFT, ORB
> Learned: SuperPoint, D2-Net

Nearest Neighbor Matching

> Heuristics: ratio test, mutual check
> Learned: classifier on set



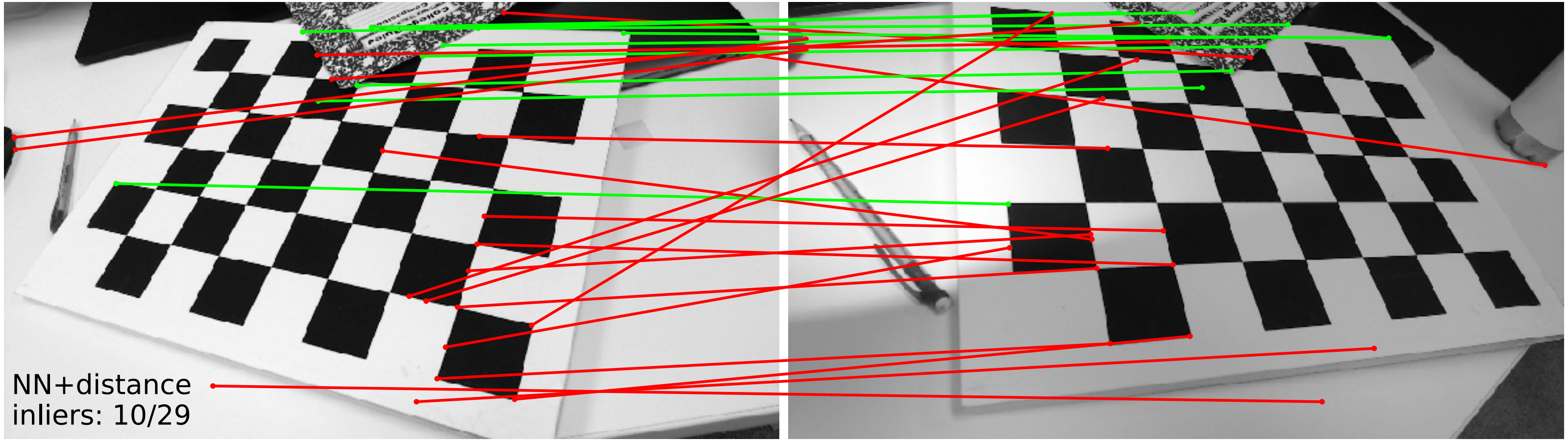
[DeTone et al, 2018]



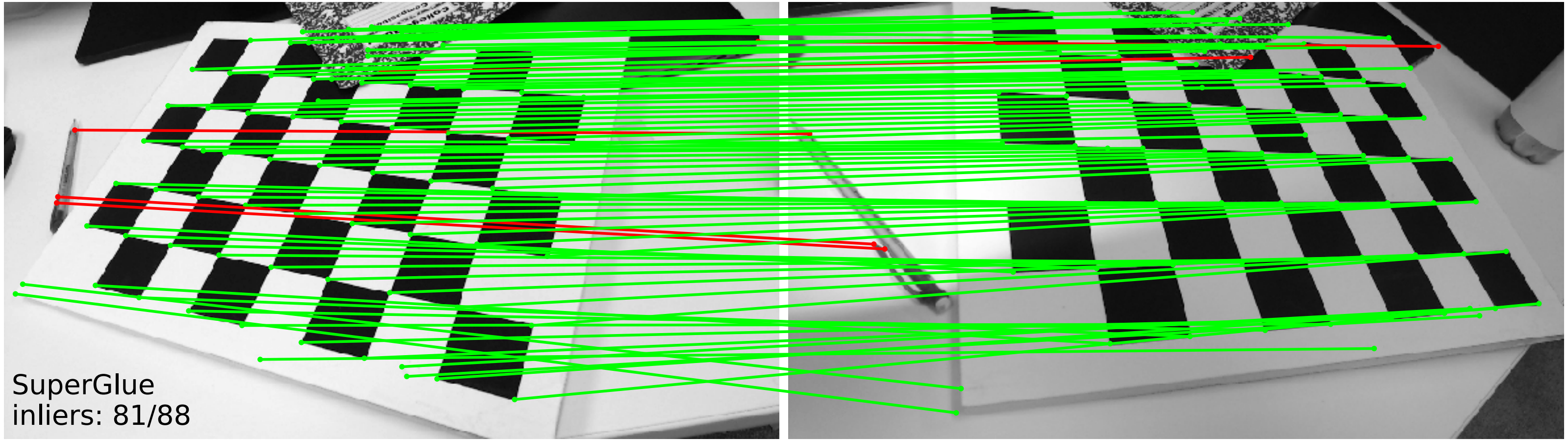
[Yi et al, 2018]

The importance of context

no
SuperGlue



**with
SuperGlue**

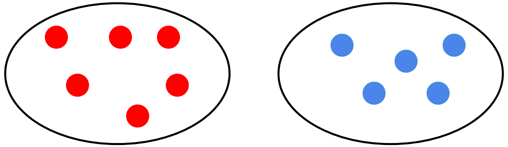


Problem formulation

Inputs → 

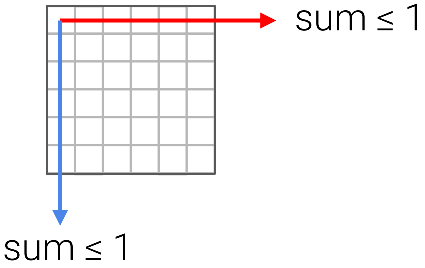
→ **Outputs**

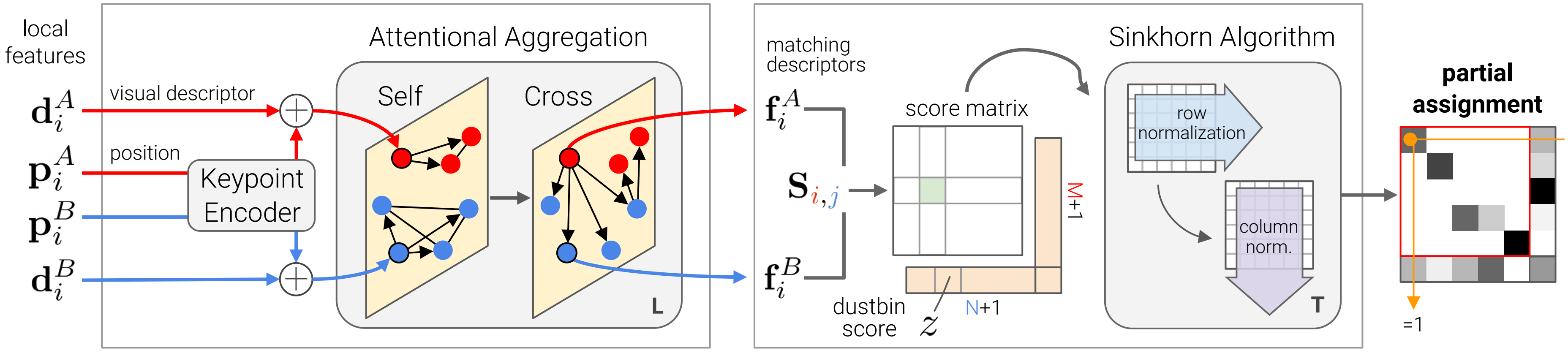
- Images A and B
- **2 sets** of **M**, **N** **local features**
  - Keypoints: $\mathbf{p}_i := (x, y, c)_i$
    - Coordinates (x, y)
    - Confidence c
  - Visual descriptors: $\mathbf{d}_i$



Single a match per keypoint
+ occlusion and noise
→ a **soft partial assignment**:

$$\mathbf{P} \in [0, 1]^{M \times N}$$





A Graph Neural Network with attention

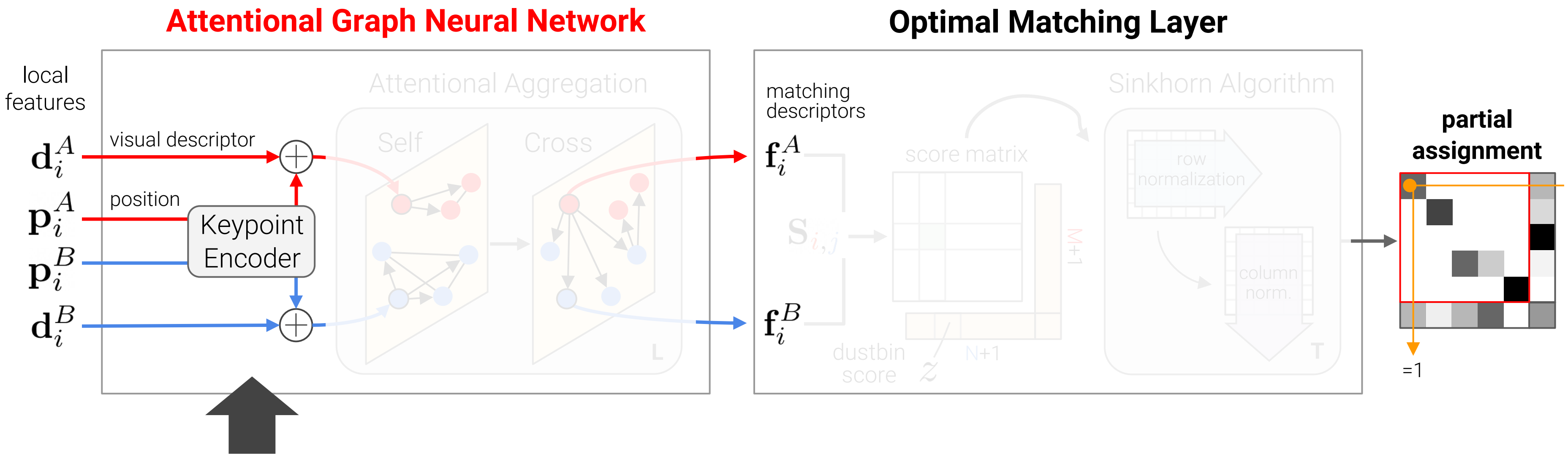
Solving a partial assignment problem

Encodes **contextual cues** & priors

Reasons about the 3D scene

Differentiable **solver**

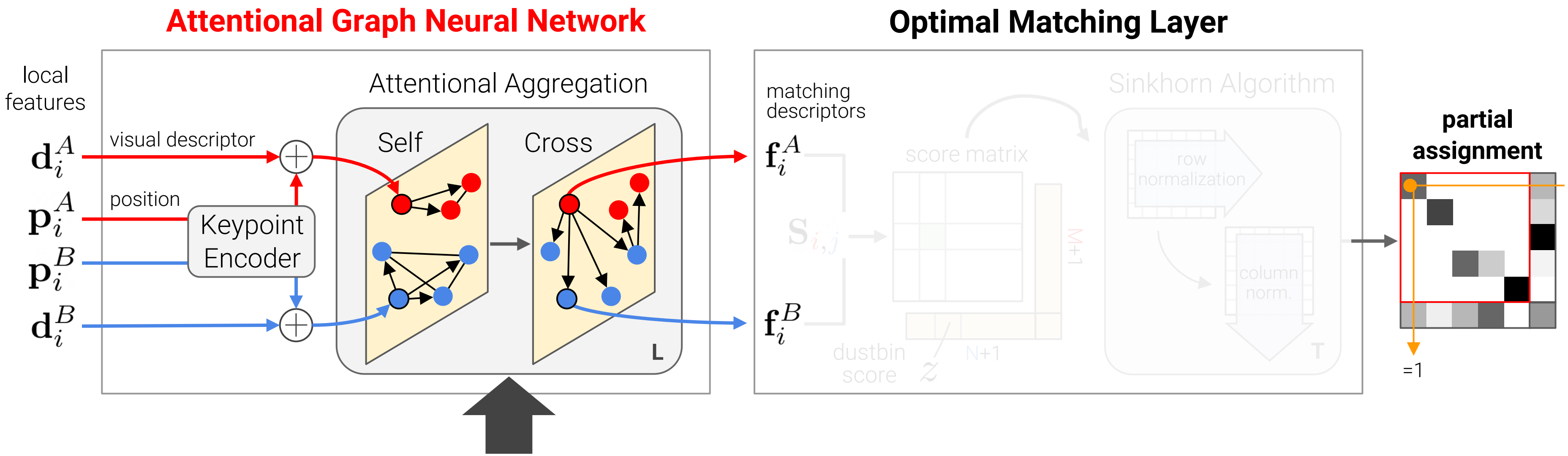
Enforces the assignment constraints = **domain knowledge**



- Initial representation for each keypoints i : ${}^{(0)}\mathbf{x}_i$
- Combines visual appearance and position with an MLP:

$$ {}^{(0)}\mathbf{x}_i = \mathbf{d}_i + \mathrm{MLP}\left(\mathbf{p}_i\right) $$

Multi-Layer Perceptron

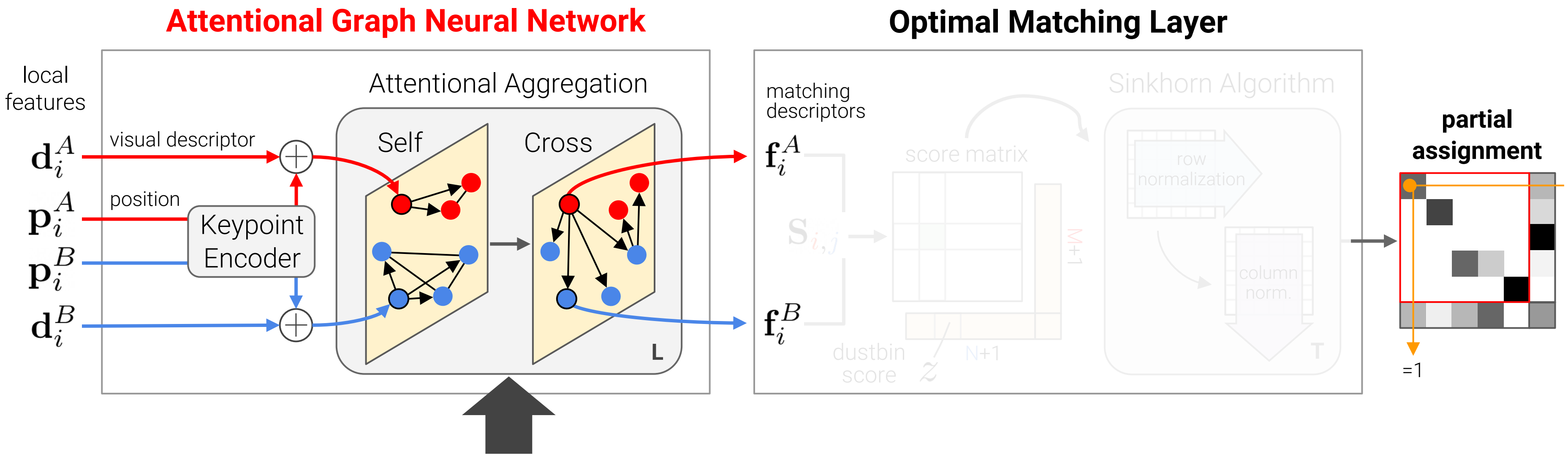


Update the representation based on other keypoints:

- in the same image: "**self**" edges
- in the other image: "**cross**" edges

$$ {}^{(\ell)}\mathbf{x}_i^A \longrightarrow {}^{(\ell+1)}\mathbf{x}_i^A $$

→ A complete **graph** with two types of edges



Update the representation using a **Message Passing Neural Network**

$$^{(\ell+1)}\mathbf{x}_i^A = {}^{(\ell)}\mathbf{x}_i^A + \mathrm{MLP}\left(\left[{}^{(\ell)}\mathbf{x}_i^A \,||\, \mathbf{m}_{\mathcal{E}\rightarrow i}\right]\right)$$

the message

Attentional Aggregation

- Compute the **message** $\mathbf{m}_{\mathcal{E}\to i}$ using **self** and **cross attention**
- Soft database retrieval: query $\mathbf{q}_i$, key $\mathbf{k}_j$, and value $\mathbf{v}_j$

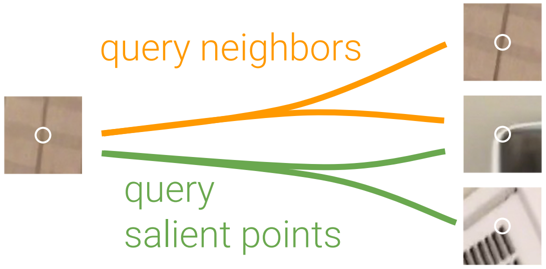
$$\mathbf{m}_{\mathcal{E}\to i} = \sum_{j:(i,j)\in\mathcal{E}} \alpha_{ij}\mathbf{v}_j$$

$$\alpha_{ij} = \mathrm{Softmax}_j\left(\mathbf{q}_i^\top \mathbf{k}_j\right)$$

$$\mathbf{q}_i = \mathbf{W}_1\,{}^{(\ell)}\mathbf{x}_i + \mathbf{b}_1$$

$$\begin{bmatrix}\mathbf{k}_j\\ \mathbf{v}_j\end{bmatrix} = \begin{bmatrix}\mathbf{W}_2\\ \mathbf{W}_3\end{bmatrix}{}^{(\ell)}\mathbf{x}_j + \begin{bmatrix}\mathbf{b}_2\\ \mathbf{b}_3\end{bmatrix}$$

$\mathbf{x}_i$ = [tile, position (70, 100)]



= [tile, pos. (80, 110)]

= [corner, pos. (60, 90)]

= [grid, pos. (400, 600)]

[Vaswani et al, 2017]

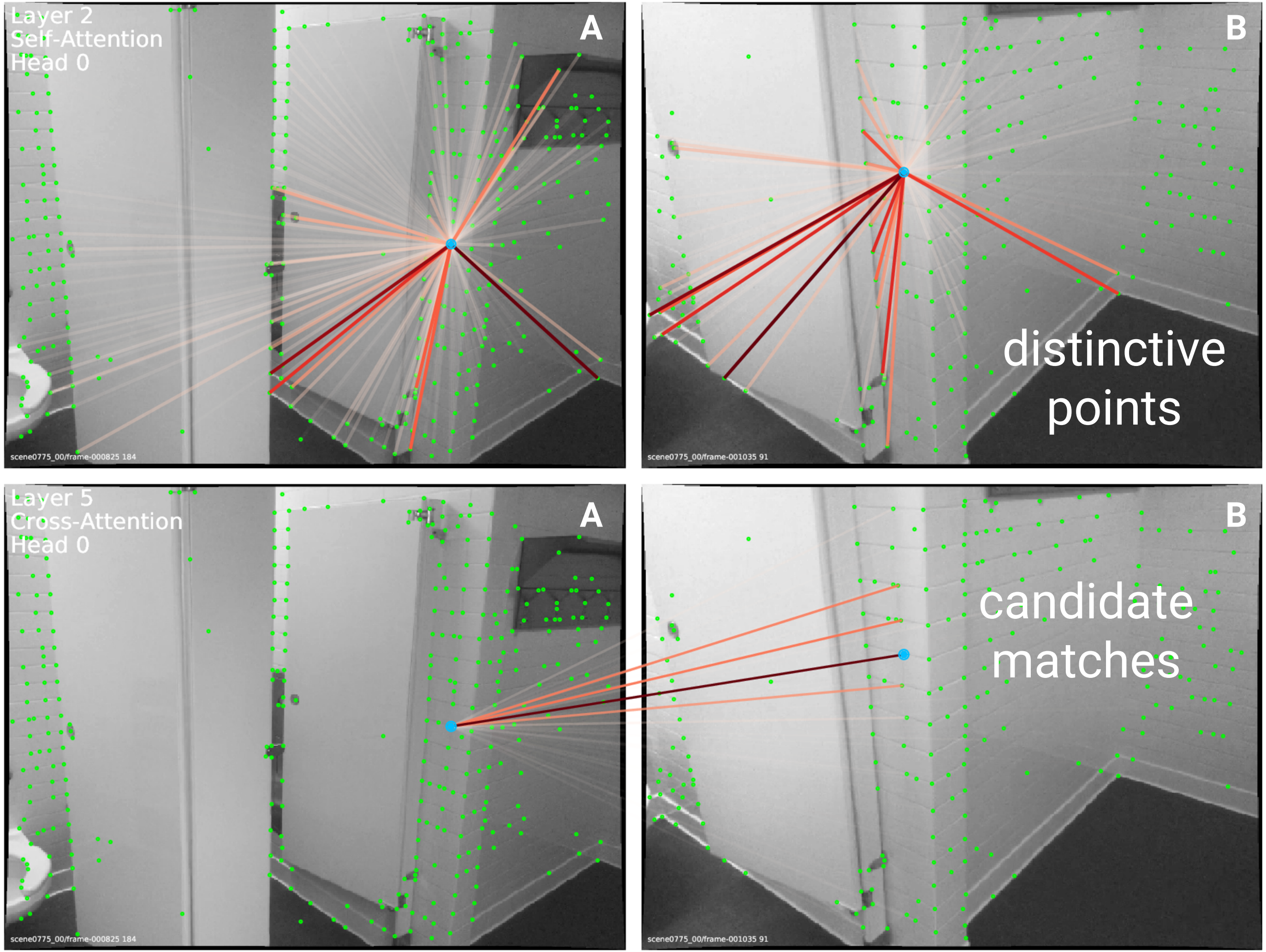
Self-attention

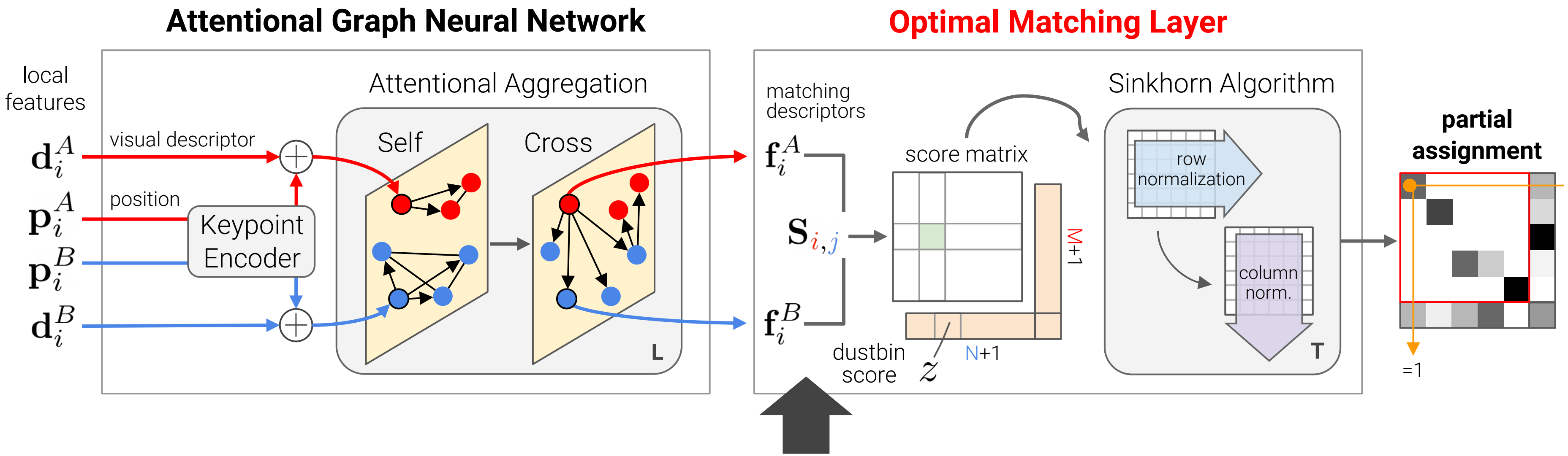
= intra-image information flow

Cross-attention

= inter-image

Attention builds a **soft**, **dynamic**, **sparse graph**

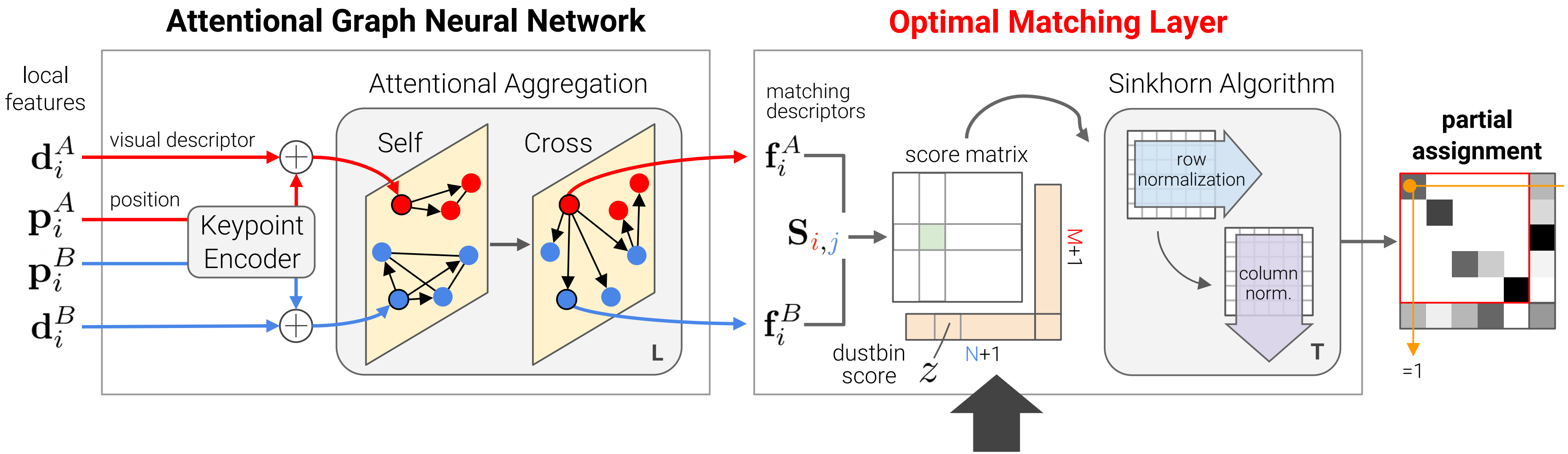




Compute a **score matrix** $\mathbf{S} \in \mathbb{R}^{M \times N}$ for all matches:

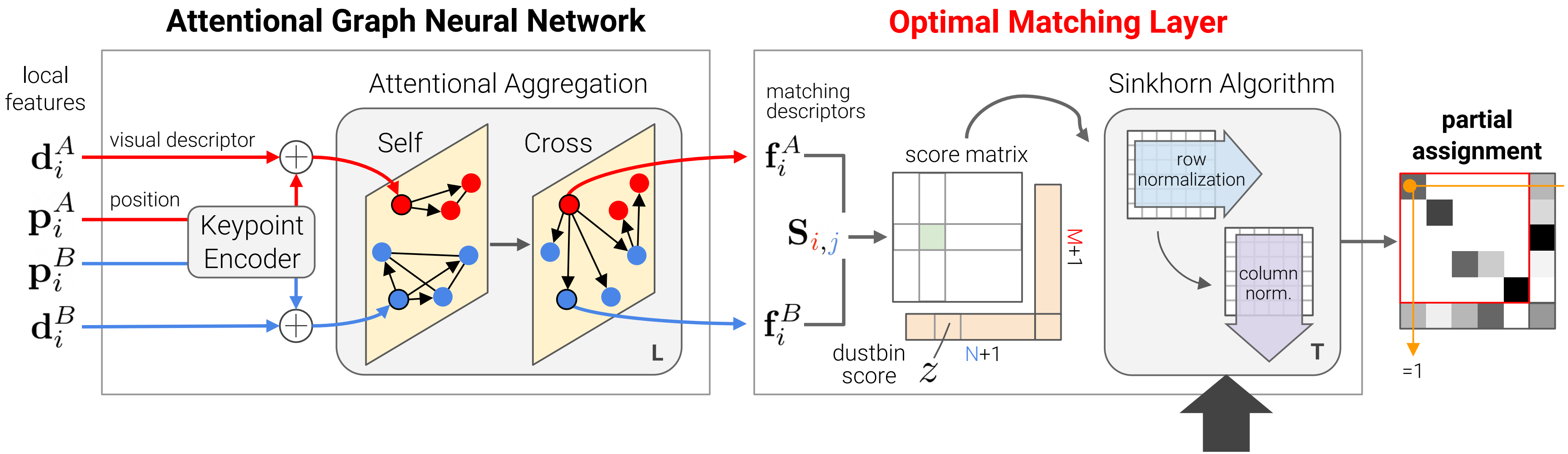
$$\mathbf{f}_i^A = \mathbf{W} \cdot {}^{(L)}\mathbf{x}_i^A + \mathbf{b}$$

$$\mathbf{S}_{i,j} = < \mathbf{f}_i^A, \mathbf{f}_j^B >$$



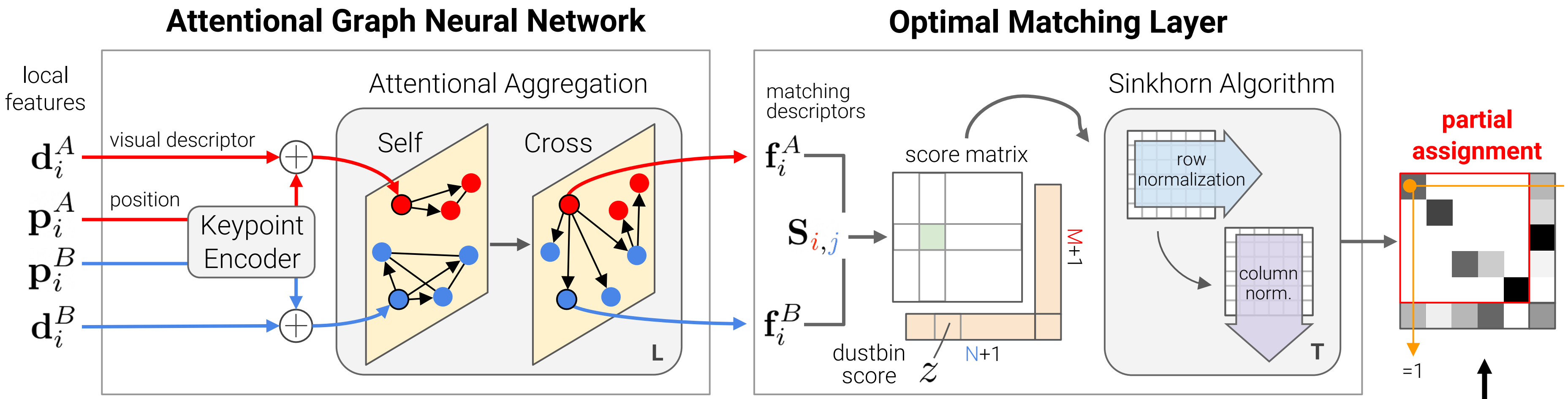
- Occlusion and noise: unmatched keypoints are assigned to a **dustbin**
- **Augment** the scores with a learnable dustbin score z

$$\bar{\mathbf{S}}_{i,N+1} = \bar{\mathbf{S}}_{M+1,j} = \bar{\mathbf{S}}_{M+1,N+1} = z \in \mathbb{R}$$



- Compute the assignment $\bar{\mathbf{P}}$ that maximizes $\sum_{i,j} \bar{\mathbf{S}}_{i,j} \bar{\mathbf{P}}_{i,j}$
- Solve an **optimal transport** problem
- With the **Sinkhorn algorithm**: differentiable & soft Hungarian algorithm

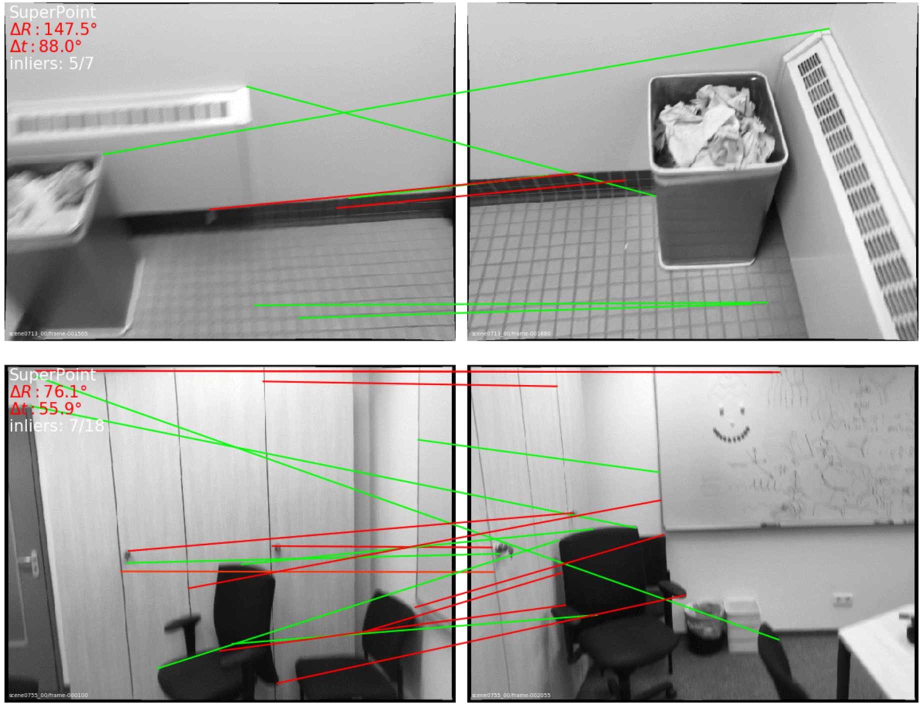
[Sinkhorn & Knopp, 1967]



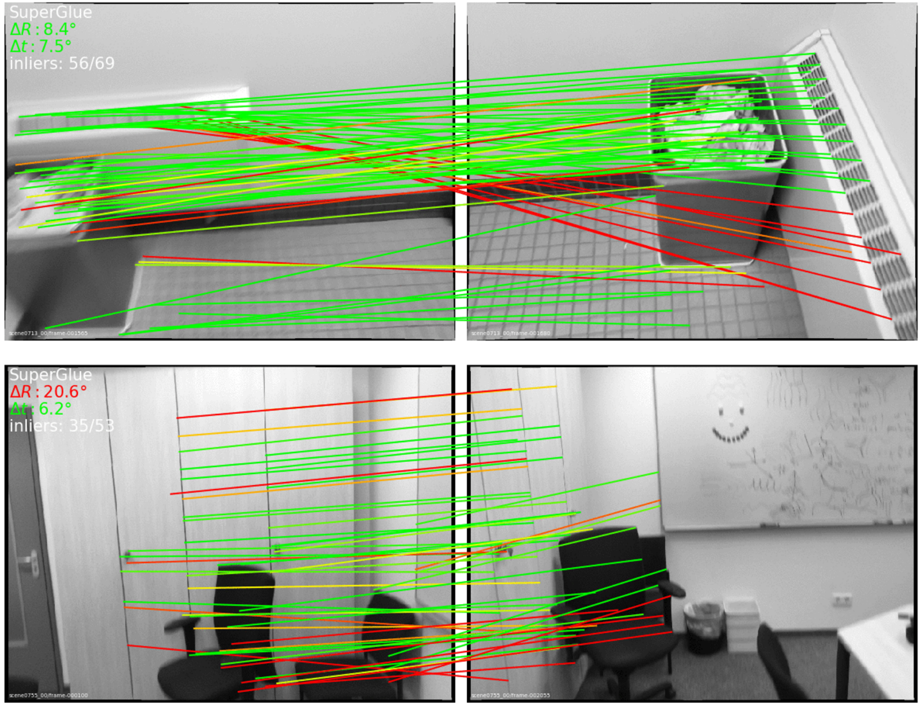
- Compute **ground truth correspondences** from pose and depth
- Find which keypoints should be **unmatched**
- Loss: maximize the log-likelihood $\bar{\mathbf{P}}_{i,j}$ of the GT cells

Results: indoor - ScanNet

SuperPoint + NN + heuristics



SuperPoint + **SuperGlue**

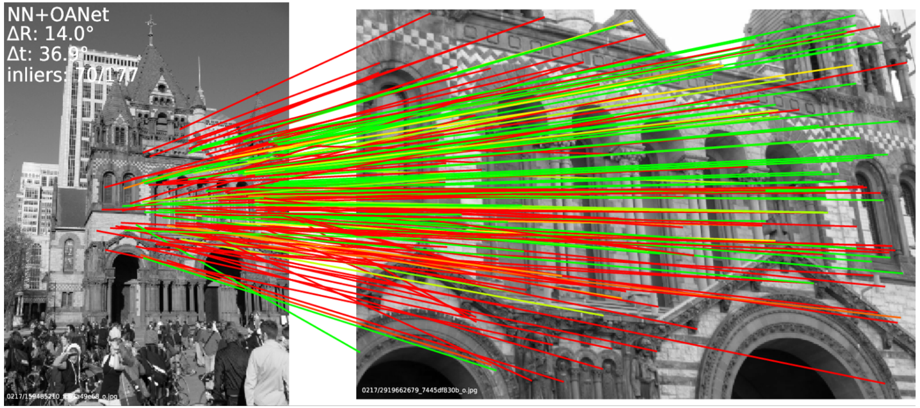


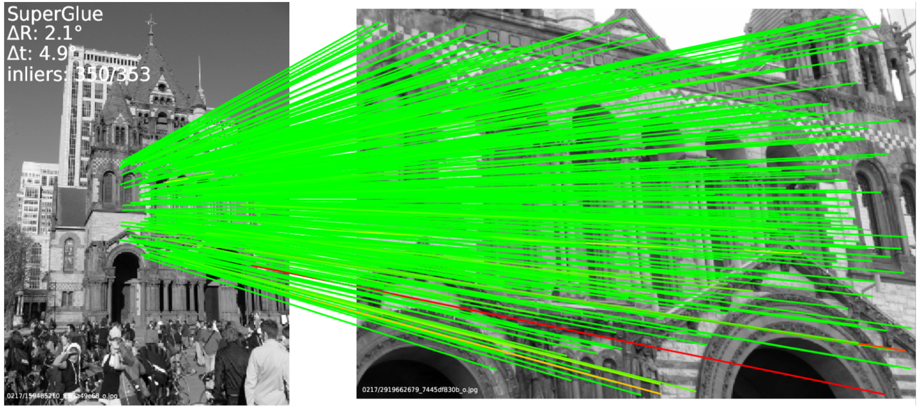
SuperGlue: more **correct matches** and fewer **mismatches**

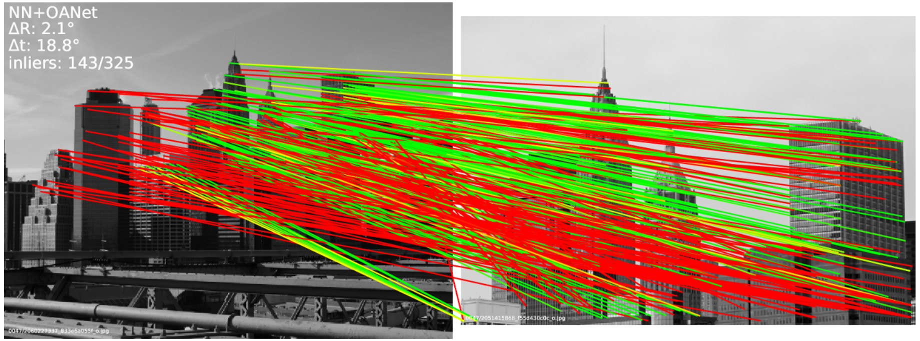
Results: outdoor - SfM

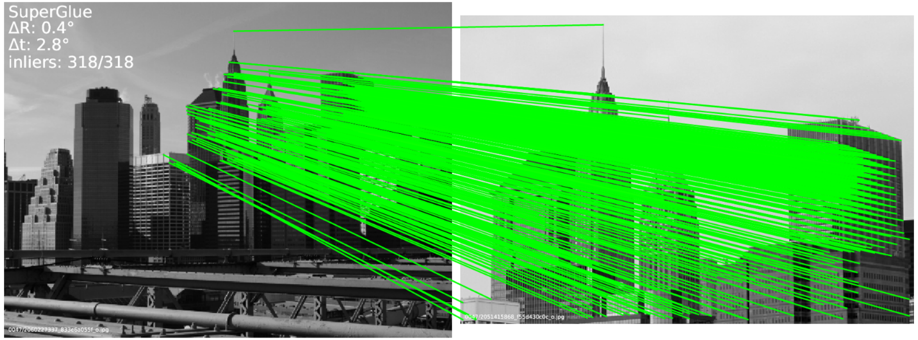
SuperPoint + NN + OA-Net (inlier classifier)

SuperPoint + **SuperGlue**



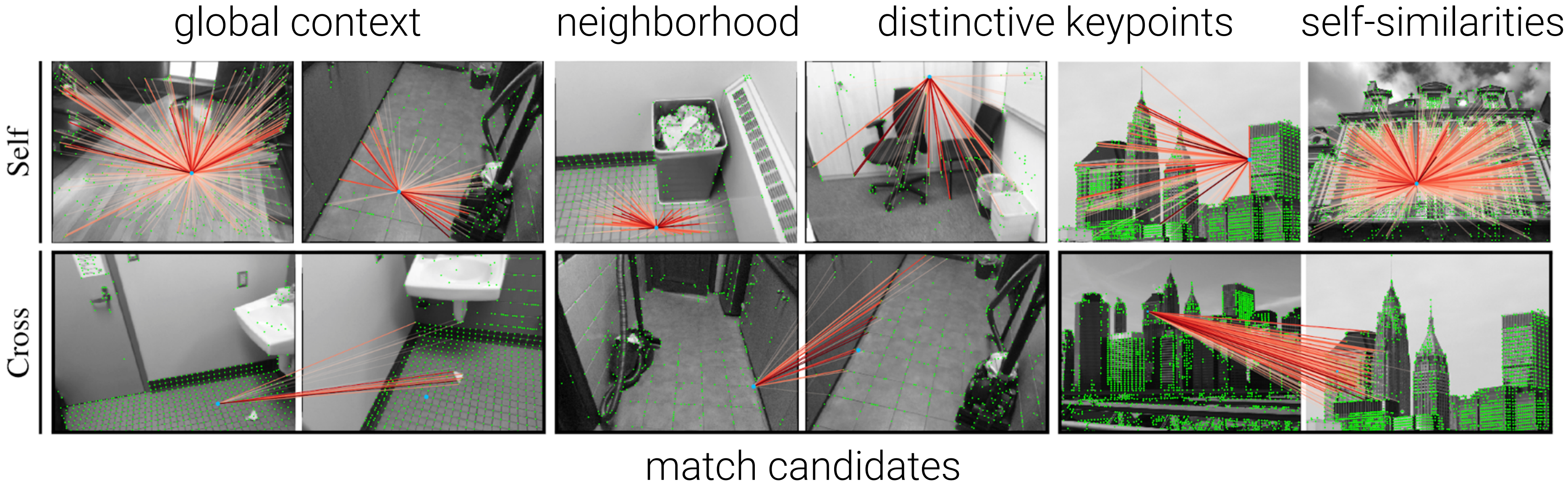






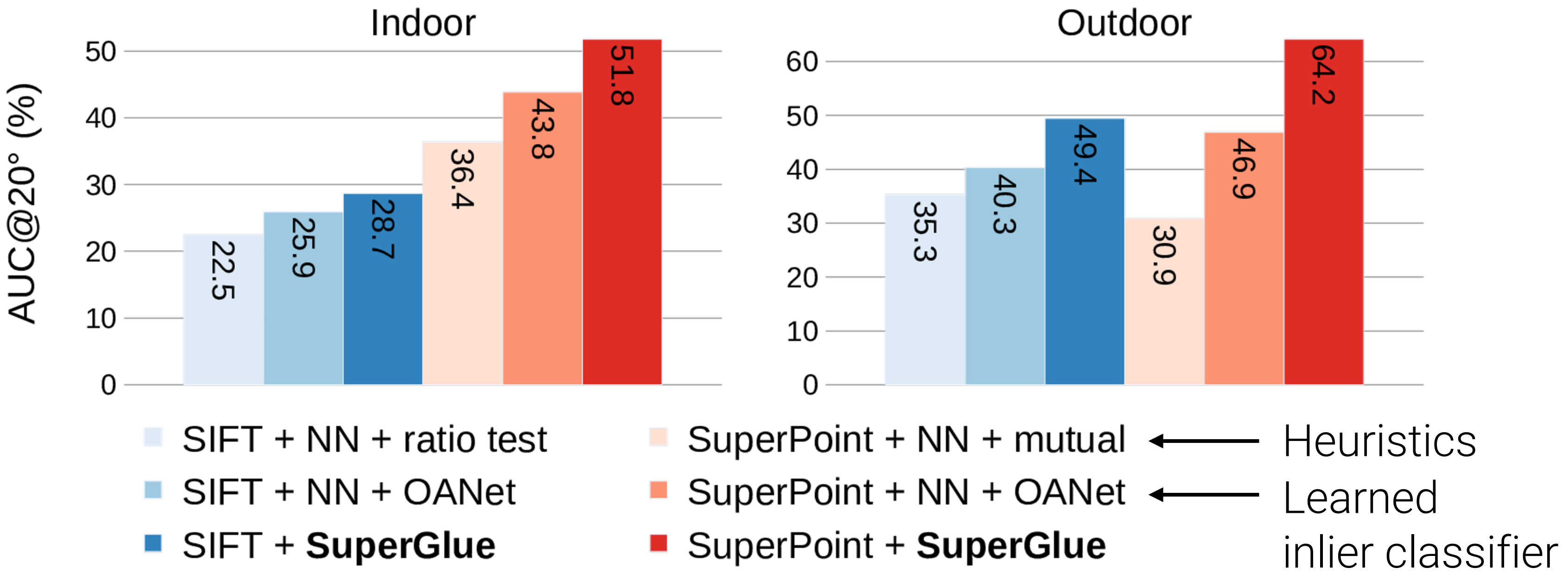
SuperGlue: more **correct matches** and fewer **mismatches**

Results: attention patterns



Flexibility of attention → **diversity of patterns**

Evaluation

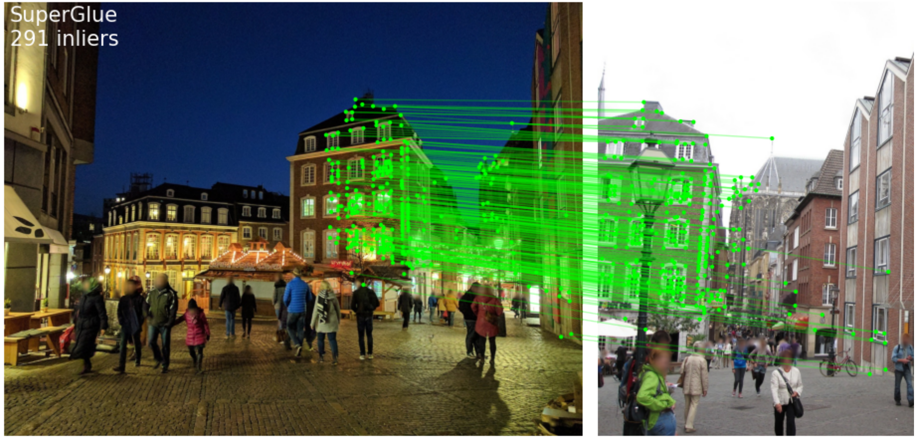


SuperGlue yields **large improvements** in all cases

SuperGlue @ CVPR 2020

First place in the following competitions:

- Image matching challenge — vision.uvic.ca/image-matching-challenge
- Local features for visual localization — www.visuallocalization.net
- Visual localization for handheld devices



SuperGlue

Learning Feature Matching with Graph Neural Networks

A major step towards end-to-end **deep SLAM & SfM**

psarlin.com/superglue