RelPose++: Recovering 6D Poses from Sparse-view Observations

Amy Lin* Jason Y. Zhang* Deva Ramanan Shubham Tulsiani
Carnegie Mellon University
{amylin2,deva}@cs.cmu.edu, {jasonyzhang,shubhtuls}@cmu.edu
https://amyxlase.github.io/relpose-plus-plus

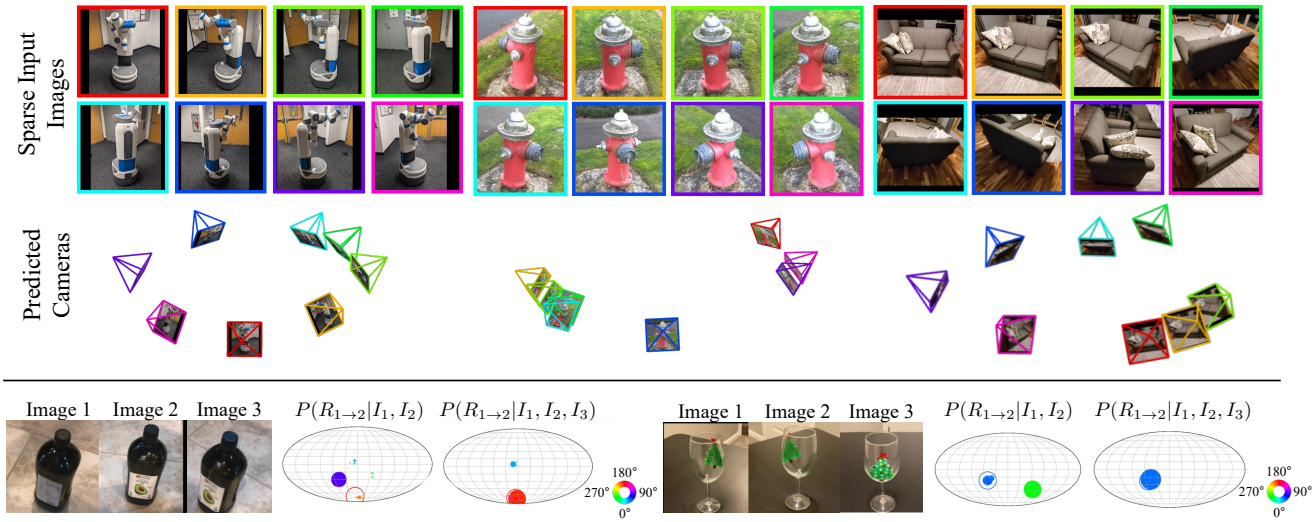

Figure 1: **Estimating 6D Camera Poses from Sparse Views.** We propose a framework *RelPose++* that, given a sparse set of input images, can infer the corresponding 6D camera rotations and translations (**top**: the cameras are colored from red to magenta based on the image index). *RelPose++* estimates a probability distribution over the relative rotations of the cameras corresponding to any 2 images, but can do so while incorporating multi-view cues. We find that the distribution improves given additional images as context (**bottom**).

Abstract

We address the task of estimating 6D camera poses from sparse-view image sets (2-8 images). This task is a vital pre-processing stage for nearly all contemporary (neural) reconstruction algorithms but remains challenging given sparse views, especially for objects with visual symmetries and texture-less surfaces. We build on the recent RelPose framework which learns a network that infers distributions over relative rotations over image pairs. We extend this approach in two key ways; first, we use attentional transformer layers to process multiple images jointly, since additional views of an object may resolve ambiguous symmetries in any given image pair (such as the handle of a mug that becomes visible in a third view). Second, we augment this network to also report camera translations by defining an appropriate coordinate system that decouples the ambiguity in rotation estimation from translation prediction. Our final system results in large improvements in 6D pose prediction over prior art on both seen and unseen object categories and also enables pose estimation and 3D reconstruction for in-the-wild objects.

1. Introduction

The longstanding task of recovering 3D from 2D images has witnessed rapid progress over the recent years, with neural field-based methods [28] enabling high-fidelity 3D capture of generic objects and scenes given dense multi-view observations. There has also been a growing interest in enabling similar reconstructions in *sparse-view* settings where only a few images of the underlying instance are available *e.g.* online marketplaces, or casual captures by everyday users. While several sparse-view reconstruction methods [54,57,59] have shown promising results, they critically rely on known (precise or approximate) 6D camera poses for this 3D inference and sidestep the question of how these 6D poses can be acquired in the first place. In

* denotes equal contribution.

this work, we develop a system that can help bridge this gap and robustly infer (coarse) 6D poses given a sparse set of images for a generic object *e.g.* a Fetch robot (Fig. 1).

The classical approach [40] to recovering camera poses from multiple images relies on bottom-up correspondences but is not robust in sparse-view settings with limited overlap across adjacent views. Our work instead adopts a top-down approach and builds on RelPose [56], which predicts distributions over pairwise relative rotations to then optimize multi-view consistent rotation hypotheses. While this optimization helps enforce multi-view consistency, RelPose's predicted distributions only consider pairs of images, which can be limiting. As an illustration, if we consider the first two images of the bottle shown in Fig. 1, we cannot narrow down the Y-axis rotation between the two (as the second label may be on either the side or the back). However, if we now additionally consider the third image, we can immediately understand that the rotation between the first two should be nearly 180 degrees!

We build on this insight in our proposed framework *RelPose++* and develop a method for jointly reasoning over multiple images for predicting pairwise relative distributions. Specifically, we incorporate a transformer-based module that leverages context across all input images to update the image-specific features subsequently used for relative rotation inference. *RelPose++* also goes beyond predicting only rotations and additionally infers the camera translation to yield 6D camera poses. A key hurdle is that the world coordinate frame used to define camera extrinsics can be arbitrary, and naive solutions to resolve this ambiguity (*e.g.* instantiating the first camera as the world origin) end up entangling predictions of camera translations with predictions of (relative) camera rotations. Instead, for roughly center-facing images, we define a world coordinate frame centered at the intersection of cameras' optical axes. We show that this helps decouple the tasks of rotation and translation prediction, and leads to clear empirical gains.

RelPose++ is trained on 41 categories from the CO3D dataset [36] and is able to recover 6D camera poses for objects in both seen and unseen categories given just a few images. We find that *RelPose++* yields over 25% improvement in the rotation prediction accuracy over the previous state-of-the-art sparse-view methods. We also evaluate the full 6D camera poses by measuring the accuracy of the predicted camera centers (while accounting for the similarity transform ambiguity), and demonstrate the benefits of prediction in our proposed coordinate system. We also formulate a metric that only evaluates the accuracy of camera translations (decoupled from the accuracy of predicted rotations), and hope that this would also be beneficial for analyzing subsequent approaches. Finally, we show that the 6D poses from *RelPose++* can be directly useful for downstream sparse-view 3D reconstruction methods.

2. Related Work

Pose Estimation Using Feature Correspondences. The classic SfM and SLAM pipelines for pose estimation from sets of images or video streams involve computing matches [23] between discriminative hand-crafted local features [2, 22]. These matches are used to estimate relative camera poses [21, 32], verified via RANSAC [10], and optimized via bundle adjustment [46]. Subsequent research has explored improving each of these components. Learned feature estimation [8] and feature matching [6, 20, 39] have improved robustness. This paradigm has been scaled by efficient parallelization [40, 41] and can even run in real-time for visual odometry [3, 29, 30]. While we consider a similar task of estimating camera poses given images, our approach differs fundamentally because we do not rely on bottom-up correspondences as they cannot be reliably computed given sparse views.

Single-view 3D Pose Estimation. In the extreme case of a single image, geometric cues are insufficient for reasoning about pose, so single-view 3D pose estimation approaches rely on learned data-driven priors. A significant challenge that arises in single-view 3D is that absolute pose must be defined with respect to a coordinate system. The typical solution is to assume a fixed set of categories (*e.g.* humans [16, 26] or ShapeNet objects [4]) with pre-defined canonical coordinate systems. Related to our approach are methods that specifically handle object symmetries, which can be done by predicting multiple hypotheses [25], parameters for the antipodal Bingham distribution [11, 35], or energy [31] (similar to us). These methods predict absolute pose which is not well-defined without a canonical pose.

Because absolute poses only make sense in the context of a canonical pose, some single-view pose estimation papers have explored learning the canonical pose of objects automatically [33, 44]. Other approaches bypass this issue by predicting poses conditioned on an input mesh [1, 34, 52, 55] or point-cloud [51]. In contrast, we resolve this issue by predicting relative poses given pairs of images.

Learned Multi-view Pose Estimation. Given more images, it is still possible to learn a data-driven prior rather than rely on geometric consistency cues alone. For instance, poses can directly be predicted using an RNN [45, 50] or auto-regressively [53] for SLAM applications. However, such approaches assume temporal locality not present in sparse-view images. Other approaches have incorporated category-specific priors, particularly for human pose [17, 18, 24, 48]. In contrast, our work focuses on learning *category-agnostic* priors that generalize beyond object categories seen at training.

Most related to our approach are methods that focus on sparse-view images. Such setups are more challenging since viewpoints have limited overlap. In the case of us-

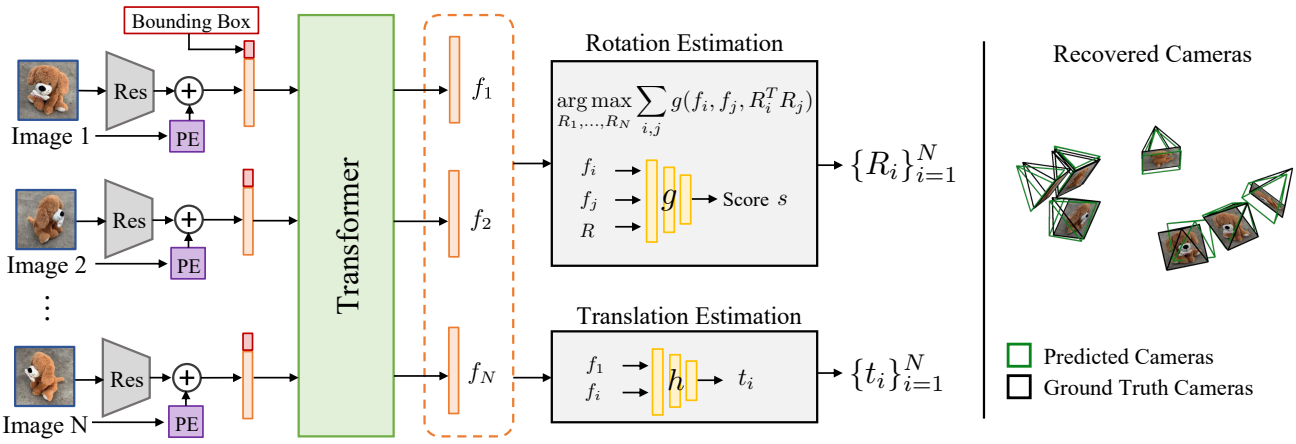

Figure 2: **Overview of RelPose++.** We present RelPose++, a method for sparse-view camera pose estimation. RelPose++ starts by extracting global image features using a ResNet 50. We positionally encode [49] the image index and concatenate bounding box parameters as input to a Transformer. After processing all image features jointly, we separately estimate rotations and translations. To handle ambiguities in pose, we model the distribution of rotations using an energy-based formulation, following [31, 56]. Because we predict the origin at the unique world coordinate closest to all optical axes, which is unambiguous (See Sec. 3.3 and Fig. 3), we can directly regress camera translation from the learned features. On the right, we visualize the recovered camera poses.

ing just 2 images for wide-baseline pose estimation, direct regression approaches [27, 37] typically do not model uncertainty or require distributions to be Gaussian [5]. Most similar to our work is the energy-based RelPose [56], which estimates distributions over relative rotations which can be composed together given more than 2 images. We build off of this energy-based framework and demonstrate significantly improved performance by incorporating multi-view context. Additionally, RelPose only predicts rotations whereas we estimate 6D pose. On the other hand, FORGE [15] jointly estimates 6D poses and 3D features by training on synthetic data. Finally, a recent concurrent work SparsePose [42] also leverages a transformer-based architecture for pose regression and subsequently learns a bundle-adjustment procedure to further refine these predictions. We view the learned refinement as complementary to our approach as it can improve any initial estimates, but demonstrate that our approach significantly improves over direct regression for initial pose prediction.

3. Method

Given a set of N (roughly center-facing) input images $\{I_1, ..., I_N\}$ of a generic object, we wish to recover consistent 6-DoF camera poses for each image *i.e.* $\{(R_1, \mathbf{t}_1), ..., (R_N, \mathbf{t}_N)\}$, where R_i and $\mathbf{t}_i$ correspond to the rotation and translation for the i^{th} camera viewpoint.

To estimate the camera rotations, we adopt the framework proposed by RelPose [56], where a consistent set of rotations can be obtained given pairwise relative rotation distributions (Sec. 3.1). However, unlike RelPose which predicts these distributions using only two images, we incorporate a transformer-based module to allow the pairwise predicted distributions to capture multi-view cues (Sec. 3.2). We then extend this multi-view reasoning module also to infer the translations associated with the cameras, while defining a world-coordinate system that helps reduce prediction ambiguity (Sec. 3.3).

3.1. Global Rotations from Pairwise Distributions

We build on RelPose [56] for inferring consistent global rotations given a set of input images and briefly summarize the key components here. As absolute camera rotation prediction is ill-posed given the world-coordinate frame ambiguity, RelPose infers pairwise relative rotations and then obtains a consistent set of global rotations. Using an energy-based model, it first approximates the (un-normalized) log-likelihood of the pairwise relative rotations given image features f_i and f_j with an MLP $g_\theta(f_i, f_j, R_{i\to j})$ which we treat as a negative energy or score.

Given the inferred distributions over pairwise relative rotations, RelPose casts the problem of finding global rotations as that of a mode-seeking optimization. Specifically, using a greedy initialization followed by block coordinate ascent, it recovers a set of global rotations that maximize the sum of relative rotation scores:

$$\{R_1, \ldots, R_N\} = \underset{\{R_i\}_{i=1}^N}{\operatorname{argmax}} \sum_{i,j} g_\theta(f_i, f_j, R_i^\top R_j) \qquad (1)$$

In RelPose, the image features are extracted using a per-frame ResNet-50 [14] encoder: $f_i = \varepsilon_\phi(I_i)$.

3.2. Multi-view Cues for Pairwise Distributions

Following RelPose, we similarly model the distribution of pairwise relative rotations using an energy-based model (eq. (1)). However, instead of only relying on the images I_i and I_j to obtain the corresponding features f_i and f_j, we propose a transformer-based module that allows for these features to depend on *other* images in the multi-view set.

Multi-view Conditioned Image Features. As illustrated in Fig. 2, we first use a ResNet [14] to extract per-image features. We also add an ID-specific encoding to the ResNet features and concatenate an embedding of the bounding box used to obtain the input crop from the larger image (as it may be informative about the scene scale when inferring translation). Unlike RelPose which then directly feeds these image-specific features as input to the energy prediction module, we use a transformer to update these features in context of the other images.

We use a transformer encoder with 8 layers of multi-headed self-attention blocks, comparable to the encoder used in ViT [9]. We denote this combination of the feature extractor and transformer as a scene encoder $\mathcal{E}_\phi$, which given N input images $\{I_n\}$ outputs multi-view conditioned features $\{f_n\}$ corresponding to each image:

$$f_i = \mathcal{E}^i_\phi(I_1, \dots I_N), \quad \forall i \in \{1 \dots N\}$$

Learning Objective. Given a dataset with posed multi-view images of diverse objects, we jointly train the scene encoder $\mathcal{E}_\phi$ and the pairwise energy-based model g_θ by simply minimizing the negative log-likelihood (NLL) of the true (relative) rotations [31, 56]. In particular, we randomly sample $N \in [2, 8]$ images for a training object, and minimize the NLL of the true relative rotations $R^{gt}_{i\to j}$ under our predicted distribution:

$$L_{\text{rot}} = \sum_{i,j} -\log \frac{\exp g_\theta(f_i, f_j, R^{gt}_{i\to j})}{\sum_{R'} \exp g_\theta(f_i, f_j, R')} \qquad (2)$$

3.3. Predicting Camera Translations

Using the multi-view aware image features f_i, we can directly predict the per-image camera translation $\mathbf{t}_i = h_\psi(f_i)$. However, a central hurdle to learning such prediction is the inherent ambiguity in the world coordinate system. Specifically, the 'ground-truth' cameras obtained from SfM are meaningful only up to an arbitrary similarity transform [13]), and training our network to predict these can lead to incoherent training targets across each sequence. We therefore first need to define a consistent coordinate frame across training instances, so that the networks can learn to make meaningful predictions.

Geometric Interpretation of Camera Translation. Recall that the camera extrinsics $(R_i, \mathbf{t}_i)$ define a transformation of a point $\mathbf{x}^w$ in world frame to camera frame $\mathbf{x}^c_i = R_i\mathbf{x}^w + \mathbf{t}_i$. The translation $\mathbf{t}_i$ is therefore the location of the world origin in each camera's coordinate frame (and not the location of the camera in the world frame!). We can also see that an arbitrary rotation of the world coordinate system ($\bar{\mathbf{x}}^w = \Delta R\ \mathbf{x}^w$), does not affect the per-camera translations and that only the location of the chosen world origin (and the scaling) are relevant factors. To define a consistent coordinate frame for predicting rotations, we must therefore decide where to place the world origin and how to choose an appropriate scale.

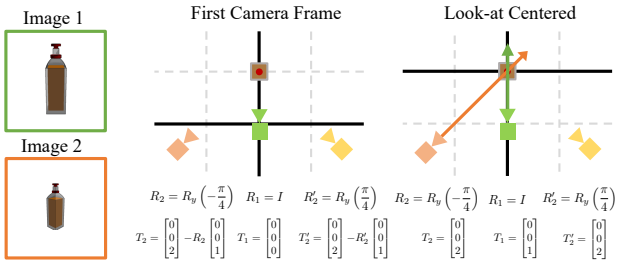

Figure 3: **Coordinate Systems for Estimating Camera Translation.** Given two images, consider the task of estimating their 6D poses, i.e., the R and T that transform points from the world frame to each camera's frame (**Left**). In typical SLAM setups, the world frame is centered at the first camera, but this implies the target camera translation T_2 depends on the target rotation R_2 (**Middle**). For symmetric objects where R_2 may be ambiguous, this may lead to unstable predictions for translation. Instead, for roughly center-facing cameras, a better solution is to set the world origin at the unique point closest to the optical axes of all cameras (**Right**). This helps decouple the task of predicting camera translations from rotations.

Look-at Centered Coordinate System. One convenient choice, often also adopted by SfM/SLAM approaches [7, 40], is to define the world coordinate system as centered on the first camera (denoted as 'First Camera Frame'). Unfortunately, the per-camera translations in this coordinate frame entangle the relative rotations between cameras (as $\mathbf{t}_i$ is the location of the first camera in the i^{th} camera's frame). As illustrated in Fig. 3, ambiguity in estimating this relative transformation for (*e.g.*, symmetric) objects can lead to uncertainty in the translation prediction.

Instead, we argue that for roughly center-facing captures, one should define the unique point closest to the optical axes of the input cameras as the world origin. Intuitively, this is akin to setting the 'object center' as the world origin, and the translation then simply corresponds to the inference of where the object is in the camera frame (and this remains invariant even if one is unsure of camera rotation as illustrated in Fig. 3). However, instead of relying on a semantically defined 'object center' which may be ambiguous given partial observations, the closest approach point across optical axes

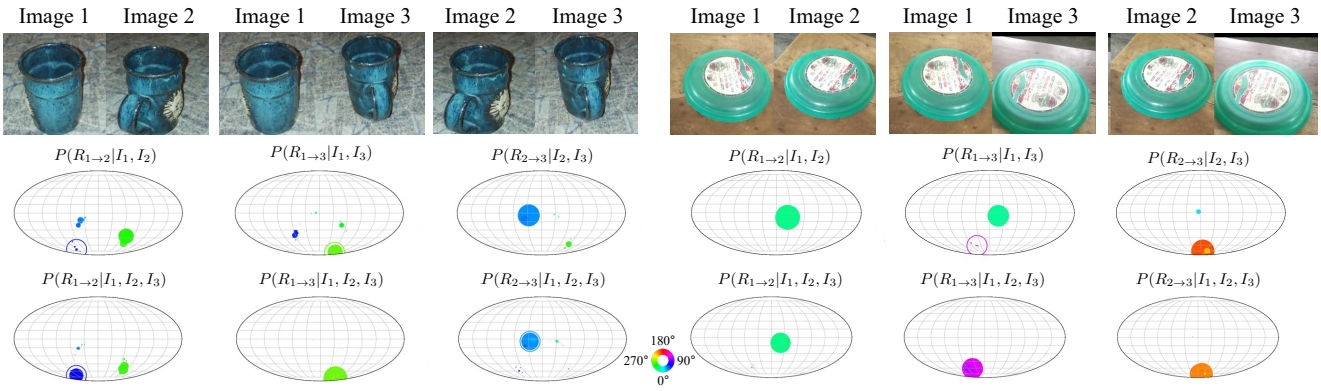

Figure 4: **Resolving Pose Ambiguity with More Images.** The relative rotation between only two views may be ambiguous for highly symmetric objects such as cups, frisbees, and apples. Often, seeing a third view will provide enough additional context to the scene to determine the correct relative rotation. When images are shown to the model in three separate pairs, as denoted by $P(R_{i\to j}|I_i, I_j)$, the output probability distribution may have more than one mode due to the symmetry of the object, but when shown all three images together to predict $P(R_{i\to j}|I_1, I_2, I_3)$, the model has a significantly more confident prediction. Following [56], we visualize distributions over relative rotations by projecting the rotation matrix such that the x-axis represents the yaw, the y-axis represents the pitch, and the color represents the roll. The size of each circle corresponds to probability, and rotations with negligible probability are filtered. The ground truth rotation is denoted by the unfilled circle.

is a well-defined geometric proxy. Finally, to resolve scale ambiguity, we assume that the first camera is a unit distance away from this point.

Putting it Together. In addition to the energy-based predictor (Eq. 1), we also train a translation prediction module that infers the per-camera $\mathbf{t}_i = h_\psi(f_1, f_i)$ given the multi-view features. Because we normalize the scene such that $\|t_1\| = 1$, we provide h_ψ with the first image feature f_1. To define the target translations for training, we use the ground-truth cameras (SfM) $\{(R_i, \mathbf{t}_i)\}$ to first identify the point $\mathbf{c}$ closest to all the optical axes. We can then transform the world coordinate to be centered at $\mathbf{c}$, thus obtaining the target translations as $\bar{\mathbf{t}}_i = s(\mathbf{t}_i - R_i\mathbf{c})$, where the scale s ensures a unit norm for $\bar{\mathbf{t}}_1$. For this training, we simply use an L1 loss between the target and predicted translations:

$$L_{\text{trans}} = \|h_\psi(f_i, f_1) - \bar{\mathbf{t}}_i\|_1 \tag{3}$$

Together with the optimized global rotations, these predicted translations yield 6-DoF cameras given a sparse set of input images at inference.

4. Evaluation

4.1. Experimental Setup

Dataset. We train and test our models on the CO3D [36] (v2) dataset, which consists of turntable-style video sequences across 51 object categories. Each video sequence is associated with ground truth camera poses acquired using COLMAP [40]. Following [56], we train on 41 object categories and hold out the same 10 object categories to evaluate generalization. After filtering for the camera pose score, we train on a total of 22,375 sequences with 2,212,952 images.

	# of Images	2	3	4	5	6	7	8
Seen Cate.	COLMAP (SP+SG)	30.7	28.4	26.5	26.8	27.0	28.1	30.6
	RelPose	56.0	56.5	57.0	57.2	57.2	57.3	57.2
	Pose Regression	49.1	50.7	53.0	54.6	55.7	56.1	56.5
	Ours (N=2)	**81.8**	82.3	82.7	83.2	83.3	83.5	83.6
	Ours (Full)	**81.8**	**82.8**	**84.1**	**84.7**	**84.9**	**85.3**	**85.5**
Unseen Cate.	COLMAP (SP+SG)	34.5	31.8	31.0	31.7	32.7	35.0	38.5
	RelPose	48.6	47.5	48.1	48.3	48.4	48.4	48.3
	Pose Regression	42.7	43.8	46.3	47.7	48.4	48.9	48.9
	Ours (N=2)	**69.8**	69.6	70.1	69.8	70.4	70.5	71.2
	Ours (Full)	**69.8**	**71.1**	**71.9**	**72.8**	**73.8**	**74.4**	**74.9**

Table 1: **Joint Rotation Accuracy.** We measure the relative angular error between pairs of relative predicted and ground truth rotations. We report the proportion of angular errors within 15 degrees. We find that our method consistently outperforms COLMAP and RelPose. The Pose Regression baseline which directly predicts poses struggles, suggesting that it is important to model uncertainty. Our performance starts out similar to the ablation that only looks at 2 images (N=2) but quickly outperforms it, suggesting that the context of additional images is helpful.

Task and Metrics. We randomly sample $2 \leq N \leq 8$ center-cropped images $\{I_n\}$ from each test sequence. Given these as input, each approach then infers a set of global 6-DoF camera poses $\{R_i, \mathbf{t}_i\}$ corresponding to each input image. To evaluate these predictions, we report accuracy under various complementary metrics, all of which are invariant under global similarity transforms for the prediction/ground-truth cameras. To reduce variance in metrics, we re-sample the N images from each test sequence 5 times and compute the mean.

Rotation Accuracy. We evaluate relative rotation error between every pair of predicted and ground truth rotations.

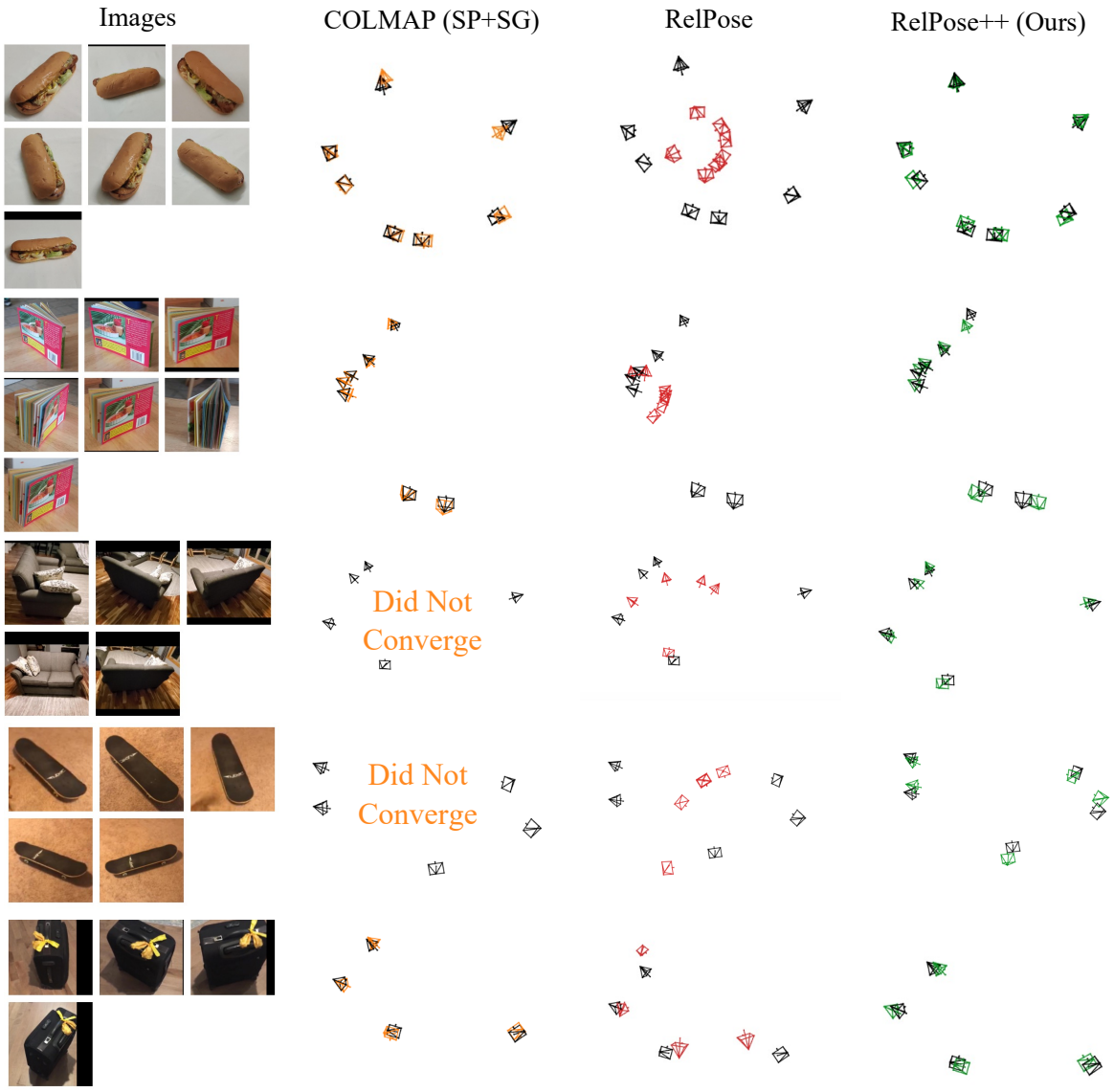

Figure 5: **Qualitative Results of Recovered Camera Trajectories.** We compare our approach with COLMAP (with SuperPoint features and SuperGlue matching) and RelPose. Since RelPose does not predict translations, we set the translations to be unit distance from the scene center. We visualize the predicted camera trajectories in color and the ground truth in black, aligned using a Procrustes optimal alignment on the camera centers. We find that COLMAP is accurate but brittle, converging only occasionally when the object has highly discriminative features and sufficient overlap between images. RelPose, while mostly accurate, usually makes 1-2 mistakes per sequence which causes misalignment. We find that our method consistently outperforms the baselines.

Following [42, 56], we report the proportion of pose errors less than 15 degrees.

Camera Center Accuracy. Following standard benchmarks in SLAM [43] that evaluated recovered poses using camera localization error, we measure the accuracy of the predicted camera centers. However, as the predicted centers $\mathbf{c}_i = -R_i\mathbf{t}_i$ may be in a different coordinate system from the SfM camera centers $\mathbf{c}_i^{gt}$, we first compute the optimal similarity transform to align the predicted centers with the ground-truth [47]. Following [42], we then report the proportion of predicted camera centers within 20% of the scale of the scene in Tab. 2, where the scale is defined as the distance from the centroid of the ground truth camera centers to the furthest camera center.

Camera Translation Accuracy. As the camera center depends on both the predicted rotation and translation, it makes it difficult to disentangle the accuracy of the inferred translations. For example, for a symmetric object, the translation may be accurate (*e.g.* object is 2m away from the camera along Z-axis) even if the rotation is incorrect. We therefore also propose to evaluate the accuracy of predicted translations in a similar way–measuring what fraction of

	# of Images	2	3	4	5	6	7	8
Seen Cate.	COLMAP (SP+SG)	-	35.8	26.1	21.6	18.9	18.3	19.2
	Pose Reg. (First Fr.)	-	87.6	81.2	77.6	75.8	74.5	73.6
	Pose Reg. (Our Fr.)	-	90.3	84.6	81.5	80.0	78.5	77.7
	Constant	-	91.3	87.2	84.7	83.1	82.0	81.1
	Ours	-	**92.3**	**89.1**	**87.5**	**86.3**	**85.9**	**85.5**
Unseen Cate.	COLMAP (SP+SG)	-	37.9	29.3	24.7	23.1	23.5	25.3
	Pose Reg. (First Fr.)	-	78.8	71.4	66.3	63.6	61.8	60.4
	Pose Reg. (Our Fr.)	-	**82.8**	74.0	70.0	67.8	65.8	65.3
	Constant	-	81.4	73.7	69.5	67.3	65.2	63.8
	Ours	-	82.5	**75.6**	**71.9**	**69.9**	**68.5**	**67.5**

Table 2: **Camera Center Accuracy.** We report the proportion of camera centers that are within 20% of the scene scale to the ground truth camera centers. We align the predicted and ground truth camera centers using an optimal similarity transform (hence all methods are at 100% for N=2). The COLMAP baseline does not perform well with sparse views. The pose regression model directly regresses rotations and translations without modeling uncertainty. We find that the performance with translations predicted in our look-at-centered coordinate frame is better than in the first camera frame. The Constant translation baseline uses our predicted rotations but a translation of [0, 0, 1] and has comparable performance. Our proposed method using energy-based rotation estimation and regressing translations in the look-at-centered coordinate frame performs the best.

	# of Images	2	3	4	5	6	7	8
Seen	COLMAP (SP+SG)	34.8	32.0	24.7	21.6	19.8	19.6	21.6
	Constant	96.3	93.6	92.0	91.0	90.4	89.9	89.5
	Pose Reg (First Fr.)	95.6	92.9	91.7	91.2	90.7	90.3	90.2
	Ours	**98.8**	**97.9**	**97.6**	**97.4**	**97.3**	**97.1**	**96.9**
Unseen	COLMAP (SP+SG)	34.1	32.0	25.2	21.7	21.3	22.4	23.5
	Constant	93.0	88.6	86.4	85.5	84.6	83.8	83.4
	Pose Reg (First Fr.)	91.1	85.4	84.1	83.6	83.0	82.6	81.9
	Ours	**96.0**	**93.8**	**93.1**	**92.3**	**92.2**	**91.8**	**91.6**

Table 3: **Camera Translation Accuracy.** Because the camera center depends on both the predicted rotation and translation, evaluating with the camera center cannot infer the accuracy of the translations alone. Thus, we propose to also evaluate the accuracy of the predicted translations alone. Since the translations predicted by each method are in an arbitrary coordinate system, we apply an optimal similarity transform. We find that our method predicts better quality translations. We note that the translation regression in the first camera frame generalizes poorly compared to the Constant translation baseline and our method, likely because the predicted translation is also a function of the predicted rotation, which may have additional errors (see Fig. 3).

these are within 20% of the scale of the scene. However, to account for the ambiguity in the scaling and world origin, we first compute the optimal $s, \mathbf{t}$ to transform the predicted translations $\{\hat{\mathbf{t}}_i = s(\mathbf{t}_i - R_i\mathbf{t})\}$ to best align with the SfM translations $\{\mathbf{t}_i^{gt}\}$ (see appendix for details). We then measure the accuracy for $\hat{\mathbf{t}}_i$ against the SfM ground-truth $\{\mathbf{t}_i^{gt}\}$.

Baselines. We compare our approach with state-of-the-art correspondence-based and learning-based methods:

COLMAP (SP+SG) [40, 41]. This represents a state-of-the-art SFM pipeline (COLMAP) that uses SuperPoint features [8] with SuperGlue matching [39]. We use the implementation provided by HLOC [38].

RelPose [56]. We evaluate RelPose, which also uses a pairwise energy-based scoring network. As this only predicts rotations, we exclude it from translation evaluation.

Variants. We also report comparisons to variants of our approach to highlight the benefits of the energy-based prediction, multi-view reasoning, and proposed translation coordinate frame.

Pose Regression. This corresponds to a regression approach that uses our ResNet and Transformer architecture to directly predict the global rotations (assuming the first camera has identity rotation) and translations. This rotation prediction is analogous to the initial regressor in a concurrent work SparsePose [42] (unfortunately, due to licensing issues, we were unable to obtain code/models for direct comparison). We consider variants that regress translations using the first camera frame and our look-at-centered coordinate frame.

Ours (N=2). This represents a variant that only has access to 2 images at a time when inferring pairwise rotation distributions. While similar to RelPose, it helps disambiguate the benefits of our transformer-based architecture.

First-frame Centered Regression. Instead of using the Look-at centered world frame, this variant defines camera translations using the first camera as world origin (while using the same scaling as the Look-at centered system).

Constant Translation. Given the center-facing cameras and unit scale normalization, we also evaluate the prediction of a constant $[0, 0, 1]$ translation.

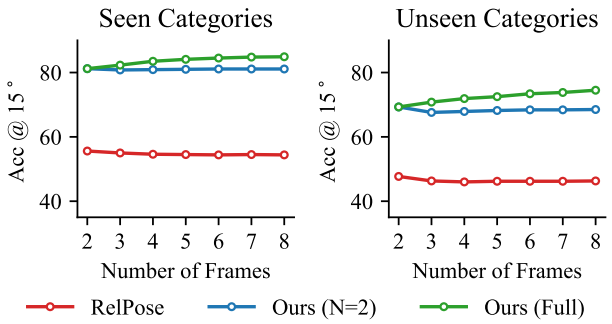

Figure 6: **Pairwise Rotation Accuracy.** We report pairwise rotation accuracy as the percent of predicted relative rotations within 15 degrees of the true relative rotation for every pair of cameras in the scene. Our (full) model exhibits increased performance with the number of frames. The ability to show our model more than just two images at a time provides valuable scene context that can help with resolving pose ambiguity caused by object symmetry.

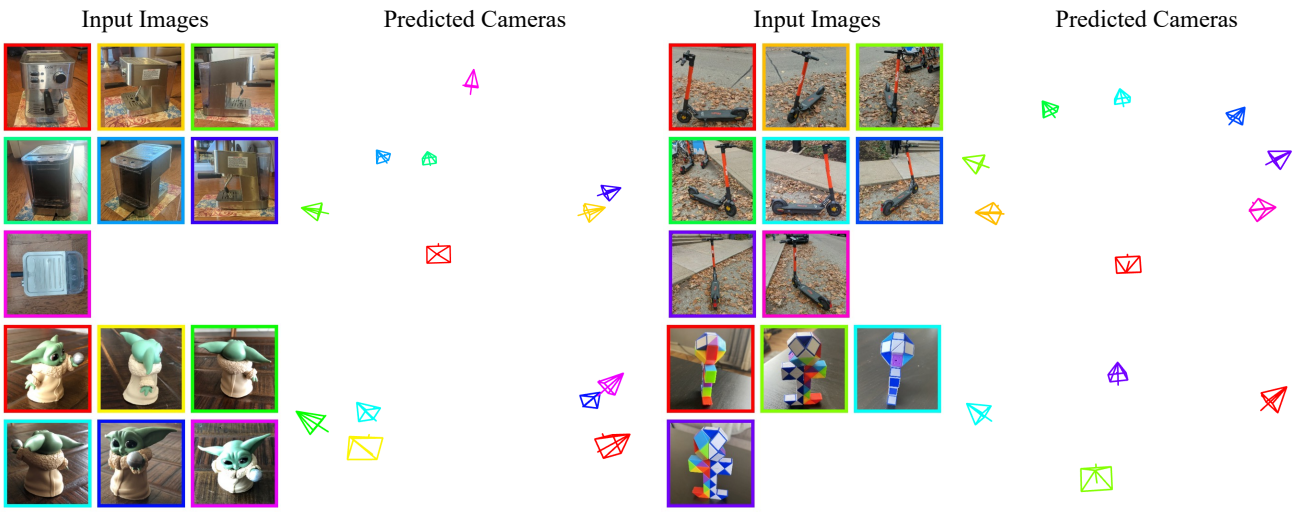

Figure 7: **Recovered Camera Poses from In-the-Wild Images.** We find that RelPose++ generalizes well to images outside of the distribution of CO3D object categories. Here, we demonstrate that RelPose++ can recover accurate camera poses even for self-captures of an espresso machine, scooter, action figure, and Rubik's snake.

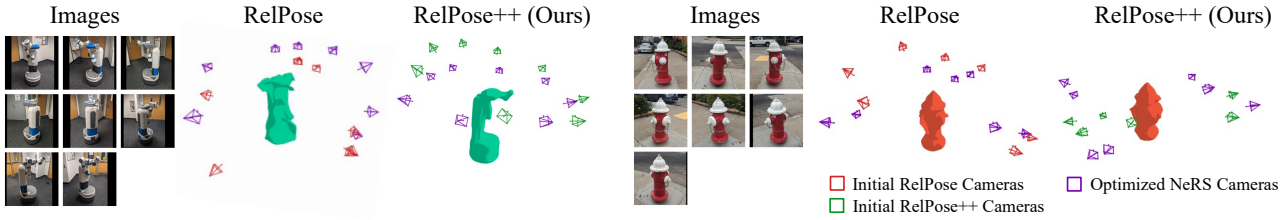

Figure 8: **Sparse-view 3D Reconstruction using NeRS.** We find that the camera poses estimated by our method are sufficient as initialization for 3D reconstruction. We compare our recovered cameras (green) with RelPose cameras (red) as initialization to NeRS. NeRS jointly optimizes these cameras and shape. We visualize the cameras at the end of the NeRS optimization in purple. We find that our cameras enable higher-fidelity 3D reconstruction, particularly for the robot.

4.2. Quantitative Results

Accuracy of Recovered 6D Poses. We evaluate rotation accuracy in Tab. 1. We find that our approach significantly outperforms COLMAP and RelPose. COLMAP does not perform well in sparse-view settings because wide baselines do not provide enough overlap to compute useful correspondences. We find that the Pose Regression baseline performs poorly, suggesting that modeling uncertainty is important for sparse-view settings. Finally, we find that our model starts out at a similar performance to the N=2 model, but quickly outperforms it for larger N, suggesting that the image context is important.

We evaluate the camera center and camera translation accuracies in in Tabs. 2 and 3. COLMAP performs poorly since it often fails to converge. We find that predicting constant translation ([0, 0, 1]) is a strong baseline, even outperforming the First Camera Frame Regression baseline. This suggests that predicting translation in a Look-at Centered coordinate system is a useful inductive bias. We find that the First Camera Frame Regression has the worst generalization to unseen object categories (see Tab. 3). This makes sense because the predicted translation must also account for any errors in the predicted rotation, which likely occur in a different distribution than seen for training categories.

Analyzing Pairwise Rotation Distributions. In the previous evaluations, we aim to recover a globally consistent set of N cameras given N images. To evaluate the importance of context, we also evaluate the rotation accuracy for pairs of images, possibly conditioned on more than 2 images. We evaluate the proportion of pairwise relative poses that are within 15 degrees of the ground truth in Fig. 6. We find that the transformer-based architectures (Ours, Pose Regression) are able to make use of the context of additional images to improve pairwise accuracy.

4.3. Qualitative Results

Visualization for Co3D Predictions. We compared recovered camera poses from sparse-view images using our method with COLMAP (with SuperPoint/SuperGlue) and RelPose in Fig. 5. We find that our method is able to consistently recover more accurate cameras than RelPose. While COLMAP recovers highly accurate trajectories when it succeeds, it usually fails to converge for sparse images.

We also visualize the effect of increasing image context

on pairwise rotation distributions in Fig. 4. Given just two images, the relative pose is often ambiguous, but we find that this ambiguity can be resolved by conditioning on more images using our transformer.

In-the-wild Generalization and 3D Reconstruction. Finally, we demonstrate that RelPose++ can be effectively demonstrated on in-the-wild images that are not categories from CO3D in Fig. 7. These cameras are even sufficient to enable 3D reconstruction, as shown in Fig. 8. We use NeRS [57], a representative sparse-view 3D reconstruction method. Please refer to the supplement for additional qualitative results.

5. Discussion

We presented *RelPose++*, a system for inferring a consistent set of 6D poses (rotations and translations) given a sparse set of input views. While it can robustly infer camera poses, these are not as precise as ones obtained from classical methods, and can be improved further via refinement [19,42]. Secondly, while the energy-based models can efficiently capture uncertainty, they are inefficient to sample from and are limited to pairwise distributions, and it may be possible to instead leverage diffusion models to overcome these limitations. Lastly, while we demonstrated that our estimated poses can enable downstream 3D reconstruction, it would be beneficial to develop unified approaches that jointly tackle the tasks of reconstruction and pose inference.

Acknowledgements: We would like to thank Samarth Sinha for useful discussion and thank Sudeep Dasari and Helen Jiang for their feedback on drafts of the paper. This work was supported in part by the NSF GFRP (Grant No. DGE1745016), Singapore DSTA, a CISCO gift award, and CMU Argo AI Center for Autonomous Vehicle Research.

References

[1] Wang Angtian, Adam Kortylewski, and Alan Yuille. NeMo: Neural Mesh Models of Contrastive Features for Robust 3D Pose Estimation. In *ICLR*, 2021. 2

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006. 2

[3] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *T-RO*, 2021. 2

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[5] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-Baseline Relative Camera Pose Estimation with Directional Learning. In *CVPR*, 2021. 3

[6] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal Correspondence Network. *NeurIPS*, 2016. 2

[7] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time Single Camera SLAM. *TPAMI*, 2007. 4

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised Interest Point Detection and Description. In *CVPR-W*, 2018. 2, 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 4

[10] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981. 2

[11] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep Orientation Uncertainty Learning Based on a Bingham Loss. In *ICLR*, 2019. 2

[12] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation Averaging. *IJCV*, 2013. 11

[13] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 3, 4, 11

[15] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-View Object Reconstruction with Unknown Categories and Camera Poses. *ArXiv*, 2212.04492, 2022. 3

[16] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end Recovery of Human Shape and Pose. In *CVPR*, 2018. 2

[17] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D Human Dynamics from Video. In *CVPR*, 2019. 2

[18] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *CVPR*, 2020. 2

[19] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. In *ICCV*, 2021. 9

[20] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense Correspondence Across Scenes and Its Applications. *TPAMI*, 2010. 2

[21] H Christopher Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 1981. 2

[22] David G Lowe. Distinctive Image Features from Scale-invariant Keypoints. *IJCV*, 2004. 2

[23] Bruce D Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981. 2

[24] Wei-Chiu Ma, Anqi Joyce Yang, Shenlong Wang, Raquel Urtasun, and Antonio Torralba. Virtual Correspondence: Humans as a Cue for Extreme-View Geometry. In *CVPR*, 2022. 2

[25] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the Ambiguity of Object Detection and 6D Pose from Visual Data. In *ICCV*, 2019. 2

[26] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *TOG*, 2017. 2

[27] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative Camera Pose Estimation Using Convolutional Neural Networks. In *ACIVS*, 2017. 3

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 1

[29] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *T-RO*, 2015. 2

[30] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *T-RO*, 2017. 2

[31] Kieran A Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-PDF: Non-Parametric Representation of Probability Distributions on the Rotation Manifold. In *ICML*, 2021. 2, 3, 4

[32] David Nistér. An Efficient Solution to the Five-point Relative Pose Problem. *TPAMI*, 2004. 2

[33] David Novotny, Diane Larlus, and Andrea Vedaldi. Learning 3D Object Categories by Looking Around Them. In *ICCV*, 2017. 2

[34] Brian Okorn, Qiao Gu, Martial Hebert, and David Held. ZePHyR: Zero-shot Pose Hypothesis Scoring. In *ICRA*, 2021. 2

[35] Brian Okorn, Mengyun Xu, Martial Hebert, and David Held. Learning Orientation Distributions for Object Pose Estimation. In *IROS*, 2020. 2

[36] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. In *ICCV*, 2021. 2, 5

[37] Chris Rockwell, Justin Johnson, and David F Fouhey. The 8-Point Algorithm as an Inductive Bias for Relative Pose Prediction by ViTs. In *3DV*, 2022. 3

[38] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 7

[39] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020. 2, 7

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 2, 4, 5, 7

[41] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*, 2016. 2, 7

[42] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. SparsePose: Sparse-View Camera Pose Regression and Refinement. In *CVPR*, 2023. 3, 6, 7, 9, 12

[43] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *IROS*, 2012. 6

[44] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical Capsules: Self-supervised Capsules in Canonical Pose. In *NeurIPS*, 2021. 2

[45] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *NeurIPS*, 2021. 2

[46] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle Adjustment—A Modern Synthesis. In *International workshop on vision algorithms*, 1999. 2

[47] Shinji Umeyama. Least-squares Estimation of Transformation Parameters Between Two Point Patterns. *TPAMI*, 1991. 6, 11

[48] Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. MetaPose: Fast 3D Pose from Multiple Views without 3D Supervision. In *CVPR*, 2022. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *NeurIPS*, 2017. 3

[50] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *ICRA*, 2017. 2

[51] Jay M Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David MS Johnson, et al. SegICP: Integrated Deep Semantic Segmentation and Pose Estimation. In *IROS*, 2017. 2

[52] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. In *BMVC*, 2019. 2

[53] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In *CVPR*, 2020. 2

[54] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*, 2021. 1

[55] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild. In *ECCV*, 2020. 2

[56] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, 2022. 2, 3, 4, 5, 6, 7, 11

[57] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural Reflectance Surfaces for

Sparse-view 3D Reconstruction in the Wild. In *NeurIPS*, 2021. 1, 9

[58] Richard Zhang. Making Convolutional Networks Shift-Invariant Again. In *ICML*, 2019. 11

[59] Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling View-conditioned Diffusion for 3D Reconstruction. In *CVPR*, 2023. 1

6. Supplemental Materials

6.1. Implementation Details

Network Details. An anti-aliased [58] ResNet-50 [14] is used to extract image features. These image features are then positionally encoded, concatenated with bounding box parameters, and passed as input to our encoder, the architecture of which is shown in Fig. 9. The pairwise rotation score predictor $g(f_1, f_2, R)$ is a 4 layer MLP which takes as input concatenated transformer features and outputs a score. The translation regressor architecture is depicted in Fig. 10. For the Pose Regression approach, a network similar to the translation regressor is used to predict the 6D rotation representation, instead of the translation residual.

Dataset. Following [56], we set aside the following 10 of 51 CO3Dv2 categories to test generalization: ball, book, couch, frisbee, hotdog, kite, remote, sandwich, skateboard, and suitcase.

Recovering Global Rotations from Pairwise Predictions. Our relative rotation predictor g infers the distribution over pairwise relative rotations, but does not directly yield a globally consistent set of rotations for all the images within the scene. To obtain these, we follow the optimization procedure proposed in RelPose [56]. We briefly summarize this below, and refer the reader to [56] for details.

As this optimization only relies on energies for relative rotations and is agnostic to a canonical global rotation, we can fix the rotation of the first camera as I. To obtain an initial set of global rotations, RelPose first constructs a dense graph between images, with each edge representing the most likely relative rotation and the weight of the edge corresponding to the score of this rotation. It then constructs a maximum spanning tree such that only N-1 edges representing relative rotations are preserved, and this spanning tree (with each edge representing a relative rotation) can be used to recover per-image rotations assuming $R_1 = I$. This serves as the initialization for the iterative coordinate ascent search algorithm to optimize the global rotations. We refer the reader to [56] for more details. Note that, unlike [56] which predicts the relative rotation distributions by only considering features from image pairs, our approach obtains these features by jointly reasoning over all images.

We also note that the task of optimizing for a globally consistent set of rotations given relative poses is similar in spirit to Rotation Averaging [12]. The main difference is that Rotation Averaging assumes a unimodal Gaussian distribution centered at every relative rotation whereas our coordinate-ascent method can accommodate a more expressive, often multi-modal probability distributions.

Optimal Placement of World Origin for Translation Prediction. We now provide a derivation of why placing the world origin at the intersection of the cameras' optical axes is optimal, and specifically, preferable to placing the origin at the first camera (as prior work does). Our derivation assumes a 'look-at' scene, where all of the cameras are 'looking at' some point in the world frame $\mathbf{p}^*$. More specifically, we assume that while the configuration of the camera pose may be undetermined/ambiguous, it is constrained such that the coordinate of this 'look-at' point is fixed w.r.t the camera to $\mathbf{l}_i$ *e.g.* if a camera is moving while looking at a ball 1 unit away, $\mathbf{l}_i = [0, 0, 1]$.

As before, one can transform a point in world coordinates into camera coordinates with $\mathbf{x}_c^i = R_i\mathbf{x}_w + \mathbf{t}_i$. In the more general case, we can then express the 'look-at' constraint as a simple linear equation:

$$\mathbf{l}_i = R_i\mathbf{p}^* + \mathbf{t}_i \tag{4}$$

If we enforce that the camera-frame coordinate of the look-at point remains fixed under possible transformations *i.e.* $\frac{\partial \mathbf{l}_i}{\partial R_i} = 0$, we see that the camera translation varies with rotation and that $||\frac{\partial \mathbf{t}_i}{\partial R_i}|| \propto ||\mathbf{p}^*||$. This dependence is minimized to be 0 if and only if $\mathbf{p}^* = 0$ *i.e.* the 'look-at' point is chosen to be the center of the world coordinate system.

6.2. Evaluation Protocol

Optimal Alignment for Camera Centers Evaluation. Given ground truth camera centers $\{\hat{\mathbf{c}}_i\}_{i=1}^N$ and predicted camera centers $\{\mathbf{c}_i\}_{i=1}^N$ where $\mathbf{c}_i, \hat{\mathbf{c}}_i \in \mathbb{R}^3$, we aim to evaluate how close the two are. However, as the world coordinate system (and thus the location of cameras in it) are ambiguous up to a similarity transform, we first compute the optimal similarity transform by solving:

$$\underset{s,R,\mathbf{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\hat{\mathbf{c}}_i - (sR\mathbf{c}_i + \mathbf{t})\|^2 \tag{5}$$

we compute the optimal scale $s \in \mathbb{R}^+$, rotation $R \in SO(3)$, and translation $\mathbf{t} \in \mathbb{R}^3$ using the algorithm derived by Umeyama [47].

After transforming the predicted camera centers $\tilde{\mathbf{c}}_i = sR\mathbf{c}_i + \mathbf{t}$, we compute the proportion of camera centers within 0.2 of the scene scale:

$$\|\hat{\mathbf{c}}_i - \tilde{\mathbf{c}}_i\| < 0.2\sigma \tag{6}$$

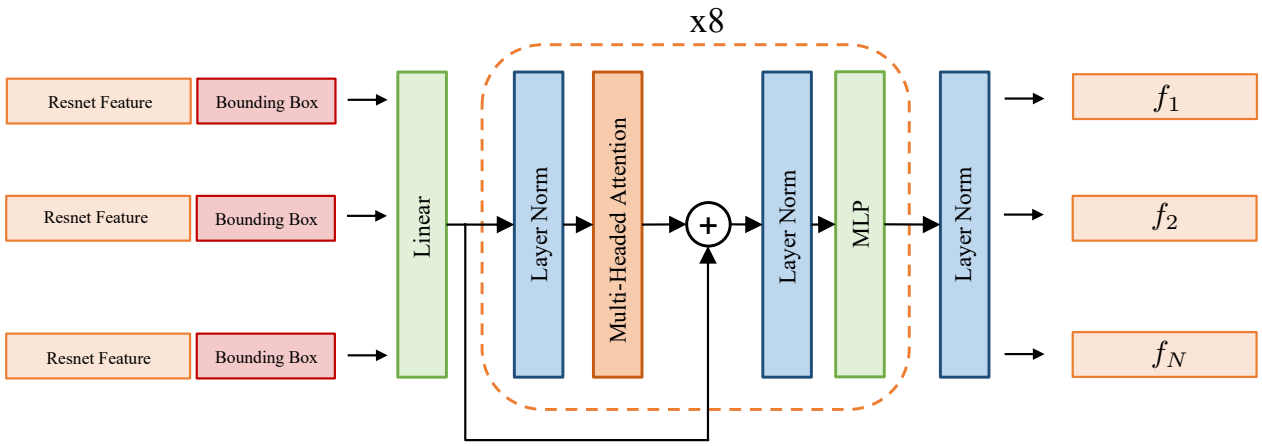

Figure 9: **Encoder Architecture Diagram** Our transformer encoder takes as input a set of N Resnet image features concatenated with the bounding box information for that image. After 8 layers of 12-headed self attention and a final Layer Normalization layer, we receive output features f_1 ... f_N. Unlike the input Resnet features, these each of these features f_1 ... f_N depend on every image in the set.

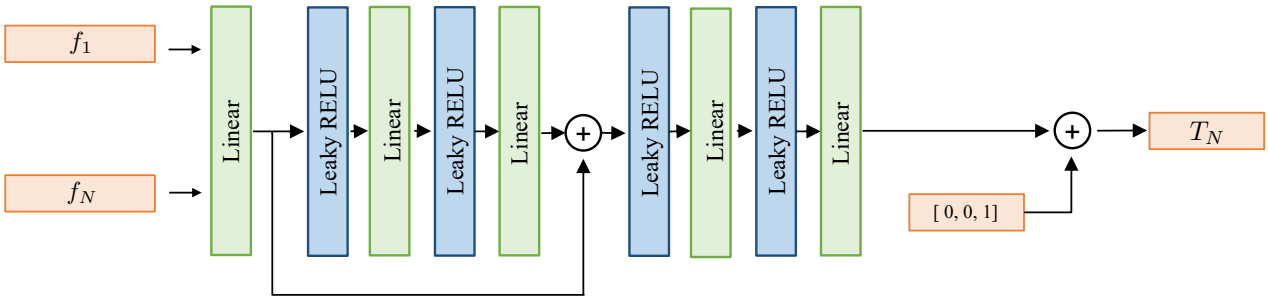

Figure 10: **Translation Regressor Architecture Diagram** For regressing to T_N, we concatenate transformer features f_1 and f_N as input to the translation regressor network. We choose to include f_1 as an input since the scale of the scene is determined by the first image (the norm of the translation target for the first camera is always set to 1).

where σ is the distance from the centroid of *all* camera centers in the video sequence to the furthest camera center (as done in [42]). Note that this scene scale σ is computed with respect to *all* camera poses in the video sequence, whereas the optimal similarity transform is computed with respect to the N cameras used for evaluation.

Optimal Alignment for Camera Translation Evaluation. We note that the camera center ($\mathbf{c}_i = -R_i\mathbf{t}_i$) encompasses both the predicted rotation and translation. We thus also propose to evaluate the predicted camera translation independently by measuring how close the predictions $\{\mathbf{t}_i\}$ are to the ground-truth $\hat{\mathbf{t}}_i$—but we first need to account for the fundamental ambiguity (up to a similarity transform) in defining the world coordinate system.

To compute the transformation of camera translations under similarity transforms, let us consider a camera extrinsic $(R_i, \mathbf{t}_i)$ which maps world points $\mathbf{x}_w$ to camera frame $\mathbf{x}_c^i = R_i\mathbf{x}_w + \mathbf{t}_i$. If we defined a new world frame $\tilde{\mathbf{x}}_w$ related to the current one via a similarity transform $\mathbf{x}_w = \frac{1}{s}(R\tilde{\mathbf{x}}_w + \mathbf{t})$, and correspondingly define a scaled camera-frame coordinate system $\tilde{\mathbf{x}}_c^i$ that follows $\mathbf{x}_c^i = \frac{1}{s}\tilde{\mathbf{x}}_c^i$, we can see that the new extrinsics relating $\tilde{\mathbf{x}}_c^i$ to $\tilde{\mathbf{x}}_w$ correspond to $\tilde{R}_i = R_iR$, and $\tilde{\mathbf{t}}_i = R_i\mathbf{t} + s\mathbf{t}_i$. Therefore, to account for a similarity transform ambiguity, we search for the optimal $s, \mathbf{t}$ that minimize (using linear least-squares):

$$\underset{s,\mathbf{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\hat{\mathbf{t}}_i - (s\mathbf{t}_i + R_i\mathbf{t})\|^2 \tag{7}$$

Note that the above alignment objective does not care for the rotation component in world similarity transform. Intuitively, this is because the translation $\mathbf{t}_i$ corresponds to the coordinate of the world origin in the camera frame, and simply rotating the world coordinate system without moving its origin does not alter the origin's coordinate in camera frame. Given the optimal alignment, we measure the accuracy between the transformed translations $\tilde{\mathbf{t}}_i = s\mathbf{t}_i + R_i\mathbf{t}$

	Num Frames	2	3	4	5	6	7	8
Seen	Ours (Acc@5)	42.1	43.6	44.4	44.7	45.0	45.1	45.4
	Ours (Acc@10)	70.5	72.3	73.6	74.2	74.7	75.2	75.2
	Ours (Acc@15)	81.8	82.8	84.1	84.7	84.9	85.3	85.5
	Ours (Acc@30)	89.2	90.1	91.0	91.5	91.7	92.0	92.1
Unseen	Ours (Acc@5)	30.7	31.9	32.8	33.5	33.9	34.0	33.9
	Ours (Acc@10)	57.7	58.7	60.4	61.2	61.8	62.0	62.5
	Ours (Acc@15)	69.8	71.1	71.9	72.8	73.8	74.4	74.9
	Ours (Acc@30)	78.3	80.7	81.7	82.7	83.4	83.6	84.0

Table 4: **Rotation accuracy at 5, 10, 15, and 30-degree thresholds.**

	Num Frames	2	3	4	5	6	7	8
Seen	Ours (Acc@0.1)	-	85.0	78.0	74.2	71.9	70.3	68.8
	Ours (Acc@0.2)	-	92.3	89.1	87.5	86.3	85.9	85.5
	Ours (Acc@0.3)	-	95.4	92.8	91.5	90.7	90.8	90.4
Unseen	Ours (Acc@0.1)	-	70.6	58.8	53.4	50.4	47.8	46.6
	Ours (Acc@0.2)	-	82.5	75.6	71.9	69.9	68.5	67.5
	Ours (Acc@0.3)	-	88.7	83.1	80.3	78.8	77.6	77.2

Table 5: **Camera center evaluation at 0.1, 0.2, 0.3 thresholds.**

	Num Frames	2	3	4	5	6	7	8
Seen	Ours (Acc@0.1)	93.9	90.5	88.9	87.8	87.0	86.5	86.2
	Ours (Acc@0.2)	98.8	97.9	97.6	97.4	97.3	97.1	96.9
	Ours (Acc@0.3)	99.5	99.2	99.1	99.0	98.9	98.9	98.8
Unseen	Ours (Acc@0.1)	86.2	79.7	76.9	74.9	74.5	73.9	73.6
	Ours (Acc@0.2)	96.0	93.8	93.1	92.3	92.2	91.8	91.6
	Ours (Acc@0.3)	97.7	97.4	97.1	96.8	96.8	96.5	96.5

Table 6: **Camera translation evaluation at 0.1, 0.2, 0.3 thresholds.**

and the GT translations $\hat{\mathbf{t}}_i$ as the the proportion of errors within 0.2 of the scene scale: $\|\hat{\mathbf{t}}_i - \tilde{\mathbf{t}}_i\| < 0.2\sigma$.

6.3. Detailed Experiments

We analyze how rotation accuracy affects camera center accuracy in Fig. 12 and translation accuracy in Fig. 11.

We report the rotation and translation performances at additional thresholds, reporting joint rotation accuracy in Tab. 4, camera center accuracy in Tab. 5, and translation accuracy in Tab. 6.

We report per-category evaluations for rotations in Tab. 7, camera centers in Tab. 8, and translations in Tab. 9. Among unseen object categories, we find that rotationally symmetric objects (ball, frisbee) are consistently challenging. Some categories may be ambiguous given few views (hotdog, kite), but these ambiguities are easy to resolve given more images. Objects that have many salient features (suitcase, remote) tend to have the highest performance.

6.4. Additional Qualitative Results

For discussion of failure modes, please refer to Fig. 13. For more qualitative results, please refer to the project webpage: https://amyxlase.github.io/relpose-plus-plus.

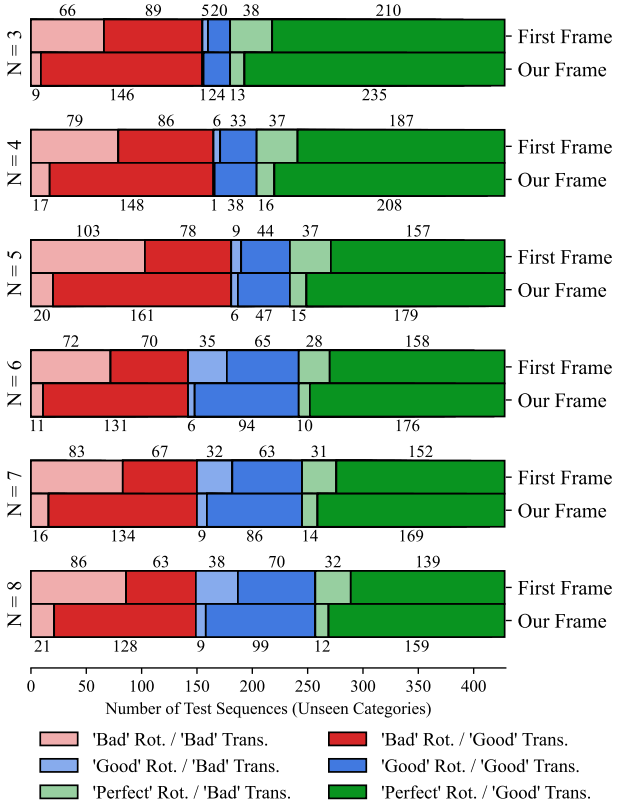

Figure 11: **Relationship between Rotation Accuracy and Translation Accuracy.** When rotation are correctly predicted, we find that translations predicted in both our proposed 'Look-at Centered' coordinate system and the standard 'First Camera Frame' coordinate system which places the world origin at the first camera perform well. However, when rotations are less accurate, only translations in our coordinate system continue to perform well. In this figure, we analyze the effect of rotation accuracy on translation accuracy. For each of the 428 sequences from unseen categories, we consider its rotation accuracy as the number of relative rotations between camera pairs correctly predicted to be within 15 degrees. We define its translation accuracy to be the number of camera translations predicted to be within 20% of the scene scale. We then bin the rotation accuracy into 'Bad' ([0, 2/3)), 'Good' ([2/3, 1)), and 'Perfect' ([1, 1]), and the translations accuracy into 'Bad' ([0, 2/3)) and 'Good' ([2/3, 1]). We compute these accuracies using a re-trained model that predicts translation in ID Frame and Our Frame using the same rotations. We find that regardless of the quality of the rotation predictions, translations predicted in our camera frame are largely accurate. On the other hand, translations predicted in the First Camera Frame degrade significantly when rotation predictions are inaccurate. This is not surprising since predicting translation in the ID frame couples the predicted rotation (see main paper).

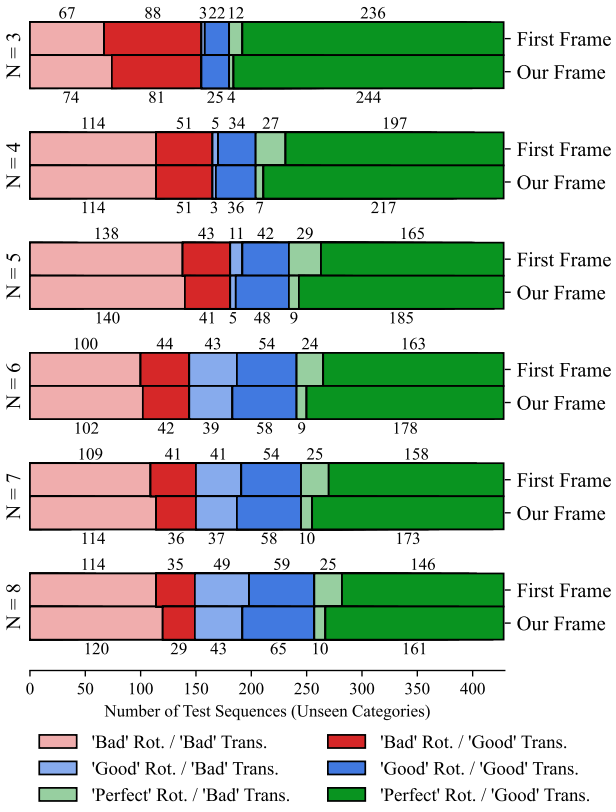

Figure 12: **Relationship between Rotation Accuracy and Camera Center Accuracy.** To compare the effect of rotation accuracy on camera center evaluation, we follow the same protocol described in Fig. 11. Because we use an optimal similarity transform, N=2 is perfectly aligned and thus omitted. When evaluating camera center error, we find that camera center error increases significantly when rotation accuracy is low, which makes it difficult to disambiguate the effect of errors due to rotations versus translations. This serves as the motivation for using our proposed translation evaluation metric, as shown in Fig. 11.

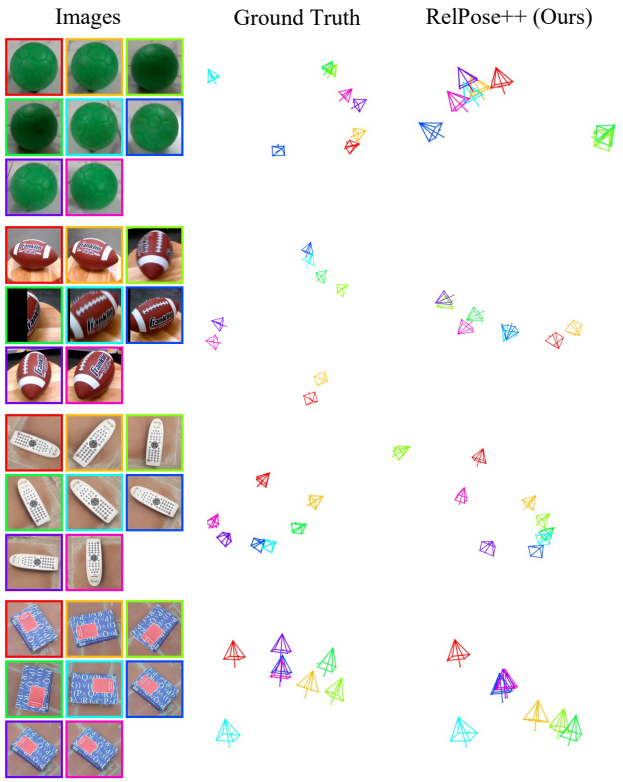

Figure 13: **Failure Modes.** Although handling symmetry is a primary motivation for our work, some symmetries are too challenging to handle. In the top two examples of the balls, our method cannot correctly recover the camera rotation since the object looks almost the same from multiple viewpoints. Unintuitively, our translation estimation is sometimes worse when all viewpoints are entirely fronto-parallel. This could be because the scene normalization procedure that places the world origin at the intersection of the optical axes at training time is unstable when all cameras are facing the exact same direction.

	Num Frames	2	3	4	5	6	7	8
Seen Categories	apple	74.8	75.1	75.3	77.6	78.5	79.3	79.5
	backpack	85.0	87.3	88.4	89.0	89.7	89.9	89.8
	banana	84.1	83.9	84.8	85.1	86.0	86.4	86.6
	baseballbat	84.3	83.3	89.5	90.7	89.8	90.7	91.1
	baseballglove	76.0	69.8	73.8	75.7	74.5	77.3	77.4
	bench	89.6	88.9	89.9	89.5	90.1	90.1	89.6
	bicycle	81.6	85.9	87.1	88.4	88.0	87.8	87.9
	bottle	77.6	81.2	82.3	82.4	83.8	83.6	84.1
	bowl	88.7	90.2	90.2	91.0	90.6	91.2	91.3
	broccoli	62.5	64.6	69.6	68.5	69.8	69.6	70.9
	cake	75.6	78.4	78.7	78.4	78.5	78.4	78.8
	car	86.8	87.4	89.6	90.9	91.4	90.5	90.8
	carrot	83.7	84.9	86.5	86.1	86.5	86.6	86.4
	cellphone	85.6	85.9	86.6	86.8	86.3	87.2	87.6
	chair	94.9	97.0	97.5	97.6	98.1	98.6	98.4
	cup	69.2	71.2	72.4	73.1	74.3	74.4	75.6
	donut	59.2	62.7	60.9	64.1	65.3	65.8	66.0
	hairdryer	86.1	86.5	86.5	88.2	88.6	89.1	88.8
	handbag	77.8	80.4	82.1	82.7	82.2	82.3	82.7
	hydrant	94.0	92.5	94.5	95.4	96.3	95.6	95.7
	keyboard	89.4	90.8	92.4	92.8	93.1	92.9	92.9
	laptop	96.8	96.9	97.2	97.3	97.4	97.1	97.0
	microwave	78.4	81.3	81.9	81.9	81.2	82.8	81.3
	motorcycle	85.2	88.9	88.9	90.2	90.3	90.2	89.7
	mouse	89.0	89.7	90.5	90.5	90.7	90.8	90.8
	orange	70.9	70.9	72.1	73.4	74.0	75.4	74.9
	parkingmeter	70.0	65.6	65.6	67.0	66.9	67.9	69.9
	pizza	78.1	80.3	83.8	83.1	84.1	85.5	84.6
	plant	73.8	74.6	74.7	75.9	75.8	77.2	77.3
	stopsign	84.5	87.8	88.6	88.8	89.2	88.6	89.2
	teddybear	85.0	86.3	86.4	87.7	88.3	88.5	88.3
	toaster	90.4	93.9	94.3	95.2	94.7	95.6	95.4
	toilet	93.1	94.0	95.1	95.7	96.1	95.8	96.6
	toybus	88.5	86.4	86.0	85.9	84.7	86.0	85.6
	toyplane	64.1	68.4	70.5	70.5	70.3	71.0	71.9
	toytrain	82.5	86.5	88.2	87.1	87.4	88.5	88.8
	toytruck	81.3	82.4	85.1	85.9	86.5	86.5	87.8
	tv	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	umbrella	81.6	84.5	85.3	85.0	85.8	86.8	86.4
	vase	77.4	79.6	80.4	81.3	81.3	82.5	81.9
	wineglass	72.6	70.9	72.9	74.3	74.3	73.8	74.4
	Mean	81.8	82.8	84.1	84.7	84.9	85.3	85.5
Unseen Categories	ball	52.5	53.9	54.7	55.2	55.3	56.3	56.1
	book	71.1	72.7	76.0	77.2	78.4	77.2	77.9
	couch	80.0	81.7	82.6	82.2	83.1	84.8	85.3
	frisbee	67.2	69.3	71.7	74.2	75.8	74.1	76.3
	hotdog	61.4	59.0	58.3	58.1	61.1	65.6	66.3
	kite	63.1	66.4	67.8	70.3	69.4	69.2	69.8
	remote	74.4	78.1	81.6	82.2	82.2	83.4	84.1
	sandwich	66.0	67.7	68.3	67.5	67.8	69.9	68.9
	skateboard	71.1	71.1	67.0	69.1	72.4	71.9	71.7
	suitcase	91.6	90.9	91.2	92.4	92.7	91.8	92.4
	Mean	69.8	71.1	71.9	72.8	73.8	74.4	74.9

Table 7: **Per category rotation accuracy at 15 degrees.**

	Num Frames	2	3	4	5	6	7	8
Seen Categories	apple	-	96.1	94.6	94.1	93.5	91.4	91.8
	backpack	-	93.6	90.3	90.1	89.1	89.0	88.9
	banana	-	98.1	94.2	93.5	92.3	91.0	90.8
	baseballbat	-	88.1	83.6	84.9	80.2	79.4	82.0
	baseballglove	-	91.1	85.7	83.5	80.4	82.1	80.3
	bench	-	91.1	86.2	85.7	86.3	85.1	82.9
	bicycle	-	91.5	89.5	89.0	87.7	87.6	86.4
	bottle	-	93.5	89.6	87.0	86.8	86.5	84.9
	bowl	-	94.0	91.5	91.0	90.5	90.0	90.5
	broccoli	-	94.4	92.8	91.1	89.8	89.2	89.7
	cake	-	89.6	84.0	81.6	80.1	79.6	80.1
	car	-	91.9	92.2	92.8	89.3	90.2	89.7
	carrot	-	95.1	92.9	91.9	90.1	89.4	88.0
	cellphone	-	92.3	90.2	85.2	82.9	81.7	82.6
	chair	-	98.6	98.2	97.9	97.7	98.4	98.2
	cup	-	89.6	83.6	81.5	80.5	79.4	80.4
	donut	-	81.9	74.8	70.2	70.1	68.1	69.4
	hairdryer	-	98.1	96.4	95.3	94.6	94.8	94.4
	handbag	-	88.6	84.4	82.3	80.7	78.8	77.9
	hydrant	-	95.6	96.6	96.0	96.3	95.7	95.4
	keyboard	-	92.7	90.7	90.5	89.2	87.6	86.9
	laptop	-	98.5	96.5	94.6	93.5	93.7	92.8
	microwave	-	85.1	79.2	74.7	69.7	74.2	71.9
	motorcycle	-	98.1	96.4	95.4	96.2	95.3	94.2
	mouse	-	98.6	97.6	97.8	97.3	97.2	95.9
	orange	-	92.0	87.4	84.3	84.3	83.4	83.3
	parkingmeter	-	83.3	71.7	70.0	70.0	67.6	72.5
	pizza	-	89.2	88.6	84.8	85.2	84.6	82.7
	plant	-	91.4	84.9	82.6	83.4	83.4	82.5
	stopsign	-	95.6	91.5	90.4	89.3	86.5	86.6
	teddybear	-	96.3	95.1	94.1	94.6	94.2	93.2
	toaster	-	94.9	93.5	94.0	92.2	94.1	93.5
	toilet	-	94.7	89.3	88.0	85.2	84.5	84.7
	toybus	-	90.5	86.5	84.3	81.2	86.0	83.1
	toyplane	-	86.3	82.1	78.6	75.4	74.8	74.2
	toytrain	-	88.3	87.2	84.6	83.8	82.4	82.2
	toytruck	-	88.4	83.9	81.3	80.4	78.9	79.7
	tv	-	95.6	100.0	100.0	100.0	100.0	100.0
	umbrella	-	95.5	94.5	93.0	92.0	92.6	90.6
	vase	-	90.4	85.2	83.6	82.3	81.7	80.5
	wineglass	-	85.8	78.1	75.9	72.6	71.6	71.6
	Mean	-	92.3	89.1	87.5	86.3	85.9	85.5
Unseen Categories	ball	-	85.4	77.1	72.9	69.4	67.7	65.2
	book	-	78.9	72.1	70.5	68.8	67.1	66.2
	couch	-	85.9	78.5	75.3	72.5	71.8	71.9
	frisbee	-	82.1	80.2	77.4	78.0	76.1	76.1
	hotdog	-	70.0	61.4	56.3	54.8	56.3	56.2
	kite	-	82.6	71.9	66.0	60.5	57.7	57.1
	remote	-	89.2	82.9	79.8	79.5	78.5	78.9
	sandwich	-	83.5	77.6	72.1	68.3	67.6	66.8
	skateboard	-	75.9	65.3	58.4	58.0	55.7	50.0
	suitcase	-	92.0	89.1	90.1	89.3	87.0	87.0
	Mean	-	82.5	75.6	71.9	69.9	68.5	67.5

Table 8: **Per category camera center accuracy at 0.2**

	Num Frames	2	3	4	5	6	7	8
Seen Categories	apple	100.0	97.9	97.7	97.8	97.7	97.3	97.2
	backpack	99.4	99.2	99.2	99.0	99.0	98.8	98.7
	banana	96.3	95.4	94.9	94.9	94.6	93.9	93.7
	baseballbat	97.1	95.7	96.4	96.3	96.0	95.3	95.4
	baseballglove	98.7	96.4	96.7	96.8	97.1	96.8	96.0
	bench	100.0	100.0	99.9	99.7	99.4	99.4	99.2
	bicycle	99.6	98.3	97.6	97.8	97.4	97.3	97.0
	bottle	100.0	100.0	99.9	99.7	99.6	99.6	99.7
	bowl	99.5	99.2	98.8	98.9	98.5	98.1	98.0
	broccoli	99.2	98.5	98.5	98.5	98.5	98.3	98.6
	cake	99.1	98.1	98.2	97.9	97.3	97.7	97.1
	car	97.4	95.3	95.1	95.8	95.5	95.9	95.1
	carrot	98.5	97.8	97.4	97.2	97.3	97.0	97.0
	cellphone	97.6	95.5	94.8	93.8	92.7	92.0	91.8
	chair	100.0	100.0	99.9	99.9	99.8	99.7	99.7
	cup	98.8	97.1	96.8	97.1	97.0	96.2	96.8
	donut	100.0	100.0	100.0	99.8	99.9	99.9	99.8
	hairdryer	100.0	99.2	99.2	98.6	98.6	98.3	98.5
	handbag	98.8	98.4	98.0	98.0	97.8	97.6	97.3
	hydrant	100.0	99.5	99.5	99.3	99.3	99.1	99.2
	keyboard	96.5	94.2	93.5	93.6	93.9	93.5	92.9
	laptop	96.5	95.3	94.9	94.3	94.3	94.3	94.1
	microwave	96.0	92.5	89.6	88.8	89.7	88.3	87.8
	motorcycle	98.8	99.6	99.5	99.4	99.2	99.4	99.0
	mouse	99.6	99.5	99.5	99.4	99.3	99.1	98.9
	orange	98.6	98.0	96.4	96.1	96.2	95.9	95.9
	parkingmeter	100.0	100.0	100.0	100.0	100.0	100.0	99.6
	pizza	96.2	95.9	96.0	94.7	95.1	94.7	93.7
	plant	100.0	99.7	99.4	99.3	99.2	99.1	99.0
	stopsign	100.0	99.3	98.9	98.9	98.5	98.4	98.5
	teddybear	97.8	97.7	97.7	97.3	97.0	97.0	97.1
	toaster	100.0	100.0	100.0	99.9	99.9	99.9	99.8
	toilet	92.4	89.2	87.8	85.2	85.1	85.5	84.5
	toybus	99.2	99.2	98.5	98.0	98.2	98.2	98.2
	toyplane	99.0	97.3	97.3	96.9	96.6	96.3	96.5
	toytrain	99.4	97.9	98.1	98.0	97.4	97.1	97.2
	toytruck	100.0	99.6	99.5	99.0	99.0	99.0	98.7
	tv	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	umbrella	100.0	99.9	99.7	99.4	99.4	99.3	99.2
	vase	99.8	99.8	99.3	99.3	99.0	98.9	99.2
	wineglass	100.0	99.6	99.4	99.4	99.4	99.4	99.4
	Mean	98.8	97.9	97.6	97.4	97.3	97.1	96.9
Unseen Categories	ball	96.9	95.1	94.1	93.4	92.3	92.2	92.1
	book	98.8	98.5	98.7	98.3	98.2	98.2	97.8
	couch	91.2	85.5	83.9	83.1	81.5	81.9	81.3
	frisbee	96.0	94.4	94.2	93.8	95.2	94.1	94.4
	hotdog	90.0	83.8	83.6	82.0	82.1	81.8	82.3
	kite	93.8	90.0	88.1	86.6	86.4	85.4	84.9
	remote	99.2	97.9	96.6	95.8	95.9	95.9	96.0
	sandwich	97.0	99.0	98.5	97.4	97.2	97.4	96.9
	skateboard	96.7	94.4	93.1	92.9	93.5	91.3	90.8
	suitcase	100.0	99.9	99.9	99.8	99.7	99.7	99.8
	Mean	96.0	93.8	93.1	92.3	92.2	91.8	91.6

Table 9: **Per category camera translation accuracy at 0.2**