Image Feature Matching: SuperPoint and SuperGlue

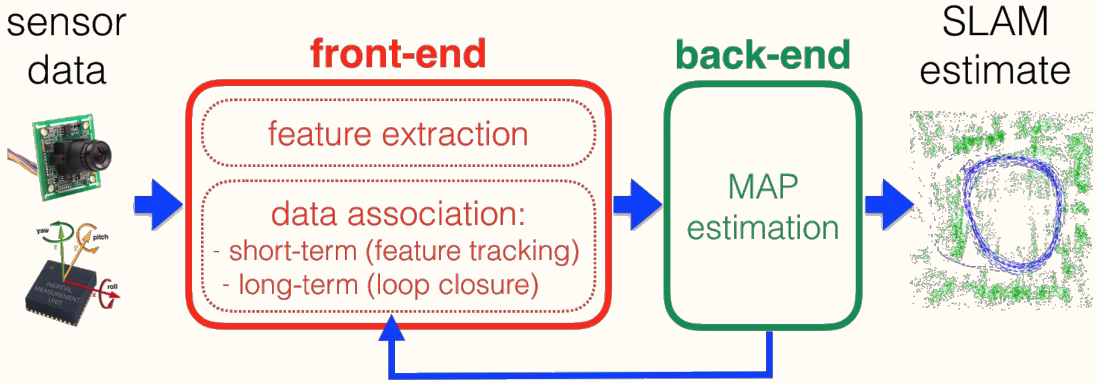
IFT 6757 - Duckietown

Acknowledgements

- This presentation uses material presented in the two papers:
  - SuperPoint: Self-Supervised Interest Point Detection and Description
  - SuperGlue: Learning Feature Matching with Graph Neural Networks
- This presentation also borrows from two presentations:
  - Deep Visual SLAM Frontends by Tomasz Malisiewicz at CVPR 2020
  - SuperGlue presentation by Paul-Edouard Sarlin at CVPR 2020
- All images used are from the above-mentioned material, unless noted otherwise.

Motivation

- Two parts of Visual SLAM
- Front-end
  - Feature extraction (SIFT, SURF, ORB)
  - Data Association (feature tracking, loop closure)
- Back-end
  - Optimize pose and 3D structure

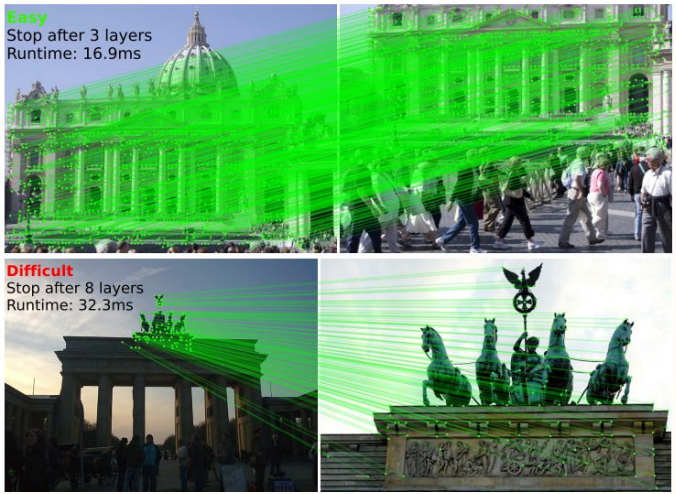


Source: C. Cadena, L. Carlone, et al. “Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age”

Motivation

Why is it so difficult?

- Appearance changes: illumination, weather, sensor noise
- Viewpoint changes: perspective distortion, scale, occlusion
- Repetitive structures: many false matches (windows, tiles, trees)
- Textureless regions: few or no features (white walls, roads)



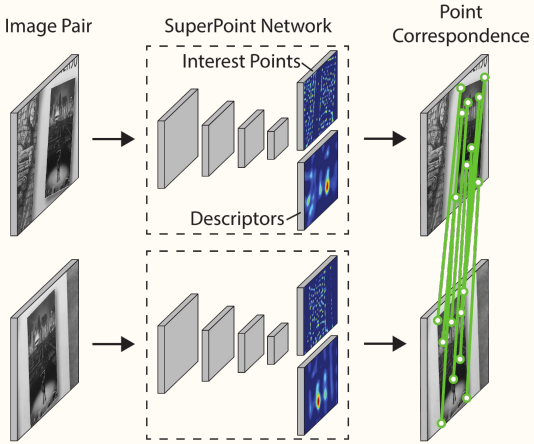
Source: P. Lindenberger, P. Sarlin, et al. “LightGlue: Local Feature Matching at Light Speed”

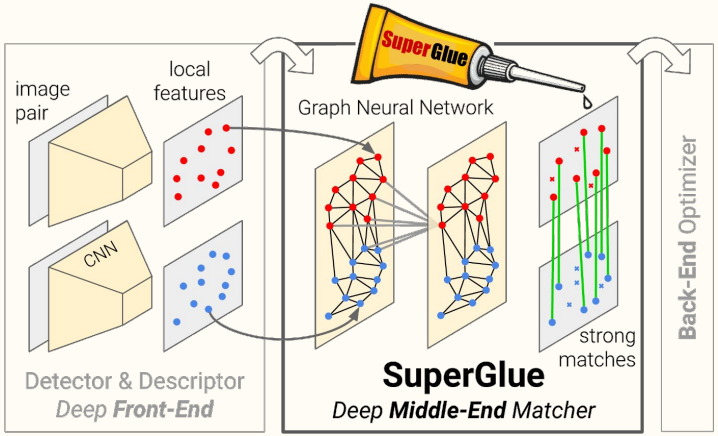
Why is it so difficult?

Source: J. Wang, N. Karaev, et al. “Visual Geometry Grounded Deep Structure From Motion”

Research Questions

- SuperPoint
  - How to learn detectors & descriptors?
  - Can we train without labeled correspondences (self-supervised)?
  - How does it compare with earlier pipelines (e.g., LIFT, SIFT)?
- SuperGlue
  - How to learn the data association between two images?





Preliminaries

- Learning-based Approaches
  - Deep Learning & Neural Networks
  - Convolutional Neural Networks (CNNs)
  - Graph Neural Networks (GNNs)
- Matching with Heuristics
  - Nearest Neighbor
  - Ratio Test
  - Mutual Consistency Test
- Geometric Verification
  - Homography & Pose Estimation
  - Direct Linear Transform (DLT)
  - Essential & Fundamental Matrices
  - 5-point / 8-point Algorithms
  - Random Sample Consensus (RANSAC)
- Optimal Assignment Methods
  - Bipartite Matching
  - Hungarian Algorithm
  - Sinkhorn Iteration

Evaluation

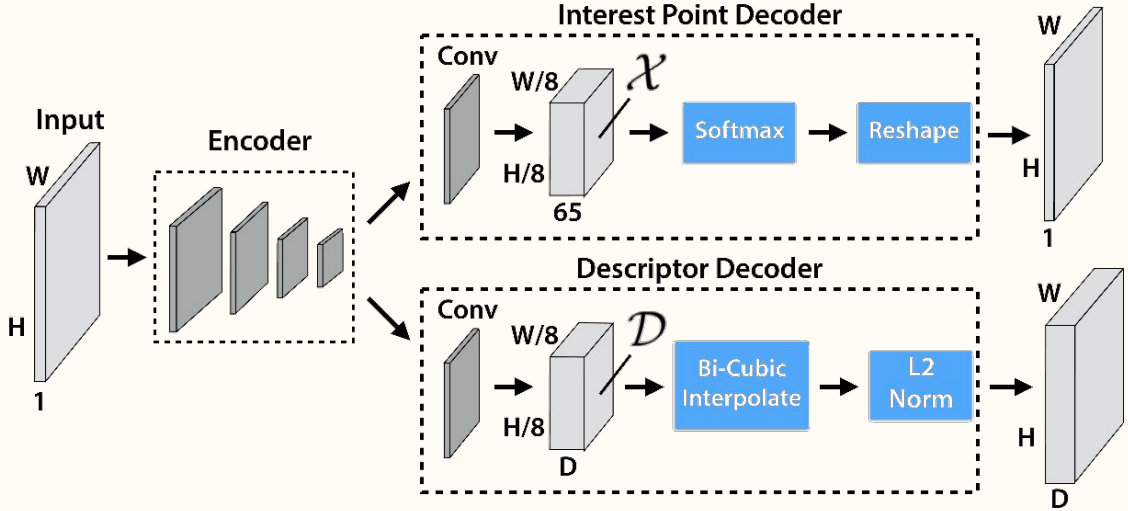
- Detector Metrics:
  - Corner Detection Average Precision (AP) - Measures how well detected points align with ground-truth corners (Precision–Recall AUC). Higher AP = better.
  - Localization Error (LE) - Average pixel distance between detected points and their closest ground-truth corner. Lower is better.
  - Repeatability (Rep) - Probability a point is re-detected in a second view of the same scene. Higher is better.
- Descriptor and Matching Metrics
  - Nearest Neighbor mAP - Discriminativeness of descriptors via NN matching (Precision–Recall AUC). Higher = better descriptors.
  - Matching Score (MS) - Fraction of ground-truth correspondences recovered by the pipeline. Combines detection + description.
  - Homography Estimation Accuracy - Ability to recover the ground-truth homography by transforming image corners correctly.
  - Pose Estimation mAP - Area under the curve of relative pose error.

SuperPoint

The art and craft of designing neural nets to replace SIFT.

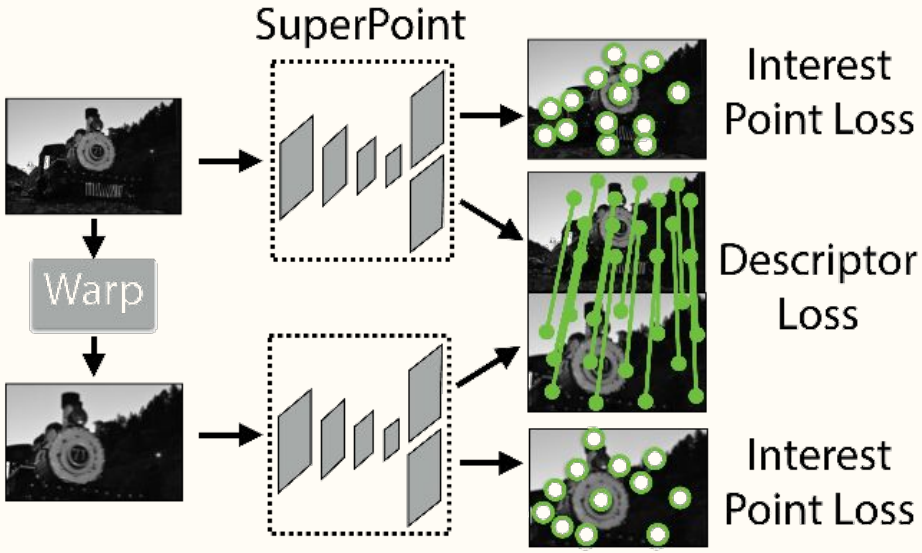
Model Architecture

- CNN architecture
  - VGG-like backbone
- Points + descriptors computed jointly, no patches
- No deconvolution layer



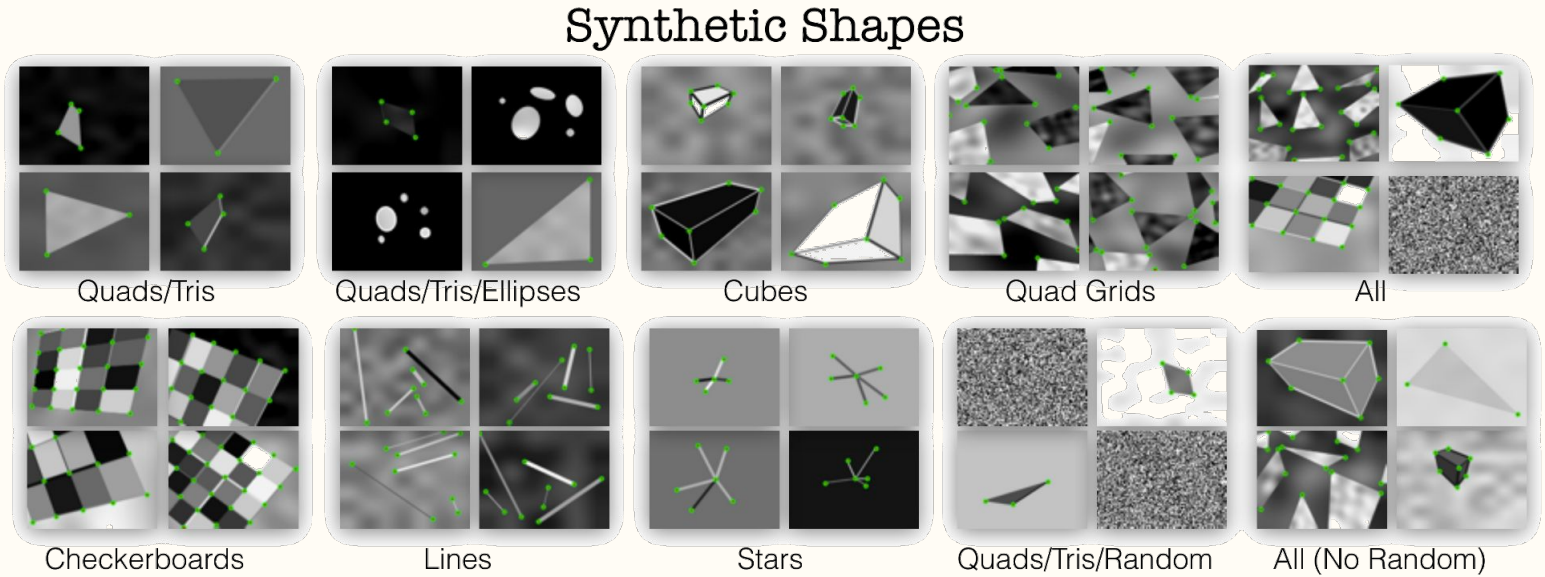
Model Training

- Siamese training with pairs of images
- Descriptor trained via metric learning (contrastive loss)
- Keypoints trained via supervised keypoint labels
  - Where do these come from?



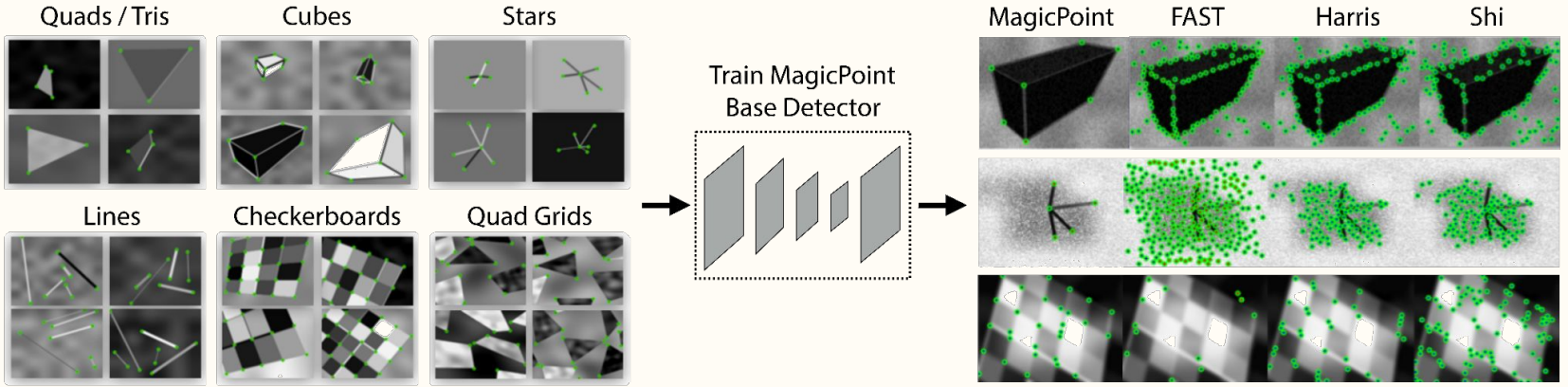
How to get Keypoint Labels?

- Need large scale dataset of annotated images
- Too hard for humans to label



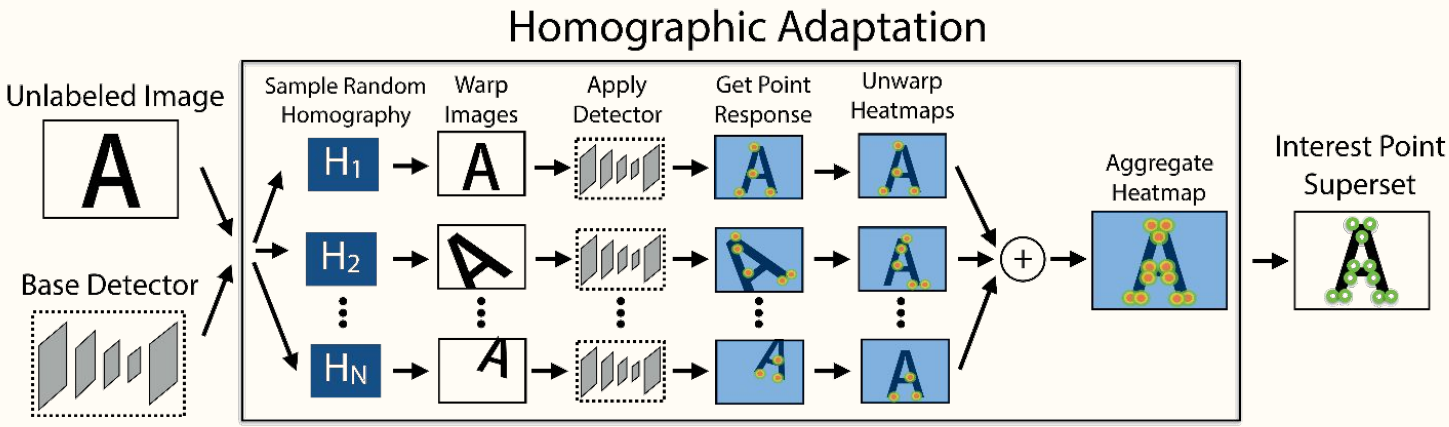
Self-Supervised Training

- Synthetic Training
  - Non-photorealistic shapes
  - Heavy noise
  - Effective and easy

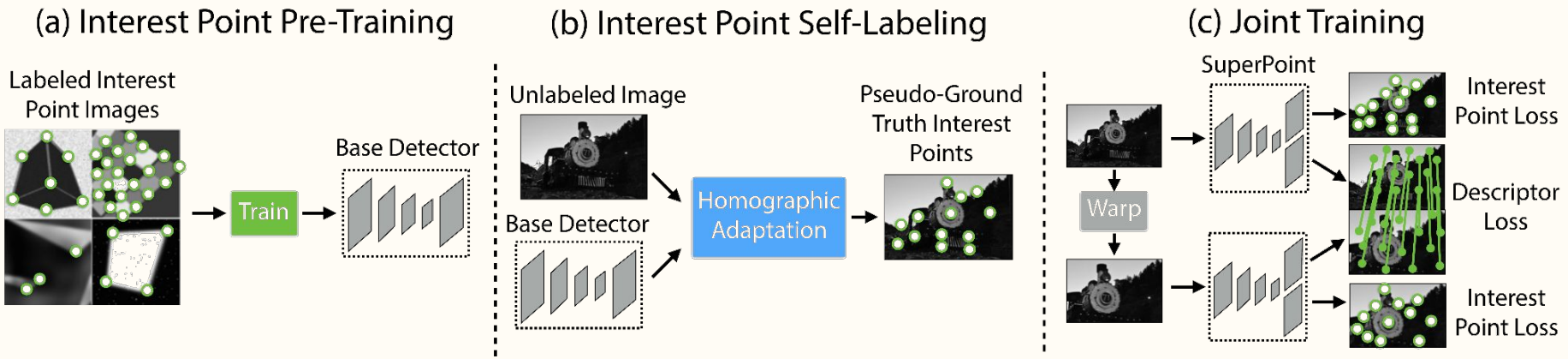


Self-Supervised Training

- Homographic Adaption
  - Use trained “MagicPoint” network
  - Simulate planar camera motion with homographies
  - Enhance repeatable points



Overview


(a) Interest Point Pre-Training
Labeled Interest Point Images
Train
Base Detector
(b) Interest Point Self-Labeling
Unlabeled Image
Base Detector
Homographic Adaptation
Pseudo-Ground Truth Interest Points
(c) Joint Training
Warp
SuperPoint
Interest Point Loss
Descriptor Loss
Interest Point Loss

Metrics

	57 Illumination Scenes		59 Viewpoint Scenes	
	NMS=4	NMS=8	NMS=4	NMS=8
SuperPoint	**.652**	**.631**	.503	**.484**
MagicPoint	.575	.507	.322	.260
FAST	.575	.472	.503	.404
Harris	.620	.533	**.556**	.461
Shi	.606	.511	.552	.453
Random	.101	.103	.100	.104

Table 3. **HPatches Detector Repeatability**. SuperPoint is the most repeatable under illumination changes, competitive on viewpoint changes, and outperforms MagicPoint in all scenarios.

	Homography Estimation			Detector Metrics		Descriptor Metrics	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	Rep.	MLE	NN mAP	M. Score
SuperPoint	.310	**.684**	**.829**	.581	1.158	**.821**	**.470**
LIFT	.284	.598	.717	.449	1.102	.664	.315
SIFT	**.424**	.676	.759	.495	**0.833**	.694	.313
ORB	.150	.395	.538	**.641**	1.157	.735	.266

Table 4. **HPatches Homography Estimation.** SuperPoint outperforms LIFT and ORB and performs comparably to SIFT using various ϵ thresholds of correctness. We also report related metrics which measure detector and descriptor performance individually.

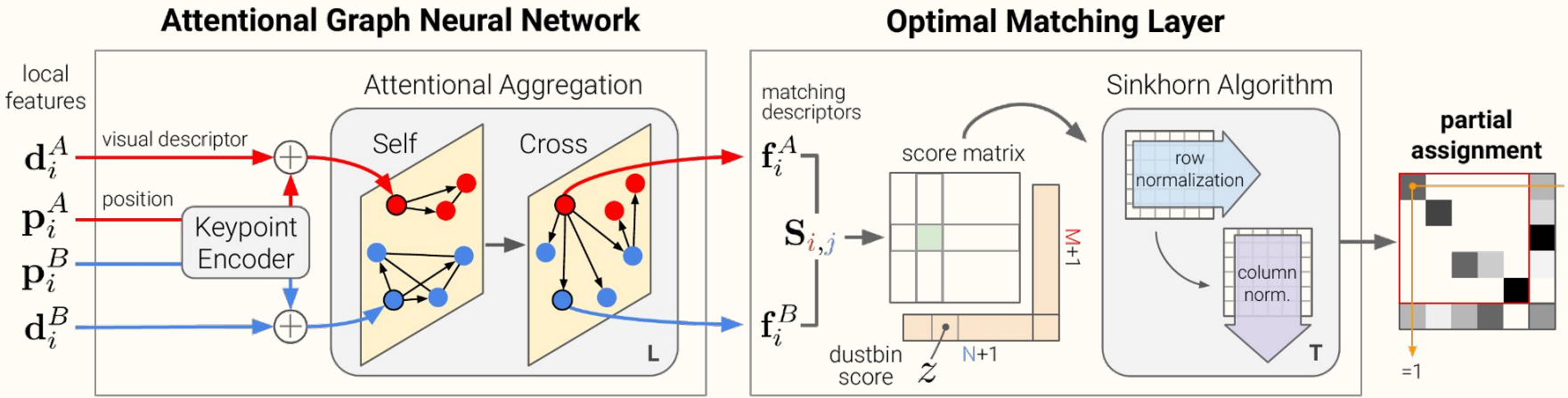
SuperGlue

Deep matching with SuperPoint: Can we learn to solve the correspondence problem?

Overview

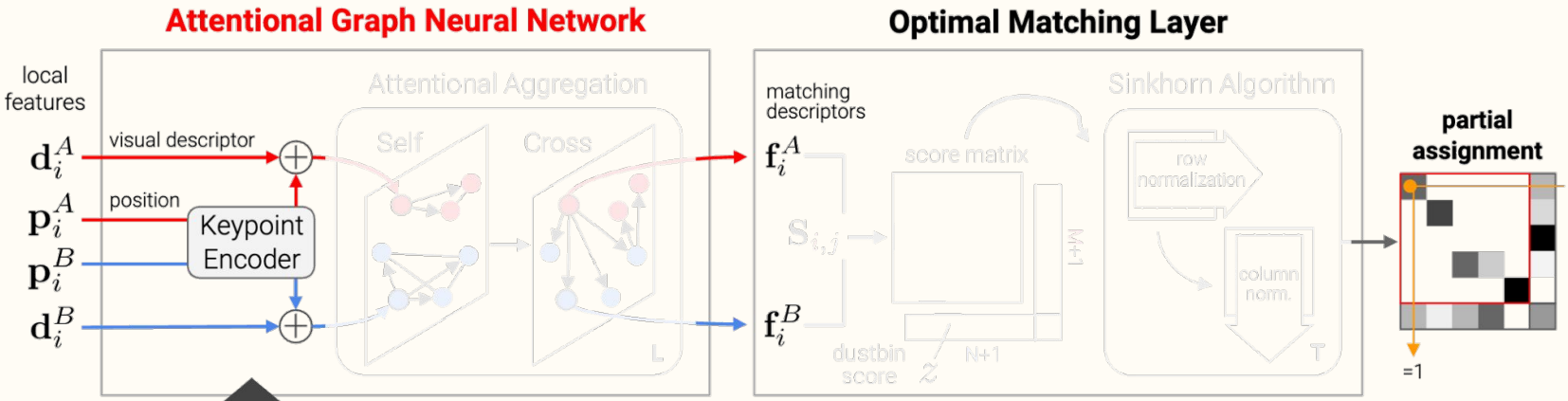
- Inputs
  - Images A and B
  - 2 sets of M, N local features (keypoints and descriptors)
- Outputs
  - Single a match per keypoint + occlusion using soft partial assignment
- Method
  - A Graph Neural Network (GNN) with attention
    - Encodes contextual cues & priors
    - Reasons about the 3D scene
  - Solving a optimal transport problem
    - Differentiable solver
    - Enforces the assignment constraints

Architecture



Attentional GNN

- Initial representation for each keypoint

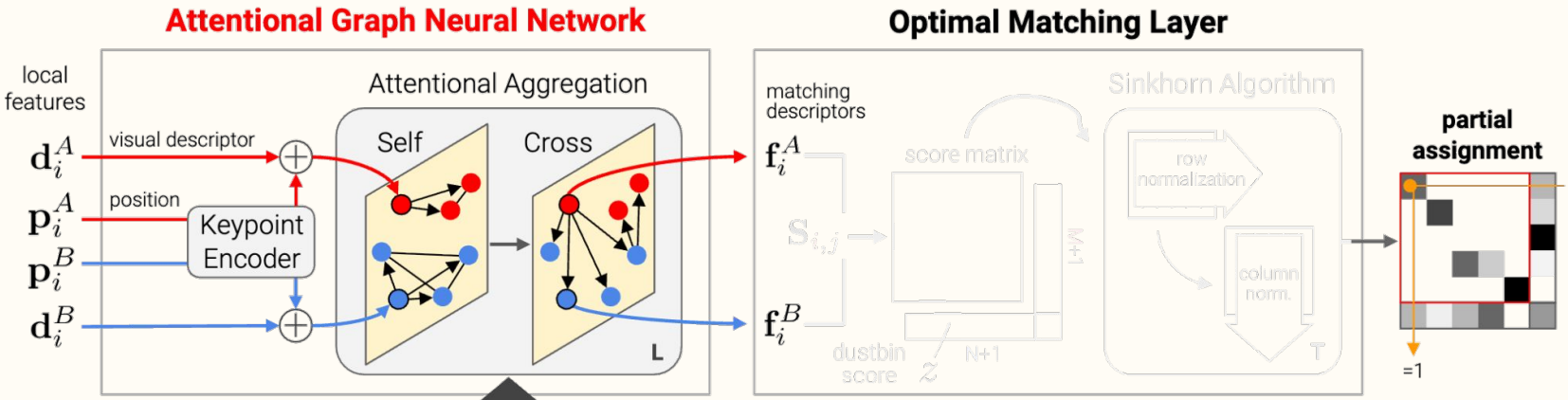


$$^{(0)}\mathbf{x}_i = \mathbf{d}_i + \mathrm{MLP}\left(\mathbf{p}_i\right)$$

Multi-Layer Perceptron

Attentional GNN

- Update the representation based on other keypoints - using a Message Passing Neural Network
  - a complete graph with two types of edges
  - in the same image: “self” edges
  - in the other image: “cross” edges



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

Attentional Aggregation

- Compute the message using self and cross attention

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

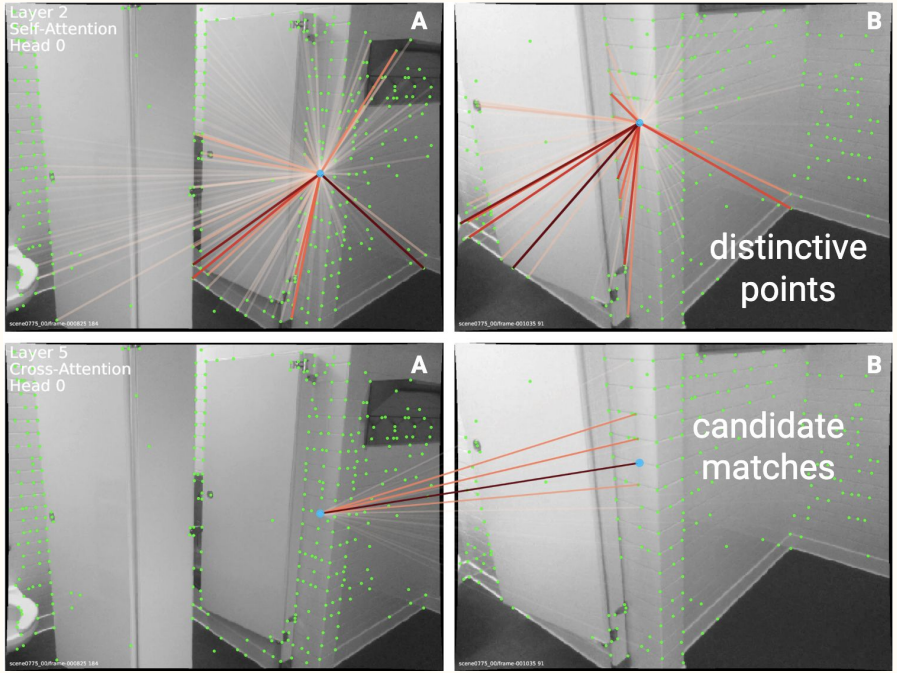
the message

$$ \mathbf{m}_{\mathcal{E}\to i} = \sum_{j:(i,j)\in\mathcal{E}} \alpha_{ij}\mathbf{v}_j \qquad \mathbf{q}_i = \mathbf{W}_1\,{}^{(\ell)}\mathbf{x}_i + \mathbf{b}_1 $$

$$ \alpha_{ij} = \mathrm{Softmax}_j\left(\mathbf{q}_i^\top \mathbf{k}_j\right) \qquad \begin{bmatrix}\mathbf{k}_j \\ \mathbf{v}_j\end{bmatrix} = \begin{bmatrix}\mathbf{W}_2 \\ \mathbf{W}_3\end{bmatrix} {}^{(\ell)}\mathbf{x}_j + \begin{bmatrix}\mathbf{b}_2 \\ \mathbf{b}_3\end{bmatrix} $$

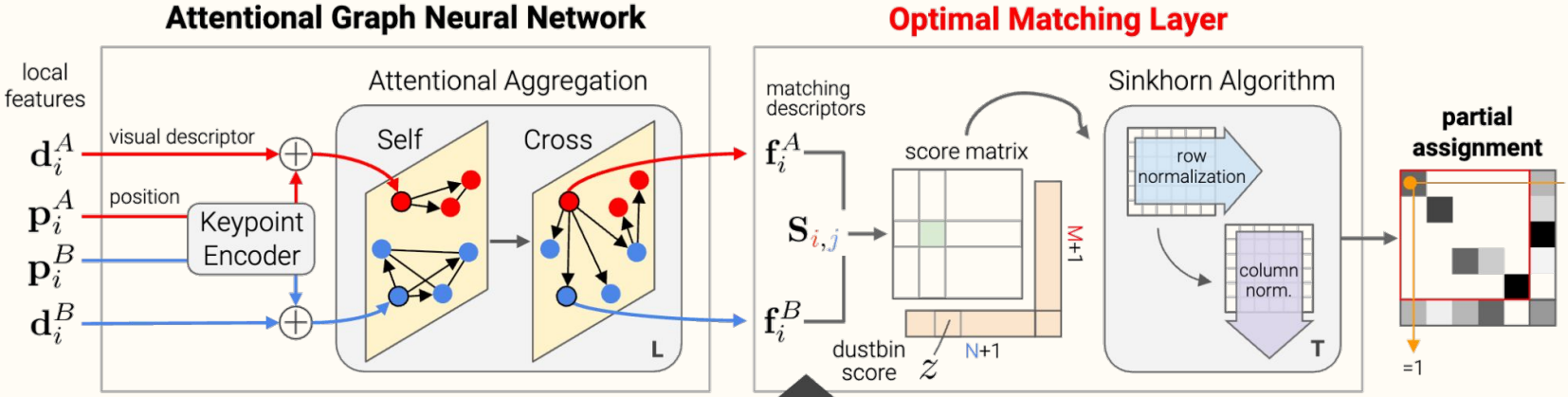
Attentional Aggregation

- Self-attention = intra-image information flow
- Cross attention = inter-image



Optimal Matching Layer

- Compute score matrix
- Occlusion and noise : unmatched keypoints are assigned to a dustbin
- Compute the partial assignment matrix
  - With the Sinkhorn algorithm: differentiable & soft Hungarian algorithm

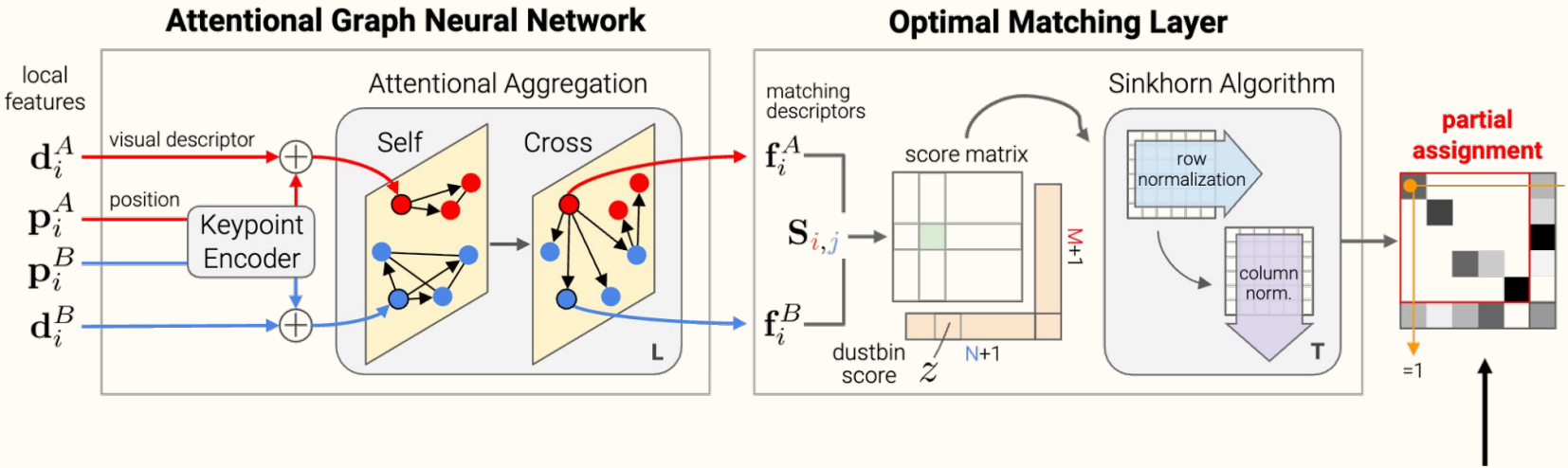


$$\mathbf{f}_i^A = \mathbf{W} \cdot {}^{(L)}\mathbf{x}_i^A + \mathbf{b}$$

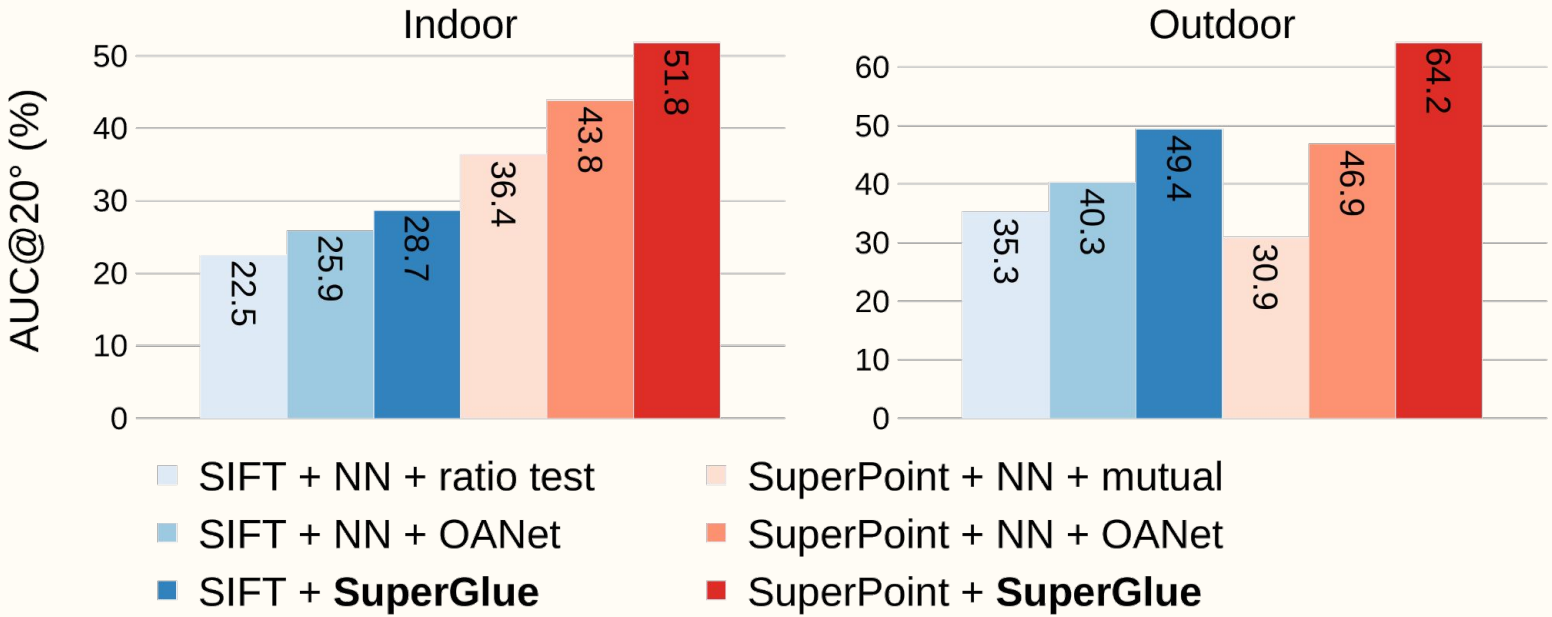
$$\mathbf{S}_{i,j} = < \mathbf{f}_i^A, \mathbf{f}_j^B >$$

Loss

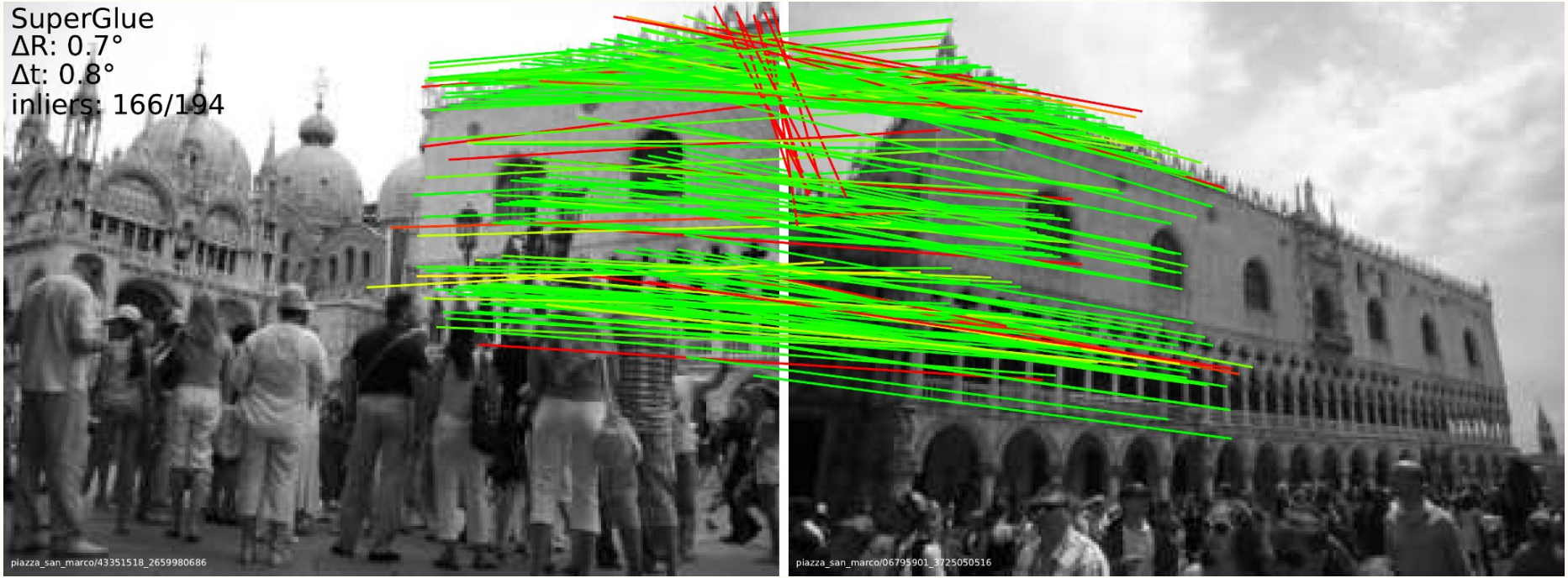
- Compute ground truth correspondences from pose and depth
- Find which keypoints should be unmatched
- Loss: maximize the log-likelihood of the GT cells in the partial assignment matrix



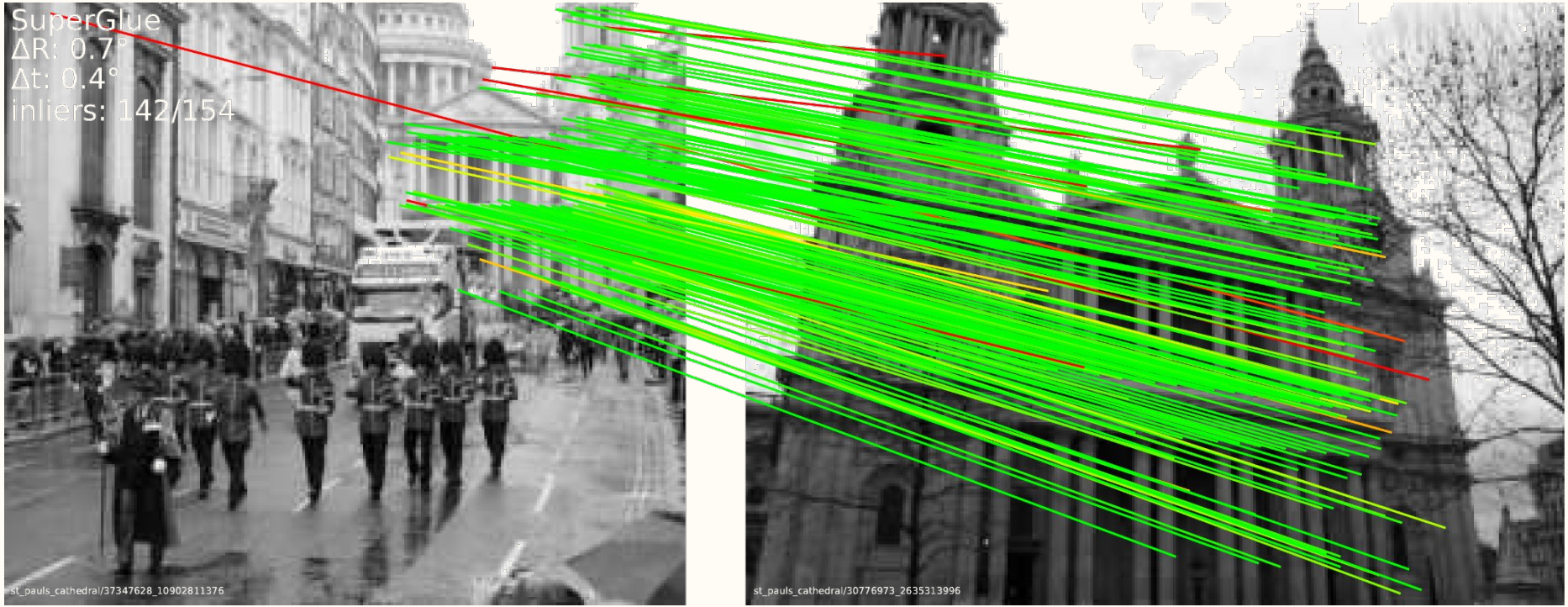
Metrics


Indoor
Outdoor
AUC@20° (%)
0
10
20
30
40
50
60
22.5
25.9
28.7
36.4
43.8
51.8
35.3
40.3
49.4
30.9
46.9
64.2
SIFT + NN + ratio test
SIFT + NN + OANet
SIFT + SuperGlue
SuperPoint + NN + mutual
SuperPoint + NN + OANet
SuperPoint + SuperGlue

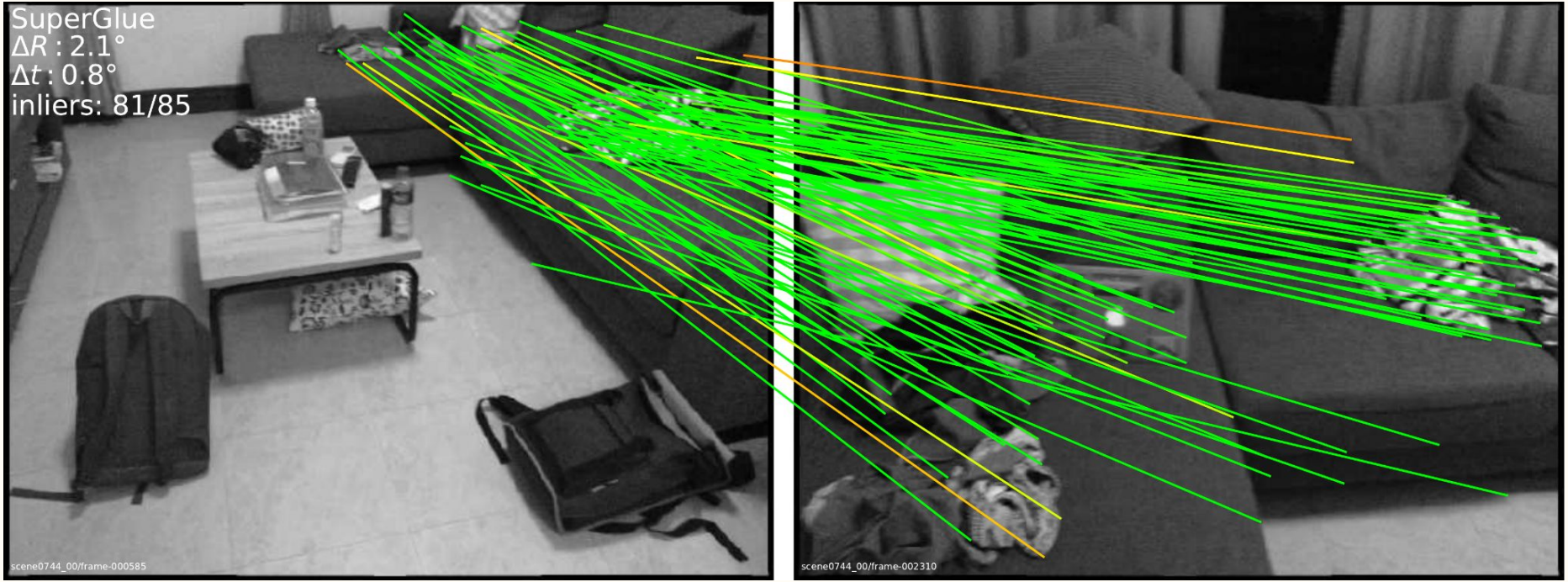
Qualitative Results



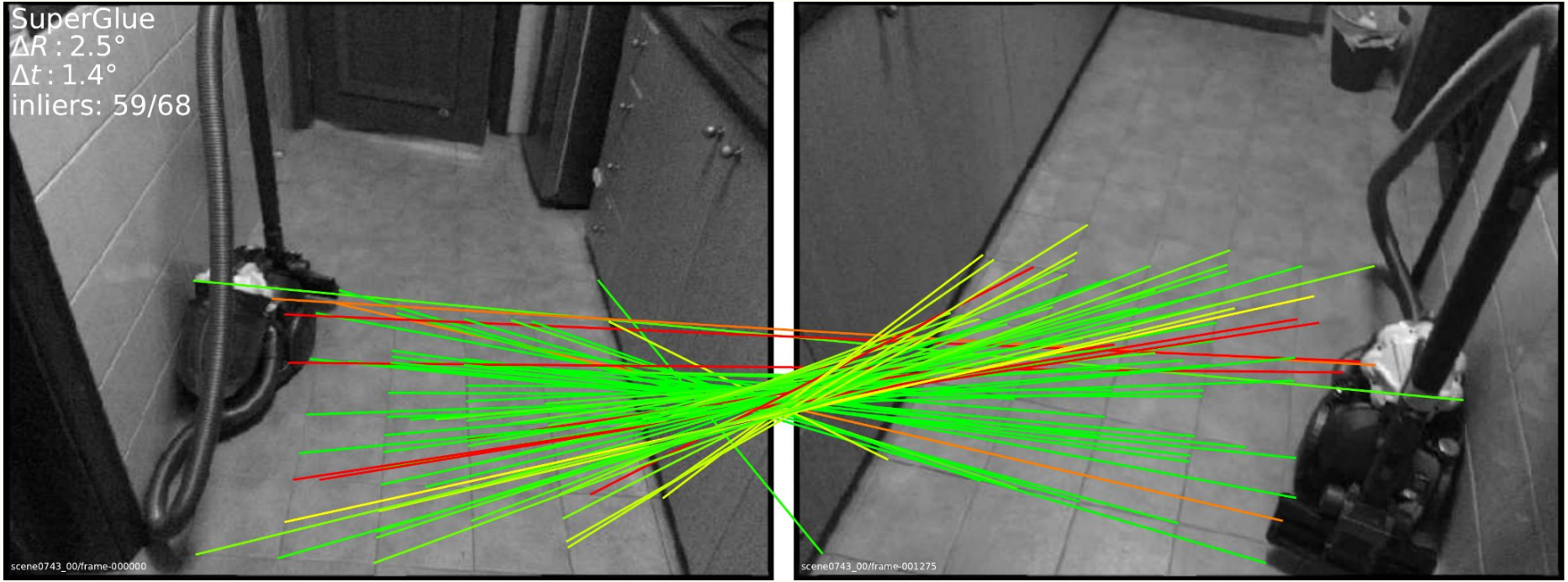
Qualitative Results



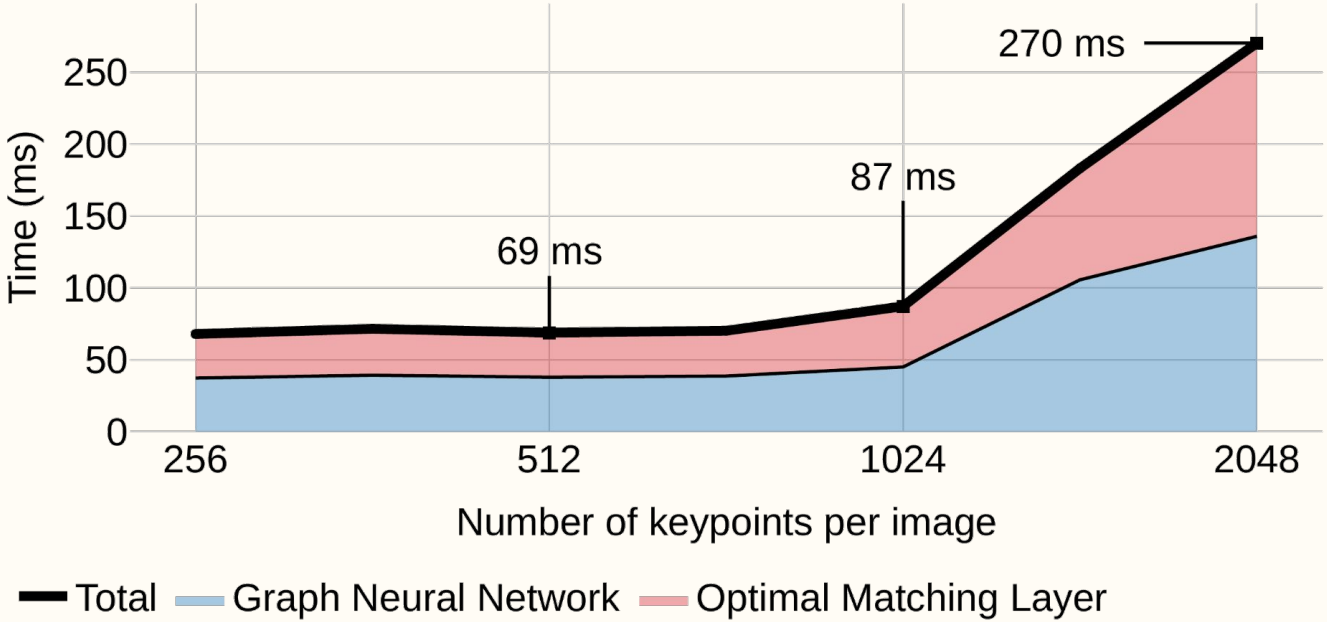
Qualitative Results



Qualitative Results



Run-time Analysis

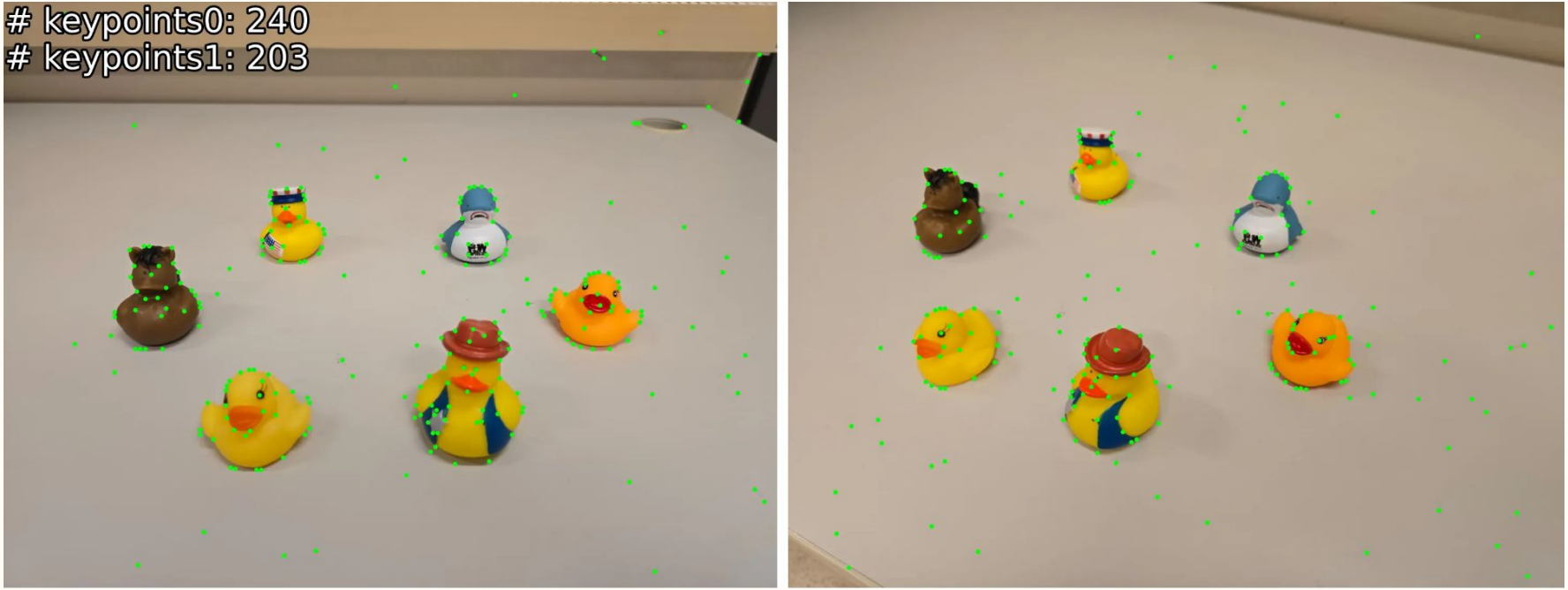


Duckie Dataset

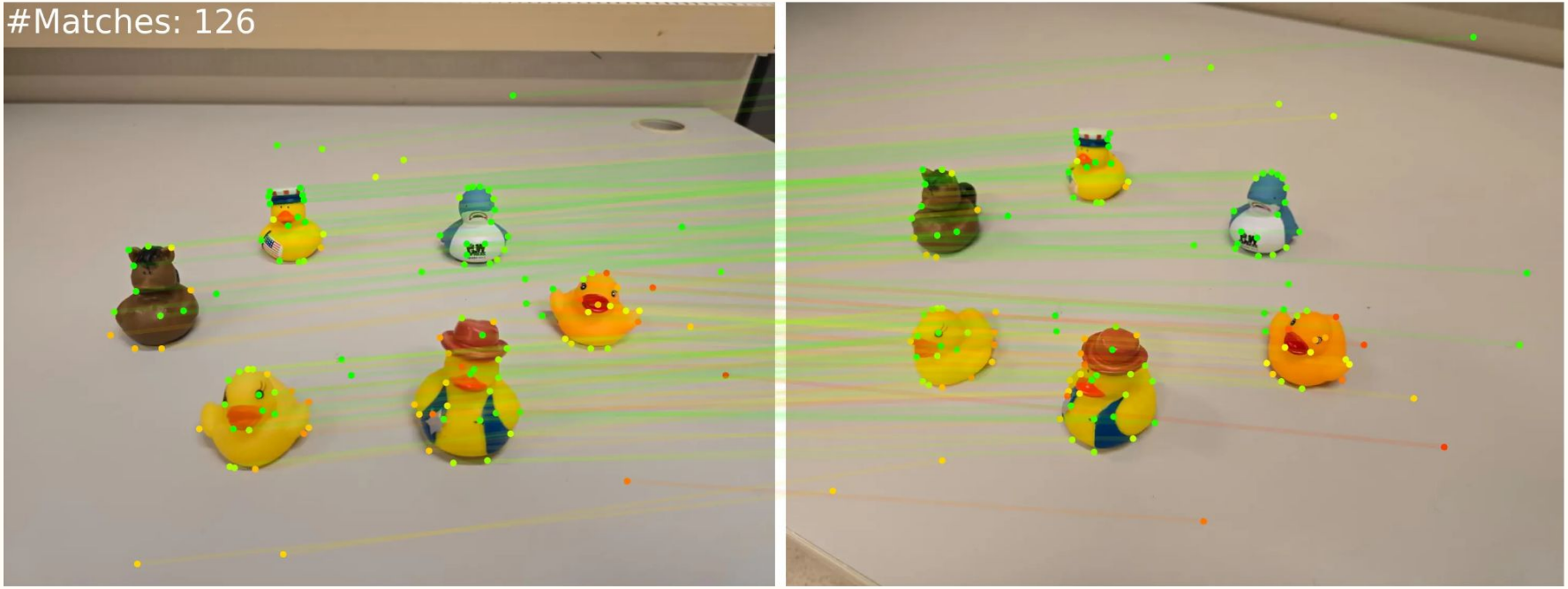
Why should they have all the fun?

N = 1

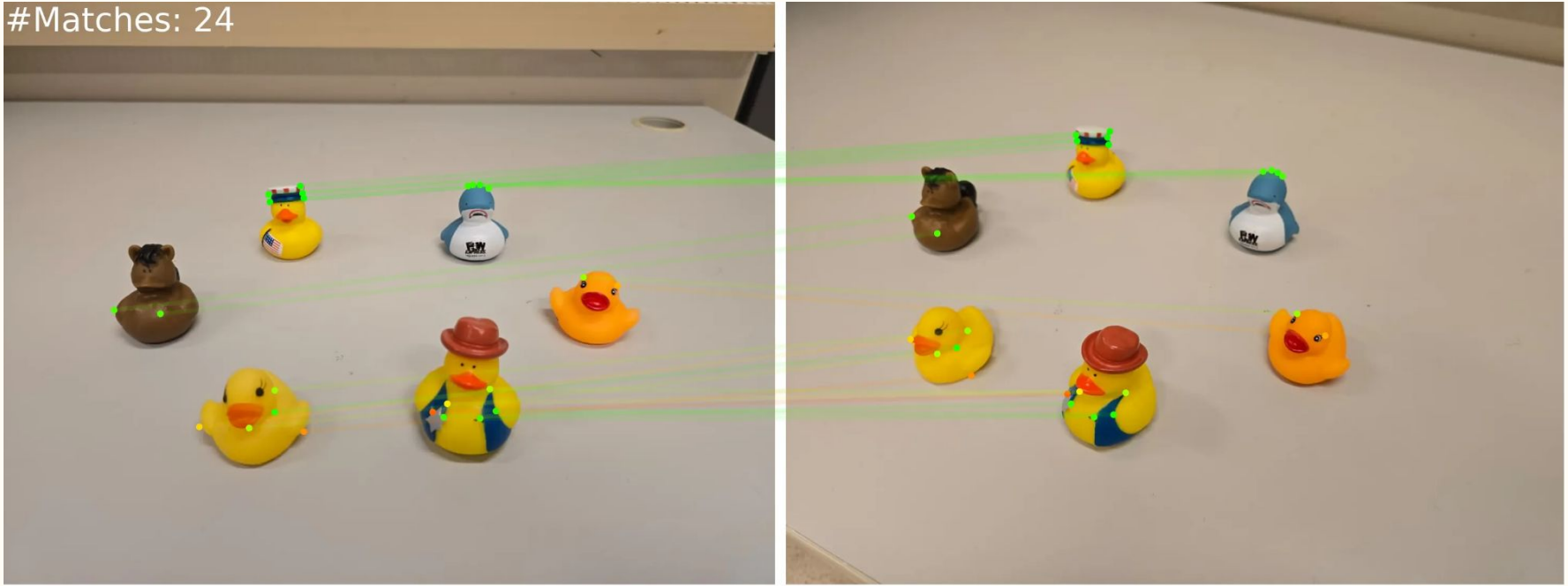
Duckie Dataset



Duckie Dataset



Duckie Dataset



Conclusions

- Problem: Robust feature detection & matching is critical for SLAM, SfM, and localization, but remains challenging under viewpoint, illumination, and texture changes.
- SuperPoint:
  - Learned detector + descriptor in a single CNN.
  - Self-supervised pretraining (synthetic homographies) enables real-world generalization.
- SuperGlue:
  - Learned matching with graph neural networks and optimal transport.
  - Context-aware, consistent correspondences outperform descriptor-only matching.
- Key lesson: Learning helps both where we detect features (SuperPoint) and how we match them (SuperGlue).

Other Relevant Work

- Earlier works
  - FAST - Features from Accelerated Segment Test
  - LIFT - Learned Invariant Feature Transform
- But can we learn generally useful image features?
  - DINO - Self-Distillation with No Labels
- Did the community just accept this approach?
  - DeDoDe - DeDoDe: Detect, Don't Describe -- Describe, Don't Detect for Local Feature Matching
  - MatchFormer - MatchFormer: Interleaving Attention in Transformers for Feature Matching
  - GlueStick - GlueStick: Robust Image Matching by Sticking Points and Lines Together
- Need for speed
  - LightGlue - Local Feature Matching at Light Speed
- Who needs detectors anyway?
  - LoFTR - Detector-Free Local Feature Matching with Transformers
  - RoMA - Robust Dense Feature Matching
- Scaling transformers for pointmap prediction
  - DUSt3R / MASt3R - Geometric 3D Vision Made Easy
  - VGGT - Visual Geometry Grounded Transformer

Questions?

Thank You!