

## A Supplementary Material

This supplementary material provides additional details about our Alligat0R model, including training curves, implementation details, and visualizations that complement the main paper.

### A.1 Pre-training Learning Curves

Figure 7 shows the learning curves for the pre-training phase of both CroCo and Alligat0R on the Cub3-50 and Cub3-all datasets. Alligat0R’s loss converges smoothly on both datasets, indicating stable training.

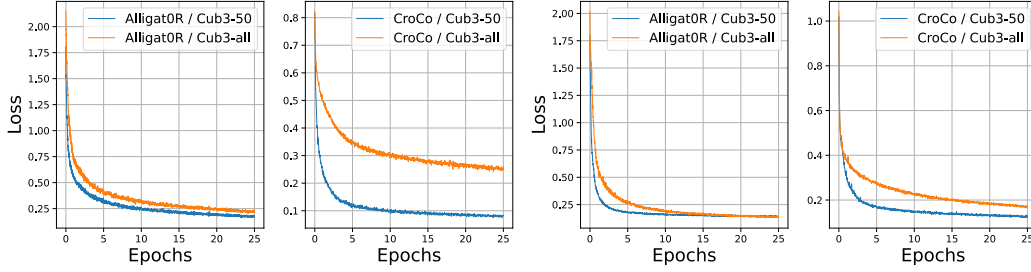


Figure 7: Learning curves during pre-training for CroCo (reconstruction loss) and Alligat0R (segmentation loss) on Cub3-50 and Cub3-all datasets for nuScenes (left) and ScanNet (right). Alligat0R shows stable convergence on both datasets.

### A.2 Fine-tuning Learning Curves

Figure 8 presents the fine-tuning curves for the pose regression task. The plots show the full loss during training (Eq. 9 in the main paper for Alligat0R, Eq. 8 for CroCo). Alligat0R fine-tuned on Cub3-all shows faster convergence and reaches higher accuracy than the same configuration for CroCo, highlighting the transferability of features learned through covisibility segmentation on difficult pairs. The loss increase at 5 epochs corresponds to unfreezing the backbone. For Alligat0R, at 5 epochs we also add the segmentation loss to maintain interpretability.

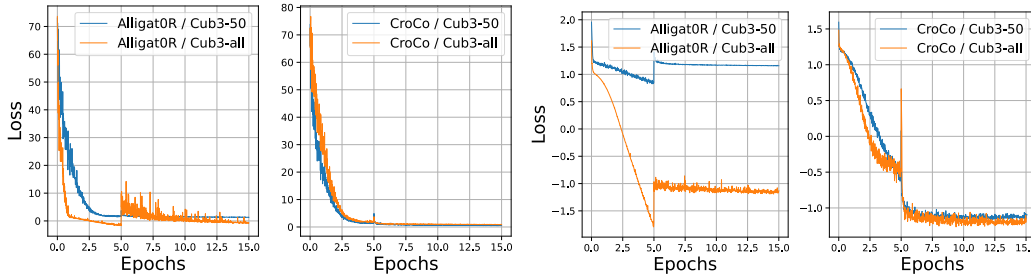


Figure 8: Learning curves during fine-tuning for pose regression for nuScenes (left) and ScanNet (right). Alligat0R pre-trained on Cub3-all converges faster and achieves higher success rates than other methods, demonstrating the effectiveness of our covisibility segmentation pre-training approach.

### A.3 Detailed Performance Analysis

Figure 9 provides a comprehensive breakdown of performance across different geometric criteria on RUBIK. This visualization emphasizes Alligat0R’s strong performance across all difficulty ranges, particularly in challenging scenarios where traditional methods struggle.

### A.4 Implementation Details

We use a ViT-based encoder and transformer decoder backbone similar to CroCo, with 24 layers for the encoder and 12 for the decoder. For pre-training, we use the AdamW optimizer with a learning

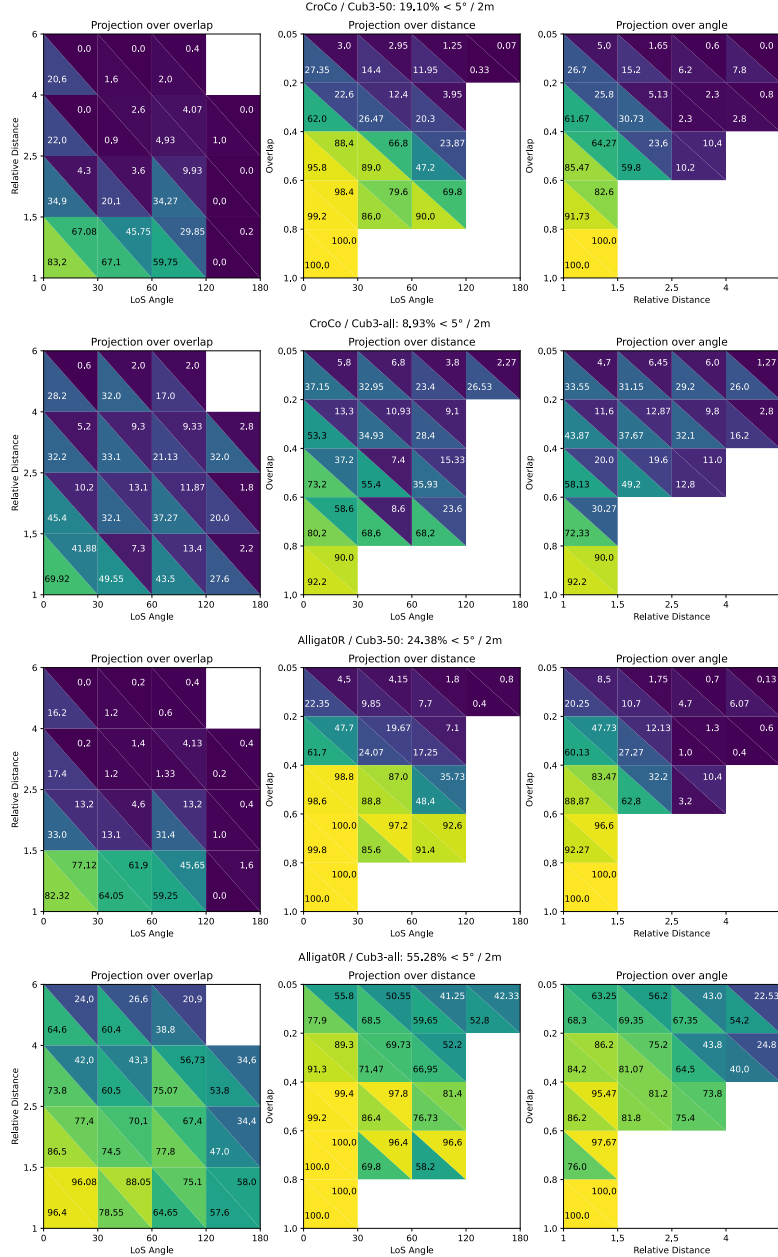


Figure 9: **Detailed breakdown of performance across different geometric criteria.** Success rate either for R@5° or t@2m (bottom-left and top-right of each triangle, respectively), when projecting results onto individual geometric criteria of RUBIK. For each method, we show three plots corresponding to the projection over overlap (left), scale ratio (middle), and viewpoint angle (right).

rate of  $1.5e-4$ , weight decay of 0.05, and a batch size of 32 per GPU. We employ a cosine learning rate schedule with 2 epochs of warmup and train for 25 epochs on our Cub3 datasets.

For fine-tuning on relative pose regression, we follow the two-phase approach described in Section 3.3 of the main paper:

- Phase 1: Freeze the backbone and train only the pose regression head for 5 epochs with a learning rate of  $1e-4$

- Phase 2: Unfreeze the entire network and jointly train with both the pose regression loss and covisibility segmentation loss for an additional 10 epochs with a learning rate of  $5e-5$

The comprehensive architecture of Alligat0R is illustrated in Figure 2 of the main paper, showing both the pre-training and fine-tuning phases. During pre-training, the model learns to segment pixels in each view as covisible, occluded, or outside the field of view with respect to the other view. During fine-tuning, we introduce a pose regression head while maintaining the segmentation capability to leverage the geometric understanding acquired during pre-training.

## A.5 Data Splits and Sampling Protocol

To ensure proper evaluation and avoid data leakage, we carefully separate training and test data:

### Data Splits:

- **RUBIK benchmark:** Uses test scenes from nuScenes, while our Cub3 dataset uses only the training split of nuScenes, with completely disjoint scenes.
- **ScanNet1500 benchmark:** Derived from the standard ScanNet test split, whereas Cub3 uses scenes from the ScanNet training split exclusively.

**Sampling Protocol:** We pre-filter all possible pairs from all training scenes with at least 5% overlap for the "all" version of Cub3, and at least 50% for the "50" version. We then sample from all those pre-filtered pairs to extract the desired number of samples and covisibility masks. The RUBIK benchmark creation follows the protocol described in [28] and is extracted from test scenes, while ScanNet1500 was created from test scenes of ScanNet as first introduced in [32].

## A.6 Additional Ablation Studies

### A.6.1 Impact of the Number of Classes

We implemented two variants of Alligat0R to investigate the importance of our pre-training with the three classes (covisible, occluded, outside FOV). We tried pre-training with only two classes, either covisible or not (in this case occluded and outside FOV are merged), or inside FOV or not (in this case, covisible and occluded are merged) on Cub3-all for nuScenes. The results on the RUBIK benchmark are presented in Table 5.

Table 5: Results on RUBIK for **metric** relative pose regression when pre-training with only two classes. For all experiments, pre-training and fine-tuning is performed using Cub3-all.

Classes	RUBIK		
	5° / 0.5m	5° / 2m	10° / 5m
Covisible or not	22.5	55.7	80.0
Inside FOV or not	<b>24.6</b>	59.6	<b>81.9</b>
All 3 classes	<b>24.6</b>	<b>60.3</b>	<b>81.9</b>

While the performance improvement with three classes is modest, we believe that the model’s knowledge about occluded regions could be beneficial for other tasks. As noted in the main paper, the annotated maps may contain noise between covisible and occluded zones due to reliance on monocular depth predictions.

### A.6.2 Non-metric Relative Pose Regression

We investigated whether our pre-training is useful for non-metric relative pose regression (regressing only the angle for translation) and compared our results with CroCo pre-training on the ScanNet1500 benchmark. For this experiment, we changed our pose regression head using the one from Reloc3r, along with its loss function. The results are shown in Table 6.

Alligat0R significantly outperforms CroCo on this task, demonstrating the versatility of our pre-training approach.

Table 6: Results on ScanNet1500 for **non-metric** relative pose regression. For Alligat0R, pre-training and fine-tuning is performed using Cub3-all, whereas for CroCo, pre-training is performed using Cub3-50 and fine-tuning using Cub3-all.

Pre-Training	ScanNet1500		
	AUC@5	AUC@10	AUC@20
CroCo	13.2	34.1	57.1
Alligat0R	<b>20.5</b>	<b>43.9</b>	<b>66.2</b>

### A.7 Zero-shot Correspondence Estimation on ETH3D

To demonstrate the generalization capabilities of our method beyond the training domains, we evaluate Alligat0R on zero-shot correspondence estimation using the ETH3D dataset [34]. We use the correlations from decoder features to estimate correspondences and measure performance using Average EndPoint Error (AEPE) as described in [1].

Table 7: Zero-shot correspondence estimation on ETH3D dataset using AEPE. Alligat0R consistently outperforms CroCo variants, demonstrating better generalization to out-of-domain dense matching tasks.

Method	Training Data	AEPE ( $\downarrow$ )
Alligat0R	nuScenes-all	<b>43.82</b>
CroCo	nuScenes-all	77.61
Alligat0R	nuScenes-50	<b>44.12</b>
CroCo	nuScenes-50	92.98
Alligat0R	ScanNet-all	<b>36.07</b>
CroCo	ScanNet-all	56.70
Alligat0R	ScanNet-50	<b>38.45</b>
CroCo	ScanNet-50	86.60
CroCo v2	5 datasets	51.55

The results in Table 7 show that Alligat0R consistently outperforms CroCo when trained on the same datasets, and even surpasses CroCo v2 which was pre-trained on 5 datasets. This demonstrates that our covisibility segmentation pre-training learns more generalizable features for dense correspondence tasks.

### A.8 Additional Qualitative Results

We provide additional visualizations from our nuScenes and ScanNet datasets in Figures 10 and 11, along with more predictions from Alligat0R and CroCo in Figure 12.



Figure 10: Covisibility annotation examples from Cub3 for nuScenes. For each image pair, we show the corresponding covisibility maps with color-coding for **covisible**, **occluded**, and **outside FOV** regions. Note how our annotation process handles varying degrees of overlap and challenging viewpoint changes. Let us highlight that some annotations, particularly the distinction between covisible and occluded pixels, may contain noise, especially for nuScenes, and we demonstrate in the experiments that Alligat0R is highly robust to this noise.



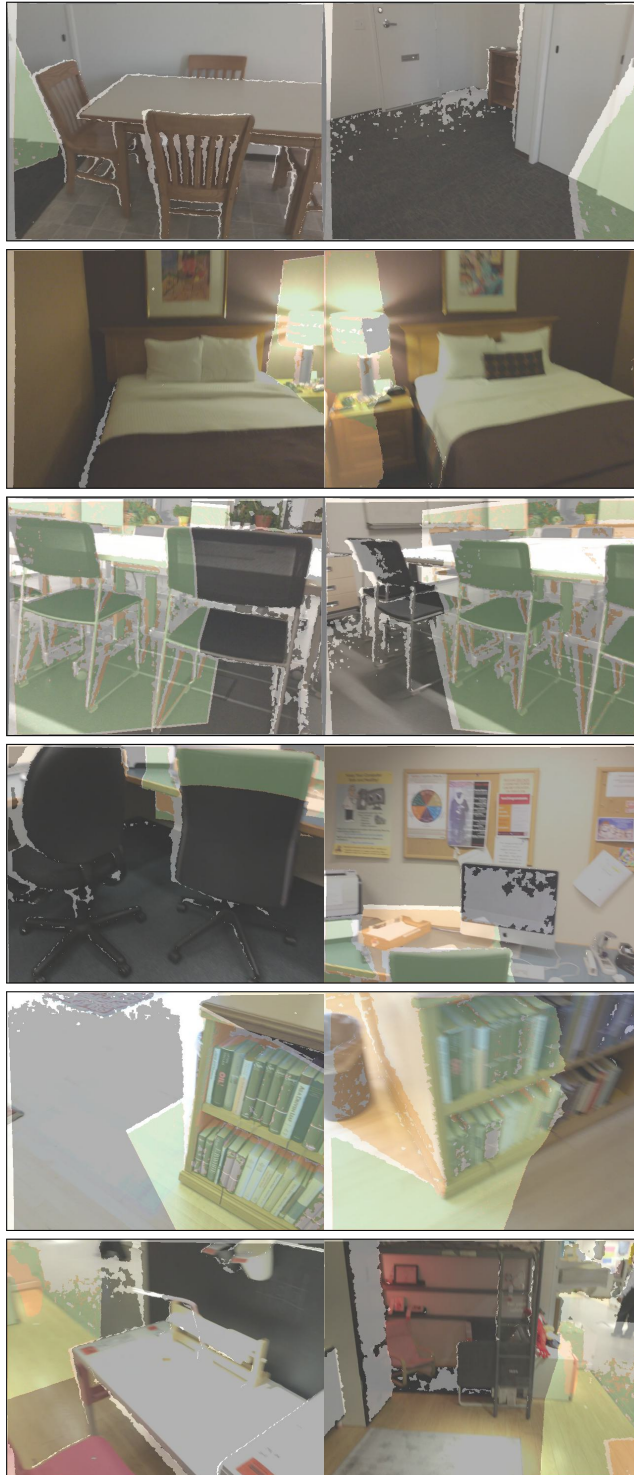


Figure 11: Covisibility annotation examples from Cub3 for ScanNet. For each image pair, we show the corresponding covisibility maps with color-coding for **covisible**, **occluded**, and outside FOV regions. Note how our annotation process handles varying degrees of overlap and challenging viewpoint changes.



Figure 12: **Qualitative comparison.** CroCo’s reconstructions are blurred in masked non-covisible regions, while Alligat0R often correctly identifies **covisible**, **occluded**, outside FOV regions across varying degrees of overlap and viewpoint changes.