1 Supplementary Materials

In this section, we show that maximizing the conditional distribution of an update to a hypothesis is equivalent to maximizing the joint likelihood in Sec. 1.1. We evaluate ablations of our approach to validate the use of coordinate ascent vs gradient ascent and MST vs sequential loop in Tab. 1. To test the quality of our SLAM and SfM baselines, we also ran them with more image frames (narrower baseline) in Fig. 1. We show per-category evaluations to compare performance across seen and unseen categories of CO3D in Tab. 2. We provide a visualization of how to interpret the relative rotations in Fig. 2 and discuss the coordinate system in which we compute relative rotations in Fig. 3. We discuss the learned symmetry modes as well as some failure modes in Fig. 4. As a proof of concept, we use our energy-based predictor on a deformable object (cat) in Fig. 5. We include architecture diagrams for our energy-based pairwise pose predictor in Fig. 6 and the direct pose predictor baseline in Fig. 7. Finally, we show qualitative comparisons between our approach and the correspondence-based baselines on *randomly selected* sequences on both seen and unseen categories in Fig. 8 and Fig. 9 respectively.

1.1 Derivation of Conditional Distribution for Coordinate Ascent

Given our pairwise conditional probabilities, the joint distribution over a set of rotations can be computed as:

$$P\left(\{R_i\}_{i=1}^N \mid \{I_i\}_{i=1}^N\right) \propto P\left(\{R_i, I_i\}_{i=1}^N\right) = \alpha \exp\left(\sum_{(i,j)\in\mathcal{P}} f(R_{i\to j}, I_i, I_j)\right) \tag{1}$$

where $\mathcal{P} = \{(i,j) \mid (i,j) \in [N] \times [N], i \neq j\}$.

We are searching for the most likely set of rotations $\{R_1, \dots, R_N\}$ under this joint distribution given images $\{I_1, \dots, I_N\}$. For each iteration of coordinate ascent, we have our current most likely set of rotations $\{R_1, \dots, R_N\}$ and wish to update R_k. If we fix all $\{R_i\}_{i\neq k}$, the only terms in $\mathcal{P}$ that can change are the ones involving k, and the rest can be folded into a scalar constant. Thus, searching for the rotation R_k that maximizes the overall likelihood is equivalent to finding the most likely hypothesis under $P\left(R'_k \mid \{R_i\}_{i=1}^k, \{I_i\}_{i=1}^N\right)$:

$$\begin{aligned}\log P(R'_k \mid \{R_i\}_{i\neq k}, \{I_i\}_i) &= \sum_{(i,j)\in\mathcal{P}} f(R_{i\to j}, I_i, I_j) + C_1 &(2)\\ &= \sum_{i\neq k} \left(f(R_{i\to k'}, I_i, I_k) + f(R_{k'\to i}, I_k, I_i)\right) + C_2 &(3)\end{aligned}$$

This simplifies each iteration of coordinate ascent from a $\mathcal{O}(N^2)$ sum to a $\mathcal{O}(N)$ sum.

Acc @ 30°	3	5	10	20
Ours (Sequential)	0.50	0.48	0.42	0.39
Ours (MST)	0.52	0.50	0.47	0.43
Ours (Grad. Asc.)	0.52	0.51	0.49	0.47
Ours (Coord. Asc.)	**0.59**	**0.58**	**0.59**	**0.59**

Table 1: **Ablations on Seen Categories in CO3D (Random Sequence Subsampling)**. One way to convert a set of relative pose predictions to a coherent set of joint poses is by naively linking them together in a sequence (Sequential). We find that greedily linking them by constructing a maximum spanning tree (MST) performs slightly better since it incorporates that most confident relative rotation predictions. To make better use of our energy-based relative pose predictor, we tried directly running gradient ascent initialized from the MST solution and maximizing energy using ADAM (Grad. Asc.). Because the loss landscape is non-smooth, we observe that it does not deviate much from the MST solution. We found the scoring-based block coordinate ascent (Coord. Asc.) to be the most effective.

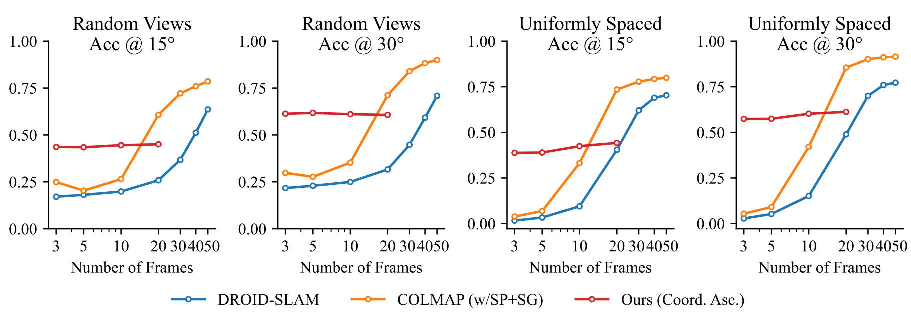

Fig. 1: **Evaluation of correspondence-based approaches on large image sets (on "Seen Categories" Split).** We evaluate the DROID-SLAM [4] and COLMAP (with SuperPoint features and SuperGlue matching) baselines on much longer image sequences (N=30, 40, 50). We verify that these approaches, which rely on correspondences between images, can achieve good performance when the cameras baselines are narrow. Nonetheless, the poor performance at $N < 20$ suggests that there is a rich space for improving camera pose estimation in the low data regime, which is the setting that we target in our work.

	Category	Acc. @ 30° (%) 3	5	10	20
Seen Categories	Apple	59	60	62	61
	Backpack	63	58	59	57
	Banana	67	54	63	55
	Baseballbat	100	67	70	73
	Baseballglove	48	56	56	55
	Bench	69	75	68	66
	Bicycle	62	61	63	62
	Bottle	59	57	60	60
	Bowl	80	75	77	80
	Broccoli	55	54	51	51
	Cake	46	47	47	54
	Car	67	71	70	62
	Carrot	60	64	63	65
	Cellphone	69	78	72	69
	Chair	53	55	55	56
	Cup	55	56	54	51
	Donut	52	44	51	51
	Hairdryer	58	56	58	54
	Handbag	66	63	62	61
	Hydrant	72	73	68	70
	Keyboard	72	73	74	74
	Laptop	88	87	89	89
	Microwave	56	65	55	58
	Motorcycle	59	60	62	61
	Mouse	68	70	69	67
	Orange	52	52	51	49
	Parkingmeter	22	27	23	22

	Category	Acc. @ 30° (%) 3	5	10	20
Seen Categories	Pizza	50	57	57	55
	Plant	46	47	49	51
	Stopsign	42	49	47	47
	Teddybear	47	52	49	48
	Toaster	76	75	71	73
	Toilet	76	80	75	77
	Toybus	63	70	72	71
	Toyplane	43	57	48	51
	Toytrain	81	73	75	75
	Toytruck	71	69	68	68
	Tv	78	83	87	86
	Umbrella	58	60	54	55
	Vase	58	55	55	51
	Wineglass	51	46	46	47
	Seen Mean	61	62	61	61
Unseen Categories	Ball	45	41	43	44
	Book	51	49	49	47
	Couch	42	58	39	35
	Frisbee	55	49	40	38
	Hotdog	58	61	50	49
	Kite	28	23	27	24
	Remote	64	58	65	66
	Sandwich	37	41	41	42
	Skateboard	56	64	64	65
	Suitcase	59	61	67	63
	Unseen Mean	49	51	48	48

Table 2: **Per-category Evaluation on CO3D with Random Sequence Sampling.** We find that rotationally symmetric objects (e.g. apple, orange, wineglass) tend to be challenging. We were surprised to find that bowls worked well, likely because the bowls in the CO3D dataset tend to have a lot of texture or even stickers. Objects with distinctive shapes (e.g. toilet, laptop) tend to be easier to orient. Note that some object categories have few instances for both training and testing (e.g. baseballbat, parkingmeter).

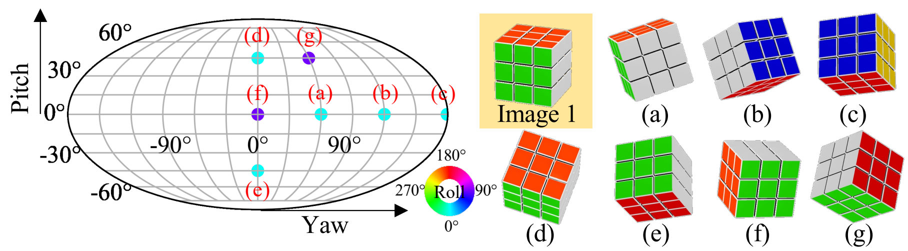

Fig. 2: **Interpreting Relative Rotations using a 2-Sphere.** Given "Image 1", we show how "Image 2" would have appeared given different relative rotations. (a), (b), and (c) show relative rotations with 60°, 120°, and 180° yaw respectively. (d) and (e) show relative rotations with 45° and -45° pitch respectively. (f) shows a relative rotation with just roll. (g) shows a relative rotation with all three components. We use a view-aligned coordinate system (See Fig. 3) when computing relative rotations. Inspired by [2], we visualize the **SO**(3) by projecting rotations onto a 2-sphere, with the x-axis representing yaw, y-axis representing pitch, and color representing roll.

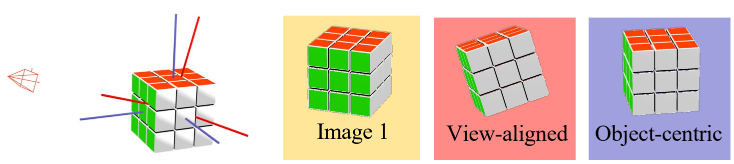

Fig. 3: **View-aligned vs Object-centric Coordinate System.** We compute relative rotations in a coordinate system (red axes on left) aligned with the camera (red wireframe on left). Relative rotations aligned to the camera viewpoint can always be computed without reasoning about the object's alignment with respect to the camera. While possibly more intuitive, relative rotations in the object coordinate system (blue axes on left) must be defined with respect to a canonical object pose and thus cannot be computed in general. On the right, we visualize a 60° yaw relative rotation from Image 1 in the view-aligned coordinate system (red) and object-centric coordinate system (blue).

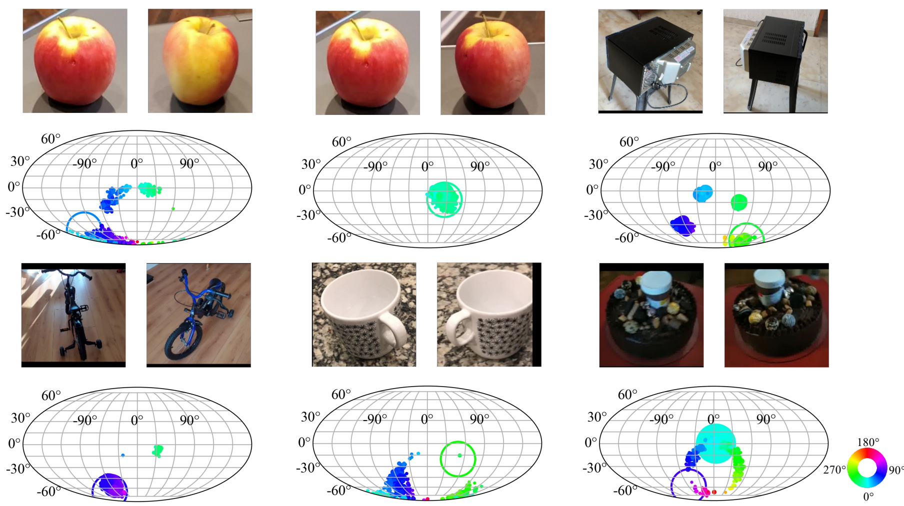

Fig. 4: **Learned Pairwise Distributions on Seen Categories (Test Set).** Here we visualize the learned pairwise distributions for various pairs of images. *Top left:* The images correspond to opposite sides of the apple, so the relative pose is ambiguous. Our approach predicts a rotationally symmetric band of possible rotations. *Top middle:* The images have sufficient overlap such that the relative rotation is unambiguous and our method predicts a single mode for the apples. *Top right:* For rectangular objects such as microwaves, our approach often predicts 4 modes corresponding to each of the 90 degree rotations. *Bottom left:* Our approach predicts 2 modes for the bicycle because the first viewpoint is challenging. *Bottom-middle:* Clashing foreground and background textures can be a challenge for our pairwise predictor. Even though the relative pose should be unambiguous, our method places low probability on the correct pose although it does recognize the rotational symmetry of the cup category. *Bottom-right:* Unusual object appearances is another failure mode of our method, which defaults to placing high probability mass on the identity matrix. Our method does recognize the rotational symmetry of the cake category.

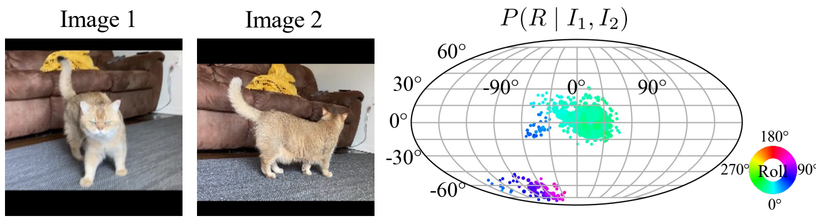

Fig. 5: **Deformable Objects.** Existing SfM and SLAM pipelines often make assumptions about rigidity or appearance constancy in order for bundle adjustment to converge. Our method has no such requirements and can be run even on deformable objects. While the ground truth poses for these images of a cat are unknown, the relative rotation of the camera w.r.t the cat is roughly -90 degrees yaw with negative pitch while the relative rotation of the camera w.r.t. the couch has no pitch or yaw but some roll in the clockwise direction (green). Although our training data does not include dynamic or deformable objects, our network outputs plausible modes.

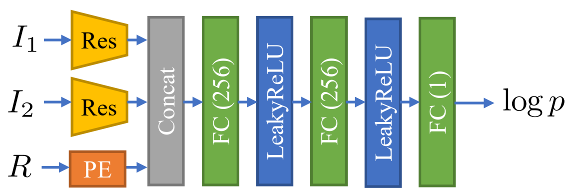

Fig. 6: **Architecture Diagram for our Pairwise Energy Predictor.** We use a ResNet-50 [1] with anti-aliasing [5] as our feature extractor. We directly apply positional encoding (8 bases) [3] to the elements of the 3×3 rotation matrix. We concatenate the image features and positionally encoded rotations into a feature vector ($2048 + 2048 + 2 \cdot 8 \cdot 9$), which we feed into an MLP that predicts energy (corresponding to unnormalized log probability).

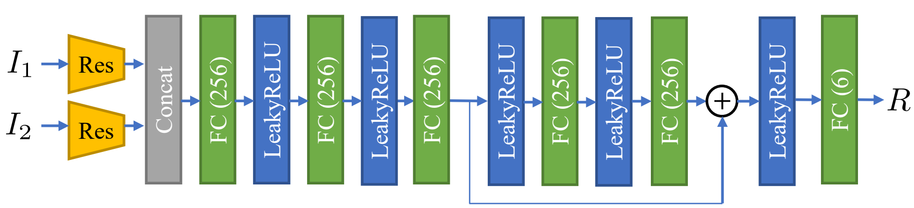

Fig. 7: **Architecture Diagram for our Direct Pairwise Rotation Predictor.** For the direct rotation regression baseline, we still input the concatenated image features ($2048 + 2048$). To make the baseline more competitive, we increase the capacity of the MLP to have 6 layers and a skip connection. The network predicts the 6-D rotation representation [6].

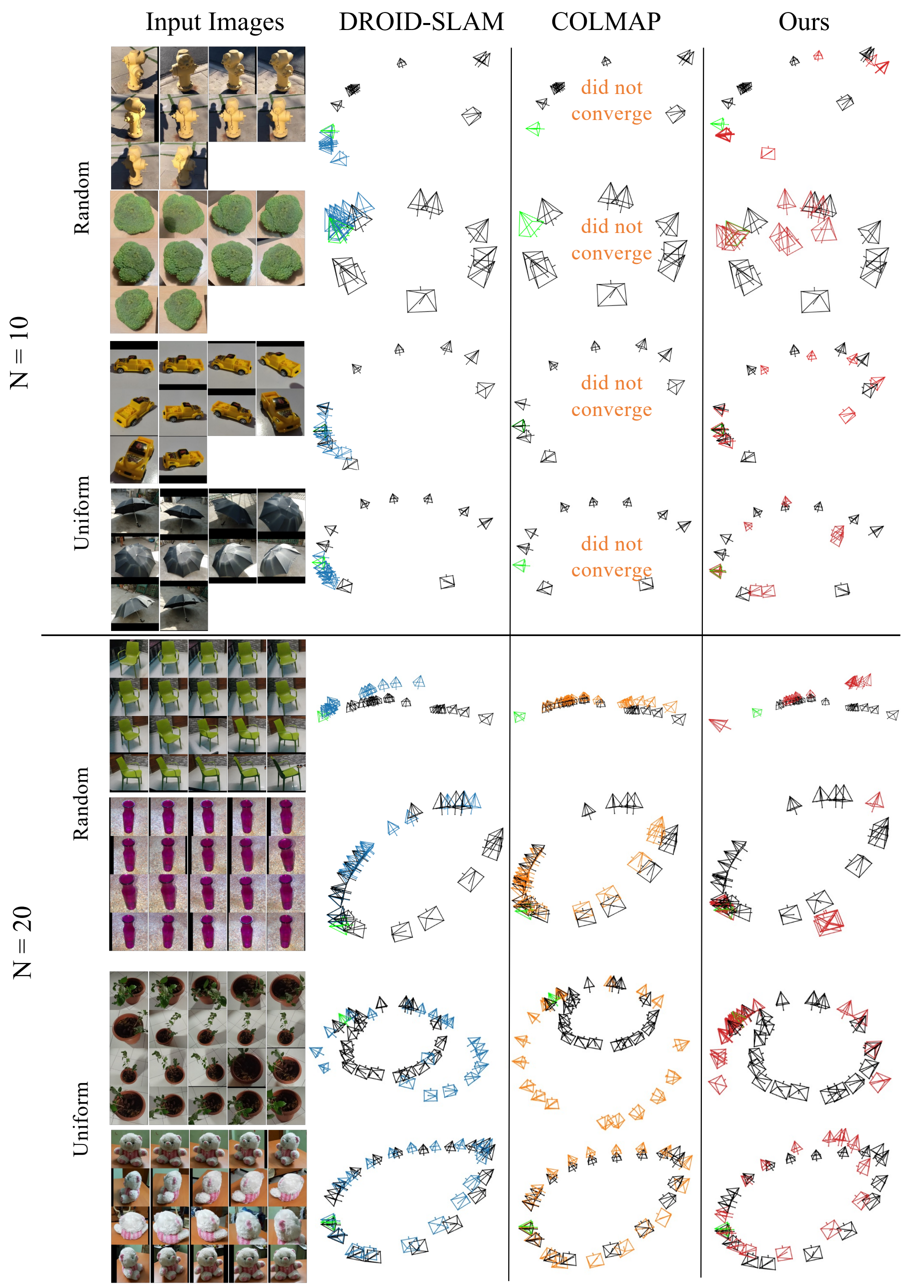

Fig. 8: **Randomly selected Qualitative Results for Seen Categories.**

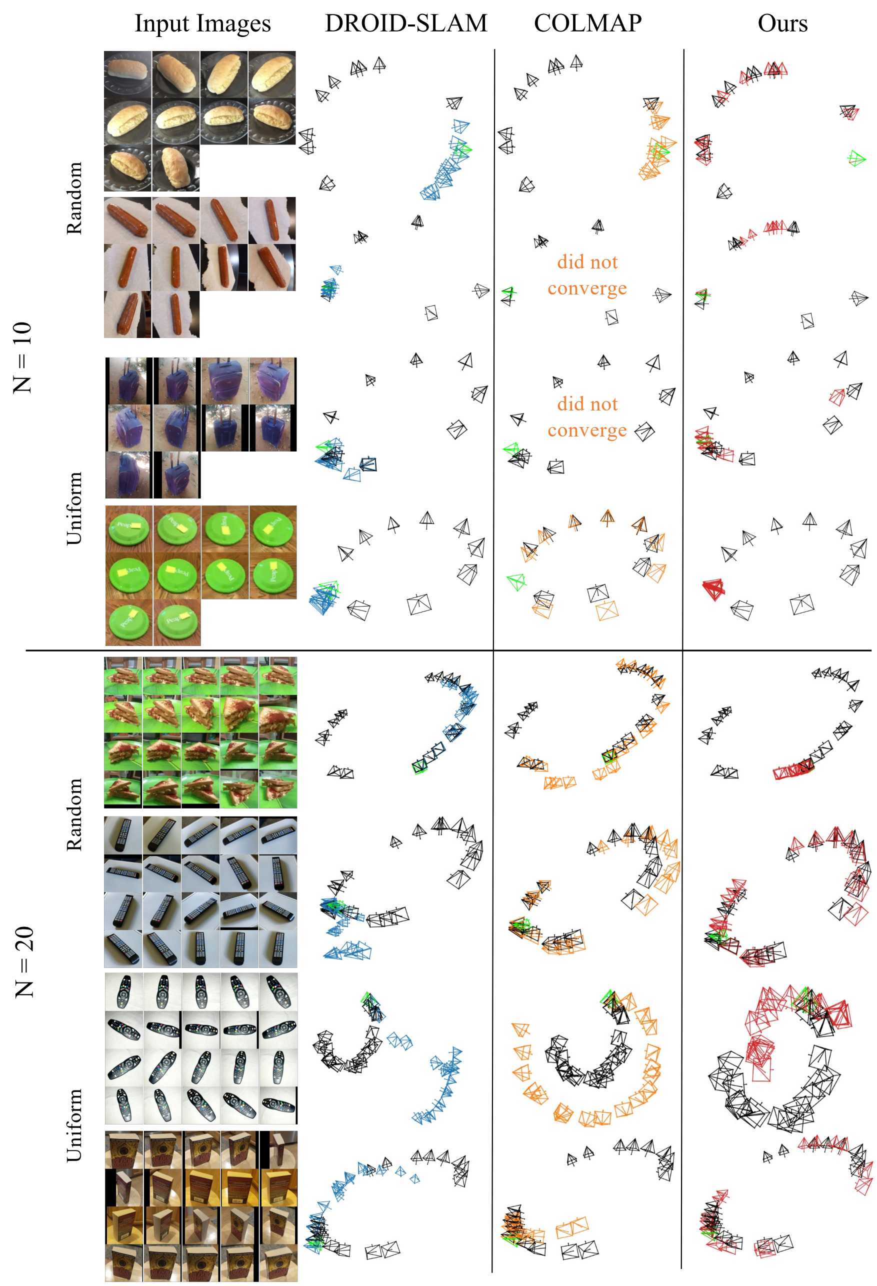

Fig. 9: **Randomly selected Qualitative Results for Unseen Categories.**

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016) 6
2. Murphy, K.A., Esteves, C., Jampani, V., Ramalingam, S., Makadia, A.: Implicit-PDF: Non-Parametric Representation of Probability Distributions on the Rotation Manifold. In: ICML (2021) 4
3. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. NeurIPS (2020) 6
4. Teed, Z., Deng, J.: DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. NeurIPS (2021) 2
5. Zhang, R.: Making Convolutional Networks Shift-Invariant Again. In: ICML (2019) 6
6. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR (2019) 6