Doppelgangers: Learning to Disambiguate Images of Similar Structures Supplemental Material

Ruojin Cai[1] Joseph Tung[1] Qianqian Wang[1]
Hadar Averbuch-Elor[2] Bharath Hariharan[1] Noah Snavely[1]
[1]Cornell University [2]Tel Aviv University

Contents

1. The Doppelgangers dataset

1.1. Data collection process

The process of creating image pairs with ground truth labels posed several challenges, including the difficulty of finding potential doppelgangers (as described in the main paper) and dealing with erroneously categorized images on Wikimedia Commons. Such images can lead to incorrect labels for image pairs that include them, which can affect the quality of our dataset. To address this issue, we propose to use a K-NN (K-Nearest Neighbor) algorithm to identify those images and ensure that Doppelgangers dataset comprises high-quality image pairs of similar structures with accurate labels.

Identifying incorrectly categorized images. While we find the most Wikimedia Commons images are correctly categorized, we found that some images are uploaded to the wrong subcategory, perhaps because people can themselves be confused about what side of a symmetric building they are looking at. Unfortunately, even a single incorrectly labeled image can lead to a large number of incorrect negative pairs that have many feature matches (because in reality they should be positive pairs). Therefore, to avoid noisy labels in our dataset, we must identify and remove such incorrectly categorized images. For this, we look at the scene graph computed by COLMAP [13] and remove images whose label is different from other images with a similar connectivity pattern in the scene graph.

Specifically, we use the K-NN (K Nearest Neighbor) algorithm [2] to identify such images, based on the similarity of connectivity computed from the scene graph. First, we construct an adjacency matrix A where each element $A(i, j)$ represents the number of matches between image i and image j. Next, we normalize the connectivity vector for each image to a unit vector, where the connectivity vector of image i is the i^{th} row vector of adjacency matrix A. We calculate the similarity of connectivity between any two images as the dot product of their respective connectivity vectors. Suspicious images are identified as those with different labels from their neighbors, and we remove pairs containing such images from our dataset.

1.2. Dataset Statistics

Table 1 and Table 2 provide additional statistics on the Doppelgangers training and test sets. The tables list the test scenes and training scenes that naturally form negative pairs, along with the average and the 95th percentile number of matches per scene. Our dataset includes a variety of landmarks, such as cathedrals, museums, castles, and other notable structures. The exteriors of these landmarks exhibit repeated and symmetric patterns. Most scenes in both the training and test sets average more than 50 matches.

2. Visual disambiguation

2.1. Implementation details of our method

Keypoint and match masks. Given a pair of images, we resize and pad them to a resolution of 1024×1024. We then use LoFTR [14], a learning-based feature matching method, to match the image pair. LoFTR produces matches and scores for each match. We filter out weak matches by

Training scene	Mean	95%
Aleppo Citadel	90	313
Almudena Cathedral	81	298
Arc de Triomphe du Carrousel	88	317
Brooklyn Bridge	47	152
Château de Chambord	96	344
Château de Cheverny	75	284
Château de Sceaux	134	629
Cinderella Castle	148	721
Cour Carrée (Louvre)	129	475
Cour Napoléon	94	295
Da Lat Station	84	253
Église de la Madeleine	119	390
Eiffel Tower	62	197
El Escorial	132	452
Grande Galerie (Louvre)	99	324
Grands Guichets du Louvre	140	321
Liberty Square, Taipei	65	197
London Eye	56	147
Mainz Cathedral	116	234
Market Square in Wrocław	118	429
Notre-Dame de Fourvière	92	318
Notre-Dame de Paris	174	730
Notre-Dame de Paris (Interior)	299	976
Notre-Dame de Strasbourg	122	437
Opéra Garnier	91	332
Patio de los Arrayanes	107	328
Patio de los Leones	126	306
Pavillion de Flore (Louvre)	63	212
Pont Alexandre III	55	131
Pont des Arts	74	157
Saint-Martin, Colmar	161	692
Salzburg Cathedral	98	303
St. Mark's Basilica	79	328
St. Paul's Cathedral	82	321
Statue of Liberty	38	113
Sukiennice	55	186
Taj Mahal	68	228
Torre de Belém	125	457
Umayyad Mosque (Courtyard)	76	252
White House	70	186

Table 1: Landmarks in the Doppelgangers training set. We present the average and 95th percentile number of matches per scene.

applying a threshold of 0.8 to the scores. To further refine matches, we perform geometric verification by estimating the fundamental matrix using RANSAC [7] with a reprojection error of 3 and a confidence level of 0.99. For this step, we use the publicly available OpenCV implementation. We use all the output matches to establish keypoint masks, and

Test scene	Mean	95%
Alexander Nevsky Cathedral, Lódz	47	115
Alexander Nevsky Cathedral, Prešov	62	231
Alexander Nevsky Cathedral, Sofia	87	244
Alexander Nevsky Cathedral, Tallinn	53	162
Arc de Triomphe de l'Étoile	100	387
Berlin Cathedral	77	372
Brandenburg Gate	36	95
Cathedral of St. Peter and Paul, Brno	283	1458
Charlottenburg Palace	40	104
Church of the Saviour on the Blood	53	195
Deutscher and Französischer Dom	67	139
Florence Cathedral	116	340
Sleeping Beauty Castle	37	112
St. Vitus Cathedral	138	666
Sydney Harbour Bridge	29	104
Washington Square Arch	70	206

Table 2: Landmarks in the Doppelgangers test set. We present the average and 95th percentile number of matches per scene.

the geometrically verified matches to establish match masks.

Input alignment. After obtaining the keypoints and matches, we estimate an affine transformation matrix using the OpenCV implementation of RANSAC with a inlier error of 20 pixels. We set a larger threshold, which means that an affine transform will only roughly fit the data, because we need a more tolerant threshold to have enough inliers to fit a transform at all. We use the estimated affine transformation matrix to align the images, keypoint masks, and match masks.

Network architecture. Our network architecture and parameter settings are similar to ResNet-18 [8], but we use three residual blocks with channel dimensions of 128, 256, and 512. After the average pooling layer, the last fully connected layer takes a 512-dimensional input and outputs a 2-dimensional vector. We then apply softmax to the vector to obtain probabilities.

Training. We train our network for 10 epochs using a batch size of 8 with two NVIDIA GeForce RTX 2080 Ti GPUs. The training process took approximately 9 hours for the 42 scenes and 30 hours for all scenes with image flipping augmentation. For optimization, we used the Adam optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, an initial learning rate of 5×10^{-4}, and linearly decayed the learning rate starting at epoch 5 until it reaches 5×10^{-6} at epoch 10.

2.2. Implementation details of baselines

We first provide additional details about the baselines evaluated in the main paper, then describe additional base-

lines provided in this supplemental material. We also evaluate two additional baselines, D2-Net [6]+RANSAC [7] and SuperPoint [4]+SuperGlue [12], with results provided in Section 2.3. With these two additional baselines, we cover a large variety of feature matching methods, including classical feature detectors such as SIFT and learning-based feature detectors such as D2-Net and SuperPoint. We also include traditional matching methods using nearest neighbor and RANSAC algorithms, as well as a learning-based matching method (SuperGlue). In addition, we evaluate detector-based feature matching methods and detector-free feature matching methods, such as LoFTR. Note that all local feature matching baselines are used as classifiers on image pairs by thresholding either the number of matches, or the ratio of number of matches to number of keypoints.

SIFT [11]+RANSAC [7]. We use the COLMAP [13] feature extraction and matching modules to produce keypoints and matches, using the default parameters. This includes the maximum extracted features set to 8192, use of cross check for matching, and geometric verification with a reprojection error of 4 and confidence level of 0.999.

LoFTR [14]. We follow the same process as previously described to use LoFTR to obtain matches for our network input.

DINO [1]. We use the pretrained ViT [5] small version model with a patch size of 16. We pass one image at a time to DINO and obtain the latent code and feature maps from the last layer. We then train a linear classifier by taking the concatenated latent codes of images in a pair as input to a fully connected layer and outputting the probability. For the feature maps, we concatenate them and pass them through a residual layer and fully connected layer to obtain the prediction.

D2-Net [6]+RANSAC [7]. D2-Net is a learning-based method for feature detector and descriptor. We use its pretraine model on MegaDepth [10] to extract keypoints and descriptors. We then use the OpenCV implementation of the brute force k-nearest neighbor matcher with cross-check setting, and apply a ratio test with a threshold of 0.75. Finally, we perform geometric verification with the same settings as previously described.

SuperPoint [4]+SuperGlue [12]. SuperPoint is an efficient learning-based method for detecting and describing keypoints. Given the keypoints and descriptors extracted from SuperPoint, we use SuperGlue to obtain matches, where SuperGlue is a learning-based approach for feature matching using a graph neural network (GNN). We use the checkpoint trained for outdoor scenes with the recommended settings for SuperGlue, including a maximum number of keypoints set to 2048 and a Non-Maximum Suppression (NMS) radius of 3.

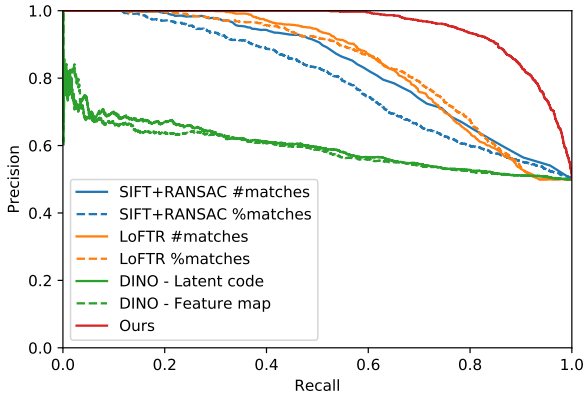

Figure 1: Precision-Recall (PR) curves on the Doppelgangers test set. The x-axis represents recall and the y-axis represents precision. A curve approaching the top-right corner indicates better performance.

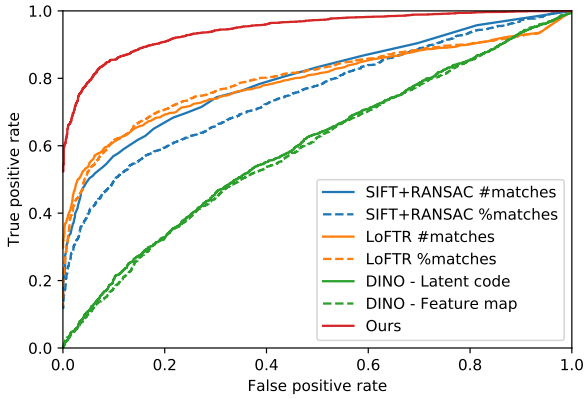

Figure 2: Receiver operating characteristic (ROC) curves on the Doppelgangers test set. The x-axis represents the false positive rate, and the y-axis represents the true positive rate. The ideal method would simultaneously have a lower false positive rate and higher true positive rate, with the curve approaching the top-left corner.

2.3. Quantitative results and analysis

We present additional comparisons of our method with baselines on the Doppelgangers test set, and report the average precision (AP) and ROC AUC scores in Tables 3 and 4, respectively. Both the AP and ROC AUC scores evaluate classification performance, and higher scores are better. While AP is more focused on positive pairs, ROC AUC is more focused on the ranking of predictions and cares equally about positive and negative pairs. Our method outperforms all other baselines for 15 out of 16 test landmarks, with an average precision (AP) of 95.2% and an ROC AUC of 93.8% across all landmarks. The SuperPoint+SuperGlue method achieves comparable results to SIFT+RANSAC, while D2-

Average Precision	D2-Net [6]+RANSAC [7]		SuperPoint [4]+SuperGlue [12]		SIFT [11]+RANSAC [7]		LoFTR [14]		DINO [1]-ViT		Ours
	#matches	%matches	#matches	%matches	#matches	%matches	#matches	%matches	Latent code	Feature map	
Average of all pairs from 16 landmarks	62.3	62.5	79.6	80.7	83.4	81.2	85.3	86.0	62.0	63.3	**95.2**
Alexander Nevsky Cathedral, Łódź	62.1	63.7	83.7	83.8	72.7	75.9	80.7	80.4	50.9	50.3	**89.5**
Alexander Nevsky Cathedral, Sofia	63.6	63.7	80.6	80.7	89.5	87.6	90.0	92.2	53.0	53.6	**98.5**
Alexander Nevsky Cathedral, Tallinn	64.1	64.4	73.9	74.3	73.1	76.0	76.1	80.3	58.8	50.8	**86.2**
Arc de Triomphe	49.7	48.8	60.1	60.9	86.1	81.7	85.7	93.3	55.4	61.1	**97.6**
Berlin Cathedral	69.0	69.7	94.4	94.6	91.8	91.6	93.6	92.7	76.4	70.6	**99.4**
Brandenburg Gate	42.1	42.8	60.1	62.6	79.3	73.7	90.9	95.6	60.8	60.9	**99.8**
Cathedral of Saints Peter and Paul in Brno	75.0	74.6	93.1	93.1	95.8	96.4	89.8	88.4	64.6	79.9	**99.8**
Cathedral of St Alexander Nevsky, Prešov	73.2	74.5	89.8	89.8	82.5	74.0	86.1	85.3	62.9	64.8	**94.6**
Charlottenburg Palace	62.0	60.6	81.4	82.3	81.5	76.1	85.6	81.1	65.8	54.1	**93.3**
Church of Savior on the Spilled Blood	62.1	61.0	86.7	86.2	82.1	73.2	84.9	75.5	63.9	67.5	**93.8**
Deutscher und Französischer Dom (Berlin)	53.9	54.6	75.4	75.9	74.5	71.9	85.8	84.2	55.6	51.5	**98.1**
Florence Cathedral	60.1	58.0	82.7	82.8	90.6	83.8	84.5	82.0	54.6	63.8	**94.2**
Sleeping Beauty Castle	54.4	56.8	71.0	81.1	81.1	81.2	75.0	85.6	67.2	66.4	**97.1**
St. Vitus Cathedral	68.8	67.6	91.7	91.0	96.8	88.0	89.2	87.5	84.0	77.0	**99.8**
Sydney Harbour Bridge	73.5	77.3	83.6	86.8	79.4	**92.3**	83.8	86.2	53.0	75.5	87.0
Washington Square Arch	63.6	62.4	65.0	65.5	77.7	75.9	82.8	86.0	65.2	65.0	**95.1**

Table 3: Quantitative results for visual disambiguation evaluated on Doppelgangers. Results are reported as the average precision (AP) multiplied by 100. We report both the average and the per-scene results for 16 landmarks.

ROC AUC	D2-Net [6]+RANSAC [7]		SuperPoint [4]+SuperGlue [12]		SIFT [11]+RANSAC [7]		LoFTR [14]		DINO [1]-ViT		Ours
	#matches	%matches	#matches	%matches	#matches	%matches	#matches	%matches	Latent code	Feature map	
Average of all pairs from 16 landmarks	53.5	53.7	76.8	76.9	80.2	77.1	78.9	80.3	60.9	61.5	**93.8**
Alexander Nevsky Cathedral, Łódź	58.5	60.1	78.7	78.7	69.7	72.7	73.9	74.8	49.2	49.7	**87.0**
Alexander Nevsky Cathedral, Sofia	57.1	57.7	82.5	82.5	87.7	84.3	86.2	89.1	53.8	49.3	**98.0**
Alexander Nevsky Cathedral, Tallinn	59.2	60.0	71.8	71.9	68.0	71.7	71.9	74.5	60.8	52.2	**84.2**
Arc de Triomphe	39.8	38.5	44.7	44.7	81.6	75.3	78.5	88.9	53.7	57.1	**96.9**
Berlin Cathedral	54.8	56.1	92.1	92.2	89.2	88.6	89.4	88.1	71.6	67.7	**99.3**
Brandenburg Gate	33.7	35.2	64.3	64.5	77.9	71.9	87.5	93.4	60.7	60.4	**99.8**
Cathedral of Saints Peter and Paul in Brno	61.2	60.4	89.6	89.6	94.0	95.0	84.4	82.8	62.7	75.8	**99.8**
Cathedral of St Alexander Nevsky, Prešov	62.3	64.8	87.4	87.4	77.0	63.9	77.4	77.6	68.9	60.6	**92.4**
Charlottenburg Palace	52.0	50.5	78.0	78.2	76.7	70.7	80.2	77.1	65.9	53.6	**92.2**
Church of Savior on the Spilled Blood	50.2	49.2	77.5	77.3	77.7	68.4	78.6	70.4	61.5	64.0	**92.5**
Deutscher und Französischer Dom (Berlin)	51.8	52.1	79.2	79.2	70.7	68.2	80.4	78.7	58.7	49.2	**97.6**
Florence Cathedral	52.7	49.5	78.2	78.2	88.7	80.3	74.6	71.9	51.3	62.5	**92.5**
Sleeping Beauty Castle	48.7	52.7	77.0	77.5	76.8	79.4	64.7	78.4	64.7	66.5	**96.0**
St. Vitus Cathedral	49.6	47.4	82.4	82.4	96.7	82.5	82.3	80.3	80.4	80.5	**99.8**
Sydney Harbour Bridge	69.4	71.8	86.3	86.3	76.3	**91.7**	77.4	79.0	50.2	72.4	80.0
Washington Square Arch	55.9	53.9	59.6	59.6	73.9	69.3	74.7	79.2	59.9	63.2	**93.5**

Table 4: Quantitative results for visual disambiguation evaluated on Doppelgangers. Results are reported as ROC AUC multiplied by 100. We report both the average and the per-scene results for 16 landmarks.

Net performs worse and similarly to DINO. These results suggest that the presence of the number or ratio of matches is not necessarily the best indicator of whether two images truly match.

We also present evaluation results as the precision-recall (PR) curves shown in Figure 1, where our method shows significant improvements over the baselines. We also provide receiver operating characteristic (ROC) curves in Figure 2. ROC curves illustrate the performance of classifiers across various classification thresholds. Our model consistently outperforms other methods across all thresholds, with the lowest false positive rate and highest true positive rate. Additionally, in Figure 3, we show the confusion matrix of our network predictions using a threshold of 0.5, indicating that our method can correctly classify approximately 88% of image pairs in the test set at this threshold.

To analyze the correlation between our network's predictions and the number of matches in the input pair, we generate 2D scatter plots where the x-axis is the number of matches and the y-axis is the probability predicted by our network. The resulting scatter plots are shown in Figure 4, using SIFT+RANSAC and LoFTR methods to compute matches, respectively. In the figure, red dots represent pairs with a ground truth label of negative, while blue dots represent positive pairs. The figure shows that our method can differentiate between positive and negative image pairs, in particular in cases when such pairs have the same number of matches. Although differentiating doppelganger pairs with larger numbers of matches can be more challenging (red dots at top right of figure), our method still predicts a probability lower than 0.8 for most negative pairs.

2.4. Additional ablation study

We conduct an additional ablation study on the design of network input. The results, reported as average precision scores, are shown in Table 5. As described in the main paper, we conduct a *w/o Augmentation* experiment where we train the classifier on 44 scenes without flip augmentations. The remaining variations are trained on the same dataset of 44 scenes without augmentation for speed of training.

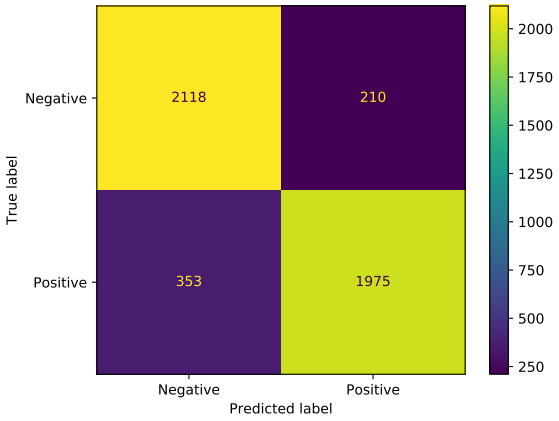

Figure 3: Confusion matrix for our method with probability threshold set to 0.5. Our method correctly identifies 1,975 true positive pairs and 2,118 true negatives, while producing 353 false positive pairs and 210 false negatives. Overall, the method achieves an accuracy of approximately 88%.

Full	95.2
w/o Augmentation	93.6
w/o Masks	64.7
w/o RGB	90.0
w/o Geo. verification	92.1

Table 5: Additional ablation study on network input design. The results are reported as the average precision multiplied by 100.

In the *w/o Masks* setting, we remove keypoint and match masks from input, leaving only RGB images. This results in significant degeneration of performance. In the *w/o RGB* experiment, we remove RGB images from the input, leaving only keypoint and match masks. This leads to a drop in average precision from 93.6% to 90.0%. This drop is not as significant as that stemming from removal of keypoint and match masks, indicating the relative importance of these inputs. The *w/o Geo. verification* setting is one where matches are not filtered and verified with Fundamental matrix estimation using RANSAC, resulting in a decrease in average prevision from 93.6% to 92.1%. In summary, the ablation study demonstrates that keypoint and match masks are essential components of input for visual disambiguation, as they contain rich information and cues for differentiating visually similar pairs.

2.5. Additional qualitative results

In Figure 5, we provide additional visualizations of test image pairs and their corresponding predicted probability by our method on a variety of test scenes.

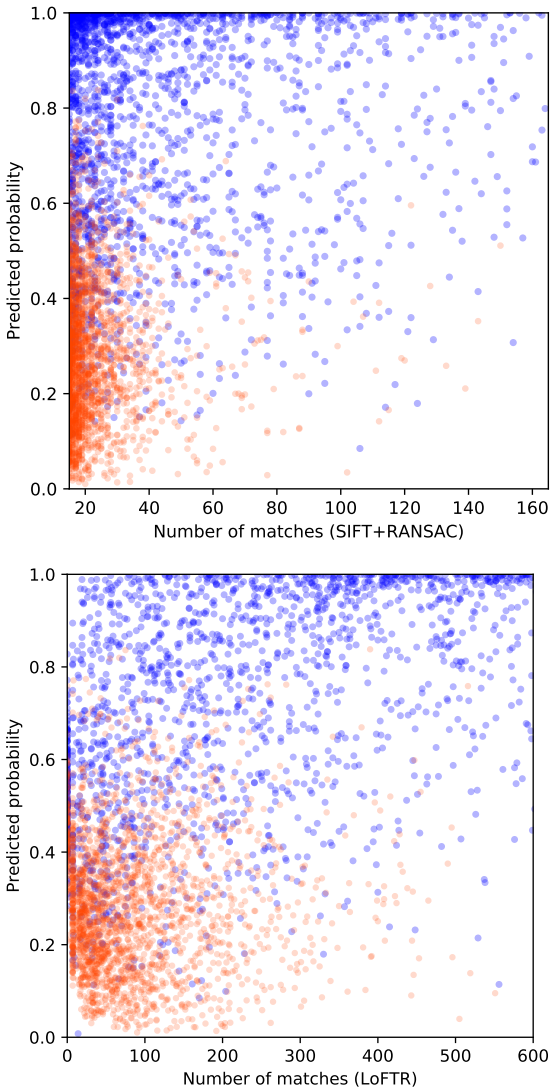

Figure 4: Correlation between predicted probability of our network and number of matches using SIFT+RANSAC (top figure) and LoFTR (bottom figure) for pairs in all test scenes. The x-axis represents the number of matches and the y-axis represents the predicted probability. Blue dots represent ground truth positive pairs and red dots represent ground truth negative pairs. Our method produces a high probability for most positive pairs and a low probability for most negative pairs. Even for challenging doppelganger pairs, our method can produce a probability of less than 0.85. In contrast, SIFT and LoFTR methods have lots of positive pairs with low numbers of matches, as well as a number of negative pairs with large numbers of matches.

We visualize some failure cases in Figure 6, all of which are negative pairs. We circle potentially useful regions for visual disambiguation in red. The pair from Alexander Nevsky

	Images	COLMAP [13]	[13] #matches>150	Heinly et al. [9]	Wilson et al. [15]	Cui et al. [3]	Yan et al. [16]	Ours					
								@0.5	@0.6	@0.7	@0.8	@0.9	@0.97
Alexander Nevsky Cathedral [9]	448	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Arc de Triomphe [9]	434	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓
Berliner Dom [9]	1,618	✗	✓	✓	✗*	✓	✗*	✓	✓	✓	✓	✓	✗*
Big Ben [9]	402	✗	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓
Brandenburg Gate [9]	175	✗	✓	✓	–	✗	✗	✗	✓	✓	✓	✓	✓
Church on the spilled blood [9]	277	✗	✗	✓	–	✗	✗	✗	✗	✗	✗	✗	✓
Radcliffe camera [9]	282	✗	✓	✓	✗*	✓	✓	✗	✓	✓	✓	✓	✓
Number of scenes: ✓/✗*/✗		0/0/7	3/0/4	7/0/0	0/2/3	4/0/3	3/1/3	4/0/3	6/0/1	6/0/1	6/0/1	6/0/1	6/1/0

Table 6: Robustness evaluation of our method to the probability threshold on SfM disambiguation results. ✓ means correctly disambiguate and reconstruct. ✗ means fail to disambiguate and ✗* means over-split. Our method exhibits robustness to the probability threshold and successfully reconstructed 6 out of 7 scenes with probability thresholds ranging from 0.6 to 0.97.

Cathedral in Tallinn has distinct regions on the facades that are difficult to observe due to the viewpoint. Given other regions and structures of the building appear similar, it is challenging even for humans to differentiate between the images. In the pair from Charlottenburg Palace, the second image is a zoom-in view that crops out other regions, leaving only a small region on the golden sculpture (at the top of the building) that can serve as a cue for visual disambiguation. In the third pair from Washington Square Arch, the illumination differences might mask the structural differences (which are in shadow in the second image), making it more difficult to discern the differences between regions. The replicas of Sleeping Beauty Castles look very similar, as shown in the last pair of images. Images captured at night can be more challenging to distinguish, since the background is obscured and important cues may be lost due to lack of observability in the background.

3. Structure from Motion disambiguation

3.1. Threshold robustness evaluation

We evaluate the robustness of our method for disambiguating SfM reconstructions to the probability threshold, and we show additional results on 7 landmark datasets from Heinly et al. [9] with thresholds at [0.5, 0.6, 0.7, 0.8, 0.9, 0.97] in Table 6. At the threshold of 0.5, some incorrect pairs are included in the scene graph, resulting in broken reconstructions for Brandenburg Gate, Church on Spilled Blood, and Radcliffe Camera. For the SfM disambiguation setting, where a single bad matching pair can break a model, we care more about false positives than keeping all positive pairs (i.e., we care more about precision than recall). Therefore setting the threshold to 0.5 may intuitively not be the best strategy, hence the better performance at higher thresholds that filter out more pairs. For thresholds ranging from 0.6 to 0.9, our method is robust, and successfully disambiguates and reconstructs 6 out of 7 scenes. At even higher thresholds, we see that one of the models (Berliner Dom) splits apart, resulting in over-splitting of the reconstruction, but at this strict threshold we can successfully disambiguate the final scene (Church on Spilled Blood). Overall, our method is able to reconstruct 6 out of 7 scenes even at this threshold, demonstrating the robustness and effectiveness of our approach.

3.2. Detailed reconstruction visualization

We present a detailed visualization of the reconstruction results for 7 scenes rendered from different viewpoints in Figure 7, comparing our method with vanilla COLMAP reconstruction. The visualizations from different viewpoints provide a clear view of the incorrect structures produced by COLMAP, such as the double towers in the Alexander Nevsky Cathedral and the missing sides of Big Ben. Our method can disambiguate different sides of these highly symmetric landmarks and produce a complete and correct reconstruction.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 3, 4

[2] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. 1

[3] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *ICCV*, pages 864–872, 2015. 6

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, pages 224–236, 2018. 3, 4

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[6] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019. 3, 4

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to

Figure 5: Additional visual disambiguation results. We visualize test image pairs with their corresponding predicted probabilities produced by our network. The left column shows negative pairs and the right column shows positive pairs.

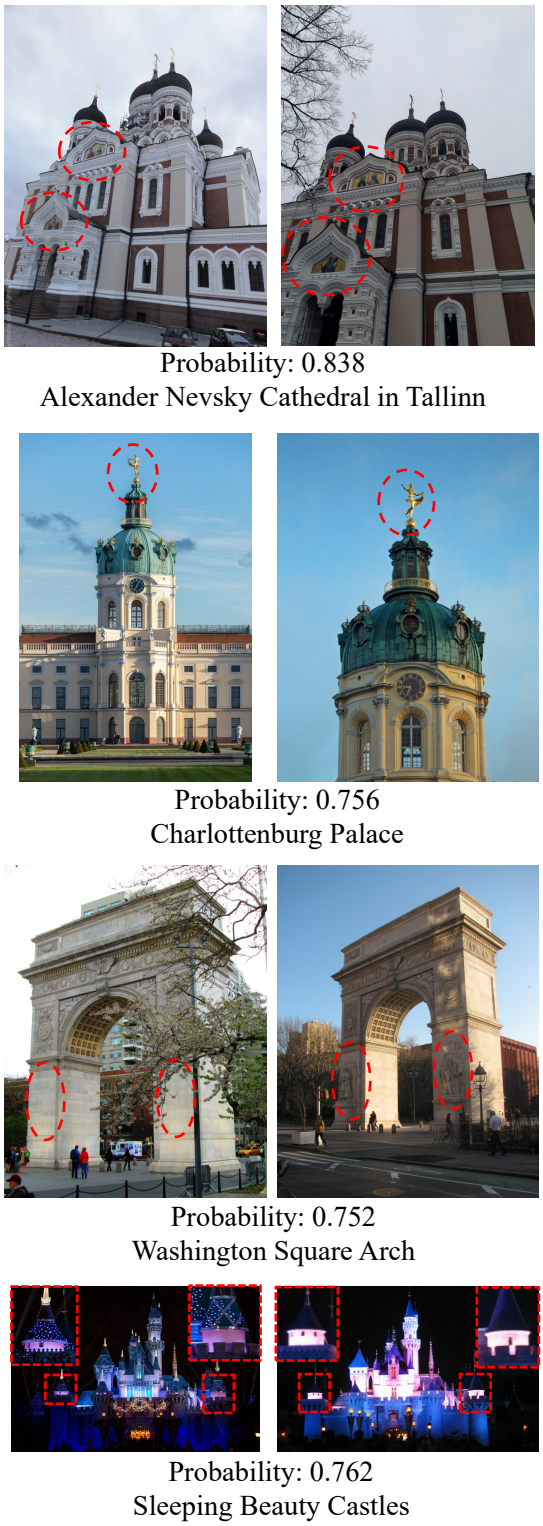

Figure 6: Failure cases. We visualize challenging doppelgangers pairs that are all negative pairs, but the predicted probabilities by our network are high. We circle the regions that might be helpful for disambiguation in red. For the last pair from Sleeping Beauty Castles, we show zoomed-in views of distinct regions with red boxes.

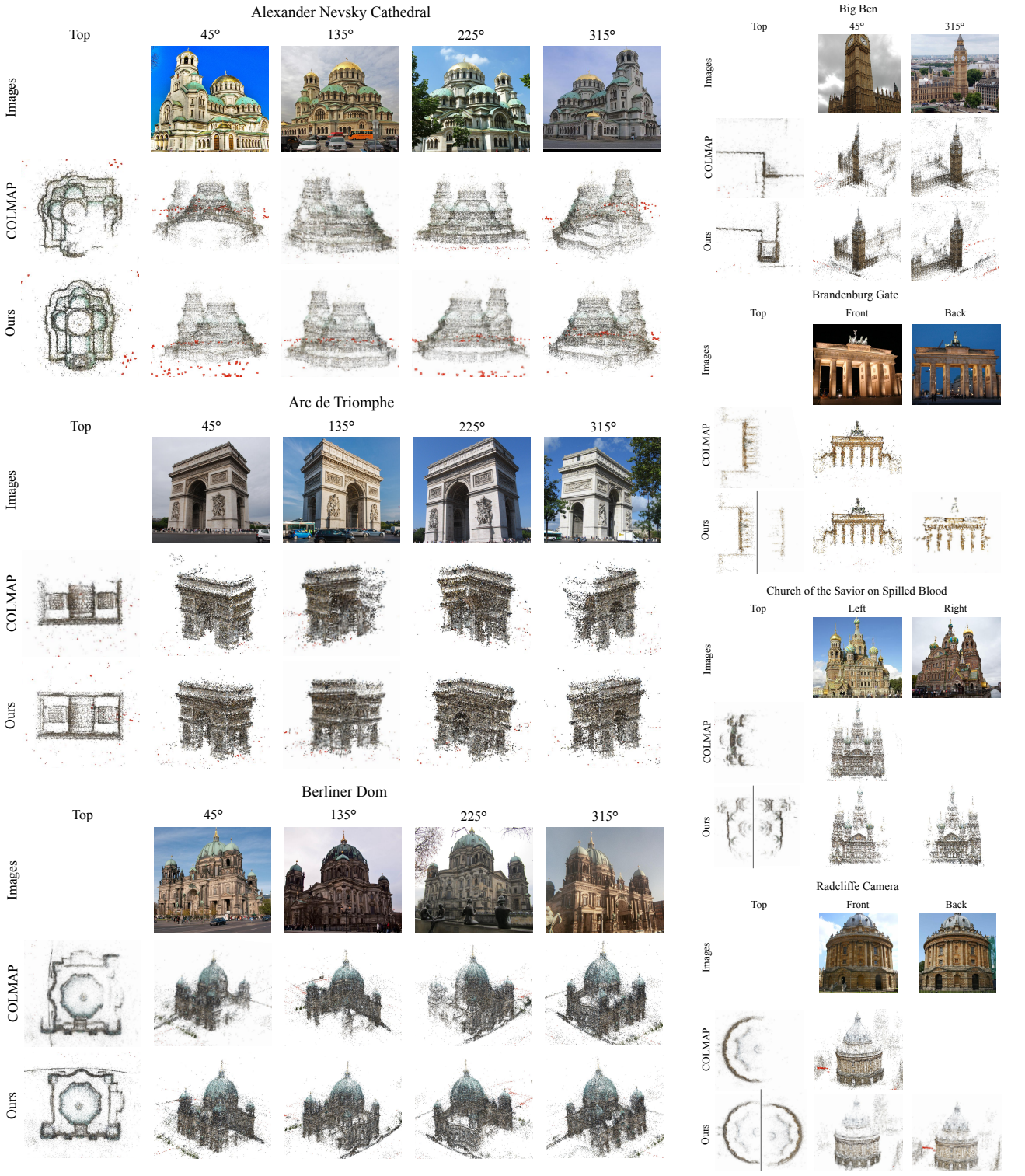

Figure 7: Visualization of Structure from Motion (SfM) disambiguation results from different viewpoints. We show a set of input RGB images at the top of each example scene, vanilla COLMAP reconstructions in the middle, and our method's disambiguated reconstructions at the bottom. For reconstructions where an angle is denoted, the 0° mark begins at the bottom of the birds-eye view and increases counterclockwise about the center of the image. Note that for some landmarks, the correct reconstruction is separated into two components when disambiguated due to a lack of camera views from sufficient viewpoints.

image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 3, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[9] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Correcting for duplicate scene structure in sparse 3d reconstruction. In *ECCV*, 2014. 6

[10] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 3

[11] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 3, 4

[12] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 3, 4

[13] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 1, 3, 6

[14] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 1, 3, 4

[15] Kyle Wilson and Noah Snavely. Network principles for sfm: Disambiguating repeated structures with local context. In *ICCV*, pages 513–520, 2013. 6

[16] Qingan Yan, Long Yang, Ling Zhang, and Chunxia Xiao. Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context. In *CVPR*, pages 3836–3844, 2017. 6