# MIND THE ORIENTATION: OVERCOMING UPRIGHT-BIAS IN FEED-FORWARD GEOMETRIC MODELS

**Anonymous authors**
Paper under double-blind review

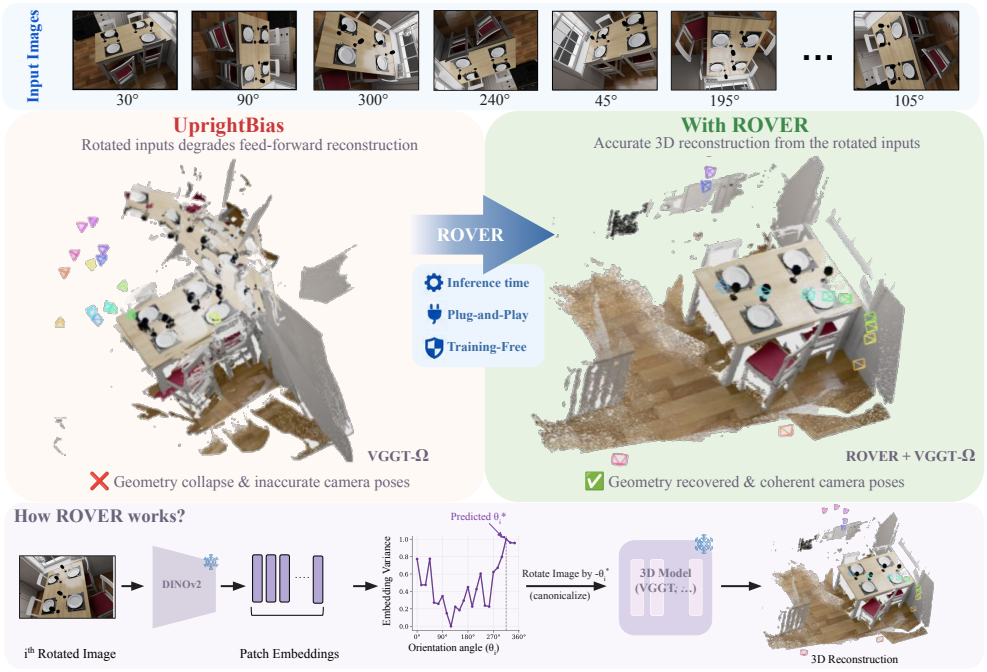



Figure 1: **Robust 3D reconstruction under image rotations with ROVER.** Feeds of arbitrarily rotated images (top) cause standard feed-forward 3D models (e.g., VGGT-Ω) to suffer from geometry collapse due to an implicit upright bias (left). Our training-free, plug-and-play module, ROVER, leverages feature properties in frozen DINOv2 to predict and canonicalize image orientation at inference time (bottom), recovering accurate 3D geometry and coherent camera poses (right).

## ABSTRACT

We uncover a largely overlooked vulnerability in feed forward reconstruction models: sensitivity to image orientation. Across state-of-the-art architectures rotational perturbations of input images cause severe degradation in both camera pose estimation and point-map quality. We term this failure mode UprightBias and trace its origin to a near-universal upright orientation of training corpora, which causes models to implicitly assume canonically upright inputs at inference time. We find that patch-level embedding variance from a DINOv2 encoder provides a reliable zero-shot signal for identifying the canonical upright orientation. Leveraging this insight, we introduce **ROVER** (**RO**tation estimation via **V**ariance-based **E**mbedding **R**ectification), a training-free framework that estimates and rectifies image orientation at inference, leaving all downstream model weights unchanged. For evaluation, we introduce a Controlled Rotation Synthesis protocol that generates rotated views from upright images. We also curate **RotMV** (**Rot**ational **M**ulti-**V**iew), which, to the best of our knowledge, is the first benchmark of paired upright and real-world rotated multi-view sequences captured from fixed viewpoints. Experiments across the datasets demonstrate that ROVER consistently shows improvements, reducing pose estimation errors upto 80% and point-map estimation errors by over 58% on average.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

# 1 INTRODUCTION

Feed-forward multi-view reconstruction models (Wang et al., 2025a;b; Lin et al., 2025; Wang et al., 2026; 2024) regress camera poses and dense geometry directly, without the iterative correspondence-based estimation used by classical structure-from-motion (SfM) (Hartley & Zisserman, 2004; Schönberger & Frahm, 2016; Snavely et al., 2006). Their strong performance on standard multi-view benchmarks, however, does not necessarily translate to robustness under the diverse conditions encountered in real-world image capture (Liu et al., 2026b). One such source of variability is camera orientation, which can vary substantially across handheld photography, egocentric video capture, and other settings with unconstrained camera motion (Xian et al., 2019). Classical SfM pipelines are inherently robust to such changes because their geometric estimation relies on rotation invariant image features (Lowe, 2004). For feed-forward reconstruction models, however, robustness to camera orientation is generally assumed as an inherent property of the reconstruction task.

We put this assumption to the test by asking a simple question: *How robust are modern feed-forward reconstruction models to changes in image orientation?* A well-behaved reconstruction model should produce point clouds invariant to such changes, while camera poses should transform equivariantly. Surprisingly, we find that applying random planar rotations on subset of input images leads to degradation in camera pose estimation and dense reconstruction across state-of-the-art (SOTA) feed-forward models. We observe this degradation across VGGT (Wang et al., 2025a), VGGT-$\Omega$ (Wang et al., 2026), $\pi^3$ (Wang et al., 2025b), MapAnything (MapA.) (Keetha et al., 2026) and Depth Anything 3 (DA3) (Lin et al., 2025), while classical SfM with COLMAP (Schönberger & Frahm, 2016) remains robust to the same perturbations. Quantitatively, Figure 2(a) shows that both Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) increase with the number of rotated input images across the feed-forward models. The corresponding qualitative degradation in reconstructed geometry is shown in Figure 2(b) We further investigate the extent of this orientation sensitivity across rotations about all three axes (roll, pitch, and yaw) in section 4.2.

We refer to this sensitivity to camera orientation as *UprightBias*: the tendency of feed-forward reconstruction models to perform substantially better on images presented in a canonical upright orientation. The effect is particularly striking because it persists across architecturally diverse models, including reference-view attention networks (VGGT, VGGT-$\Omega$), reference-free permutation-equivariant approaches ($\boldsymbol{\pi^3}$), generalized multi-view matchers such as MapA. and DA3. Notably, however, each of these models relies on large-scale 3D reconstruction datasets which predominantly consists of canonically-upright images, and models trained on such image sets may potentially learn orientation-specific regularities instead of generalizing across arbitrary rotations. We investigate this hypothesis through an audit of the training data used by these models in section 4.3, which shows that an overwhelming majority of these datasets consist of images that are upright by construction, spanning indoor RGB-D benchmarks, outdoor photo collections, and a range of other sources.

One natural remedy to fix the Upright Bias would be to rebalance the training data with rotated multi-view images and corresponding geometric annotations (He et al., 2021; Tang et al., 2022; Kang et al., 2019); however, these feed-forward models are built on pretrained visual backbones like DINO-v2/v3 (Oquab et al., 2023; Siméoni et al., 2025), and fine-tuning on a rotation-heavy distribution shifts the backbone away from its pretraining regime, which is known to distort learned representations, even when the fine-tuning objective is otherwise met (Kumar et al., 2022; Tang et al., 2020; Kang et al., 2019; Tang et al., 2026). Further, despite both DA3 and VGGT-$\Omega$ employing $90^\circ$ rotation augmentation in their training pipeline, yet, as shown in Figure 2(a), they *still exhibit* substantial degradation under arbitrarily-rotated inputs. For instance, we observe that a simple fine-tuning of $\pi^3$ on arbitrarily-rotated training images yields not just inconsistent but often degraded performance (see section K of the supplementary), further suggesting that rebalancing the training distribution *alone* is insufficient to eliminate the upright bias. Lastly, we note that retraining large-scale feed-forward models requires *paired* rotated multi-view data with ground-truth geometry, which is currently missing in the existing benchmarks.

Rather than adapting a reconstruction model to rotated inputs, we instead ask whether the inputs *themselves* can be rectified to the canonical orientation before reconstruction. This requires a visual representation that can provide a reliable signal of image orientation without rotation labels or model retraining. To this end, we find that DINOv2 token embeddings exhibit *systematic orientation-dependent variation*, which can be leveraged to enable zero-shot rectification by

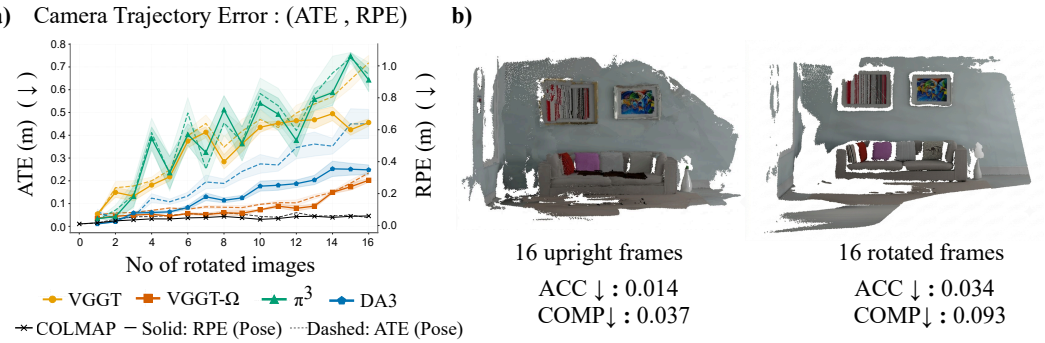



Figure 2: **Robustness against in-plane rotations:** (a) Camera poses error increases as number of rotated input images increases ($\theta \sim \mathcal{U}(0°, 360°)$). (b) Qualitative 3D reconstruction degradation on VGGT-Ω when introducing 16 rotated images sampled from $\theta \sim \mathcal{U}(0°, 360°)$.

identifying the orientation that maximizes feature variance. Building on this observation, we propose **ROVER** (**RO**tation estimation via **V**ariance-based **E**mbedding **R**ectification), a lightweight *training-free* framework that operates at inference time in a plug-and-play manner with any encoder, as illustrated in fig. 1. Along with this, we introduce **RotMV** (**Rot**ational **M**ulti-**V**iew), a real-world benchmark comprising of seven multi-view sequences captured across 16 viewpoints and 12 in-plane camera orientations spanning 360°, enabling systematic evaluation of rotation-invariance in feed-forward models. Extensive Experiments on 7-Scenes, NRGB-D, DTU, ETH3D, Mip-NeRF 360, Tanks & Temples and RotMV, shows that ROVER recovers reconstruction quality under arbitrary planar rotations while preserving performance on canonical inputs. In summary, our main contributions are:

• We identify a previously uncharacterized sensitivity to image orientation in state-of-the-art feed-forward 3D reconstruction models, a phenomenon we term UprightBias, and show that it is systemic across diverse model architectures and families.

• To address it, we propose ROVER – a lightweight, training-free framework that rectifies input rotations at inference time using orientation-dependent variance in DINOv2 embeddings.

• Across six existing multi-view datasets, we conduct a systematic assessment of rotation-invariance of various feed-forward models under controlled rotation synthesis, isolating the effect of camera orientation on reconstruction.

• To complement synthetic evaluation with real-world data, we introduce RotMV, a benchmark comprising paired upright and rotated multi-view sequences captured under systematically varied camera orientations.

## 2 Related Work

**Robustness and Bias in Vision Models:** Although modern deep learning models encode overrepresented training patterns as implicit priors, these task-extrinsic biases often remain hidden under standard benchmarks until those assumptions are violated. Extensive work has demonstrated this across visual recognition, including biases toward texture over shape, shortcut correlations, photographer preferences, object position, and background context (Geirhos et al., 2018; 2020; Torralba & Efros, 2011; Xiao et al., 2020; Liu et al., 2026a). These findings have motivated robustness benchmarks beyond clean in-distribution accuracy (Hendrycks & Dietterich, 2019) and approaches based on architectural inductive biases or inference-time adaptation (Cohen & Welling, 2016; Weiler & Cesa, 2019; Sun et al., 2020b; Wang et al., 2020a). Despite this extensive literature, robustness has been studied primarily in recognition tasks such as classification, segmentation, and detection. In single-image geometry, prior studies have identified specific camera pose and spatial distribution biases in depth estimation (Dijk & Croon, 2019; Zhao et al., 2021), while recent work highlights broader viewpoint instabilities in vision foundation models (Michalkiewicz et al., 2025). Similarly, Jiang et al. (2025) investigates viewpoint invariance in policy learning. However, whether similar priors arise in feed-forward geometric foundation models remains largely unexplored.

**Image Orientation and Uprightness Estimation:** The resulting sensitivity to image orientation naturally raises the question of whether canonical upright orientation can be estimated reliably at inference time. Existing approaches include geometric methods based on line classification (Lee et al., 2021; Huang et al., 2026), feature matching (Nordstrom et al., 2026; Nordström et al., 2026) and learning-based methods such as UprightNet (Xian et al., 2019), which predicts 2-DoF camera orientation from indoor surface normals, and GeoCalib (Veicht et al., 2024), which jointly estimates camera intrinsics and gravity direction through differentiable geometric optimization. Inference-time canonicalization offers an alternative for robustness under distribution shifts. FOCAL (Singhal et al., 2025) searches for a canonical input transformation using test time optimization of likelihoods with foundation-model, while G3TUP! (Nagoor Kani & Snavely, 2026) fine-tunes VGGT to produce upright point clouds.

In contrast, we identify upright bias as a previously unrecognized failure mode shared across modern feed-forward reconstruction models and introduce ROVER, a Plug-and-Play inference-time canonicalization framework that estimates orientation using DINOv2, requiring neither scene-specific geometric structure nor model retraining and generalizing across indoor and outdoor scenes.

# 3 METHOD

**Problem Setup:** Let $\mathcal{F}$ denote a pre-trained feed-forward multi-view reconstruction model that maps a set of $N$ multi-view images to camera pose and dense geometric predictions. Given a set of input images $\mathcal{I}^R = \{\mathbf{I}_i^{\theta_i}\}_{i=1}^N$ where each view $\mathbf{I}_i^{\theta_i}$ is captured at an arbitrary in-plane orientation $\theta_i \in [0°, 360°)$. Our goal is to recover a scene representation $\hat{\mathcal{G}}$ comparable in quality to that obtained from canonically upright inputs, without retraining or modifying $\mathcal{F}$.

Existing multi-view benchmarks lack systematically rotated images with corresponding ground-truth annotations, making it difficult to evaluate robustness to arbitrary camera orientations. We therefore develop a Controlled Rotation Synthesis, a procedure that generates rotated multi-view inputs while preserving the underlying scene geometry and avoiding rotation-induced artifacts. The proposed procedure enables synthetic rotations about all three axes (roll, pitch, and yaw), while avoiding rotation-induced artifacts. To complement the synthetically rotated benchmarks, we construct RotMV, a real-world benchmark comprising multiple scenes, with each scene containing multi-view images captured at multiple camera orientations for every viewpoint. In this section, we discuss the controlled rotation synthesis procedure in section 3.1 and our proposed method for correcting UprightBias in section 3.2.

## 3.1 CONTROLLED ROTATION SYNTHESIS

Our goal is to generate synthetic views by isolating the effect of camera orientation only. The created rotation should alter only the camera pose while leaving the scene content and image formation otherwise unchanged. This is not possible if we naively rotate the images in image space, because it introduces various artifacts like padding, cropping, and resolution changes, as illustrated in Figure 3.

Given an upright multi-view sequence $\mathcal{I}$, we first reconstruct its 3D scene representation $\mathcal{G}_{\text{3DGS}}$ and corresponding camera poses using a 3D Gaussian Splatting(Kerbl et al., 2023). For each view $i$, let its original camera pose be $\mathbf{P}_i = [\mathbf{R}_i \mid \mathbf{t}_i]$, where $\mathbf{R}_i$ and $\mathbf{t}_i$ denote the camera rotation and translation, respectively following camera to world convention. For a synthetic in-plane rotation of $\theta$, represented by $\mathbf{R}_z(\theta) \in \text{SO}(3)$, the corresponding camera pose is given by $\mathbf{P}_i^\theta = [\mathbf{R}_i\mathbf{R}_z(\theta) \mid \mathbf{t}_i]$. Analogously, pitch and yaw rotations are synthesized by replacing $\mathbf{R}_z(\theta)$ with $\mathbf{R}_x(\theta)$ and $\mathbf{R}_y(\theta)$, respectively. The corresponding view is then rendered as:

$$\tilde{\mathbf{I}}_i^{\theta_i} = \text{Render}(\mathcal{G}_{\text{3DGS}}, \mathbf{P}_i^{\theta_i}).$$

The fidelity of $\tilde{\mathbf{I}}_i^{\theta_i}$ depends on the quality of the reconstructed $\mathcal{G}_{\text{3DGS}}$, which may vary with the input views and rotation angle $\theta$. To improve it, we further refine $\mathcal{G}_{\text{3DGS}}$ using DiFix3D+(Wu et al., 2025) Denoiser, and then render the rotated views to obtain the final $\mathbf{I}_i^{\theta_i}$. Low-quality renders may still occur and are discarded using a two-stage filtering pipeline (Wu et al., 2026), detailed in the supplementary section H. Full pipeline details are included in supplementary, specifically the rendering algorithm in section B, quality filtering procedure and dataset statistics in section H.

### 3.2 ROVER

ROVER constructs a canonicalization operator $T$ that maps an arbitrarily oriented input sequence to the canonical upright orientation expected by the pre-trained reconstruction model. Our method realizes the canonicalization operator $T$ by decomposing it into two sequential operators, $T = C \circ E$, where $E : \mathcal{I}^R \to \hat{\Theta}$ estimates the canonical orientation of each image in the input sequence, with $\hat{\Theta} = \{\hat{\theta}_i\}_{i=1}^N$ denoting the estimated orientations, and $C : (\mathcal{I}, \hat{\Theta}) \to \mathcal{I}^{\text{can}}$ applies the corresponding transformations to produce the canonicalized image sequence. Since both operators act entirely in image space, ROVER requires neither retraining nor architectural modifications to the underlying feed-forward reconstruction model.

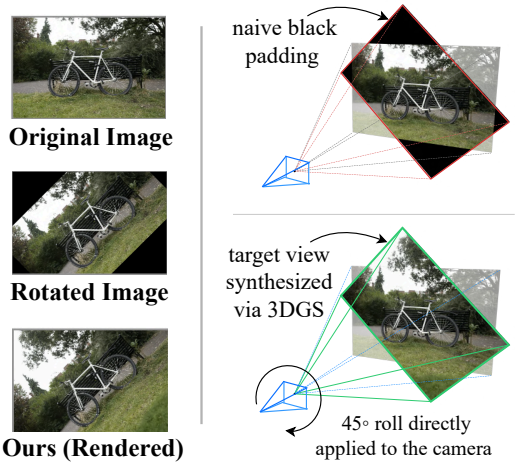



Figure 3: **Controlled Rotation Synthesis.** Naive 2D image rotation (*top right*) introduces black padding, cropping artifacts, or resolution shifts. In contrast, our approach (*bottom right*) applies planar roll $\mathbf{R}_z(\theta)$ directly to the camera pose and synthesizes the target view via 3DGS, generating naturally rotated images.

**Estimating the Canonical Orientation:** We made an important observation that enabled us to estimate orientation directly from the representations of DINOv2 encoder. DINOv2 employs the KoLeo regularizer (Sablayrolles et al., 2018), which encourages uniform dispersion of the learned embeddings. This provides a natural basis for measuring the spatial variance of its patch-level embeddings. Since the DINOv2 pretraining distribution is dominated by naturally upright images, we hypothesize that this dispersion is strongest when an image is presented in its canonical upright orientation. We observe that the spatial variance of final-layer patch embeddings indeed peaks at the canonical upright orientation and decreases as the image deviates from upright, consistently across datasets and scenes (provided in supplementary section C). This makes patch embedding variance a reliable proxy for image uprightness, allowing us to estimate the canonical orientation by maximizing the variance over candidate orientations. For each input image $\mathbf{I}$ and candidate orientation $\theta \in \Theta$, we compute the DINOv2 patch embeddings $\mathbf{Z}^\theta = \phi(\mathbf{I}^\theta) = \{\mathbf{z}_j^\theta\}_{j=1}^M$ with mean embedding $\bar{\mathbf{z}}^\theta$, and estimate the canonical orientation $\hat{\theta}$ by maximizing their spatial variance:

$$V(\theta) = \frac{1}{M} \sum_{j=1}^{M} \left\| \mathbf{z}_j^\theta - \bar{\mathbf{z}}^\theta \right\|_2^2 \quad \Longrightarrow \quad E(\mathbf{I}) = \hat{\theta} = \arg\max_{\theta \in \Theta} V(\theta).$$

**Applying the Canonicalization Operator:** Given $\hat{\theta}$, the canonicalization operator $C$ applies a corrective rotation to restore the image to its upright orientation $\mathbf{I}^{\text{can}} = C(\mathbf{I}) = \mathbf{I}^{-\hat{\theta}}$. Applying $T = C \circ E$ independently to every image in $\mathcal{I}^R$ yields the canonicalized sequence $\mathcal{I}^{\text{can}} = \{\mathbf{I}_i^{\text{can}}\}_{i=1}^N$, which is passed to feed-forward model $\mathcal{F}$. Exhaustive evaluation of $V(\theta)$ over all 360 integer-degree candidates requires 360 DINOv2 forward passes per image. We instead employ a coarse-to-fine search that requires only two batched DINOv2 forward passes per multi-view sequence at inference time. In the coarse stage, we evaluate $V(\theta)$ over $\Theta_{\text{coarse}} = \{15^\circ \times k \mid k = 1, \ldots, 24\}$ and obtain $\theta_c = \arg\max_{\theta \in \Theta_{\text{coarse}}} V(\theta)$. In the fine stage, we evaluate $V(\theta)$ within $[\theta_c - 7^\circ, \theta_c + 7^\circ]$ at $1^\circ$ intervals. Since all candidate rotations for a stage can be batched, each stage requires only a one DINOv2 forward pass for the entire multi-view sequence. We include results for latency in section A and a full algorithmic description is provided in the supplementary algorithm 2.

## 4 Experiments & Results

### 4.1 Experimental Setup

**Implementation & Dataset Details:** We evaluate models across six standard benchmarks: 7-Scenes (Glocker et al., 2013; Shotton et al., 2013), NRGB-D (Azinović et al., 2022), and RotMV,

presented in the main paper, and DTU (Jensen et al., 2014), ETH3D (Schöps et al., 2019; 2017), Mip-NeRF 360 (Barron et al., 2022), and Tanks & Temples (Knapitsch et al., 2017), detailed in the supplementary (section A). Similarly, we include results for in-plane rotation (roll) in the main paper, with pitch and yaw results included in the supplementary section A

While our rendering-based rotation construction provides synthetically rotated sequences, it may not fully capture the characteristics of real-world rotated captures. Thus, we present a novel benchmark RotMV, consisting of seven real-world scenes (*chairs*, *foosball-table*, *building*, *rooftop*, *room*, *statue*, and *table-tennis*). We capture each scene using a camera in $4096 \times 3072$ resolution. Each scene is imaged from 16 fixed viewpoints, and at each viewpoint, the camera is rolled in-plane through 12 orientations spanning a full $360^\circ$ (in $30^\circ$ increments), resulting in 192 images per scene. For each viewpoint, we estimate the base camera pose from the upright ($0^\circ$) image using VGGT. The remaining 11 roll-angle poses are derived analytically by applying the corresponding in-plane rotation to the base camera orientation, keeping the camera center and viewing direction fixed. Additional details about the dataset included visual examples is included in supplementary section G.

**Evaluation Metrics:** Following the evaluation criteria of $\pi^3$ (Wang et al., 2025b), we use 16-view input sequence and evaluate models on two tasks. For camera pose estimation, we report Rotation Accuracy ($\text{RAcc}_{30}\uparrow$), Translation Accuracy ($\text{TAcc}_{30}\uparrow$), and Area Under the Curve ($\text{AUC}_{30}\uparrow$) at a $30^\circ$ threshold (Wang et al., 2023), measuring the fraction of predictions within the threshold for rotation, translation, and both jointly. For point map estimation, we report Accuracy (Acc. $\downarrow$), Completeness (Comp. $\downarrow$), and Normal Consistency (N.C. $\uparrow$), the mean cosine similarity between predicted and ground-truth surface normals (Leroy et al., 2024).

**Evaluation Protocol:** We vary the number of rotated views $m \in \{0, 4, 8, 12, 16\}$, corresponding to $\{0\%, 25\%, 50\%, 75\%, 100\%\}$ rotated view configurations. For each budget $m$, we select $m$ views uniformly at random and assign each an independent roll angles $\theta_i \sim \mathcal{U}(0^\circ, 360^\circ)$ (similarly for pitch and yaw), synthesizing rotated inputs via our controlled rotation synthesis. Ground-truth camera pose are updated to reflect the applied rotation angles and model predictions are aligned to ground truth using Umeyama (Umeyama, 1991). We report the mean and standard deviation across 10 independent random trials per $m$, unless noted otherwise.

### 4.2 How Severe is Upright Bias?

To evaluate whether arbitrary image rotations constitute a meaningful failure mode for feed-forward 3D reconstruction models, we measure performance across the rotation budgets. Table 1 reveals consistent and severe degradation across all models and datasets as the fraction of rotated views increases. On 7-Scenes, camera pose estimation ($\text{AUC}_{30}$) drops for all models, with absolute declines ranging from 17.1 points (VGGT-$\Omega$: 87.5 to 70.4) to 57.2 points (MapA.: 80.9 to 23.7) at $100\%$ rotation. Point-map quality (N.C.) follows the same trend across all models. On NRGB-D, degradation is similarly consistent and universal, with pose accuracy falling by $10.2 - 66.6$ points depending on the model. On ETH3D, all models exhibit substantial declines in both metrics, with $\text{AUC}_{30}$ dropping by 8.5–22.3 points at full rotation. Across all three datasets, these results confirm that UprightBias is a systematic and severe failure mode affecting all evaluated feed-forward reconstruction models. Results on the remaining dataset along with pitch and yaw are included in supplementary section A.

### 4.3 What Causes this Upright Bias?

Having established that rotations consistently degrade reconstruction quality, we investigate the origin of this behavior. As summarized in Table 2, a comprehensive audit of the training corpora of VGGT, VGGT-$\Omega$, DA3, MapA., and $\pi^3$ reveals that the overwhelming majority of training datasets are upright by construction, spanning indoor RGB-D benchmarks, outdoor photo collections, and synthetic datasets alike. Consequently, feed-forward models learn a strong implicit prior that the camera and world coordinate systems share an aligned gravity vector, which fails when rotated inputs are encountered at inference time. A detailed dataset-by-dataset breakdown is provided in the supplementary section K.

Table 2: **Dataset Audit.** Majority of training data across all models is Upright by construction

| Dataset | VGGT | VGGT-$\Omega$ | DA3 | MapA | $\pi^3$ |
|---|---|---|---|---|---|
| Fully Upright | 13 | 18 | 19 | 10 | 11 |
| Partially Upright | 1 | 2 | 1 | 0 | 2 |

Table 1: **Point-map and camera-pose estimation.** Performance under increasing fractions of rotated input views measured by Normal Consistency (N.C.) and $AUC_{30}$, respectively.

| | | Point-map Estimation (N.C.) | | | | | Camera Pose Estimation ($AUC_{30}$) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Rot.** | VGGT | VGGT-Ω | $\pi^3$ | MapA. | DA3 | VGGT | VGGT-Ω | $\pi^3$ | MapA. | DA3 |
| **7-Scenes** | 0% | 0.658 ±0.034 | 0.666 ±0.046 | 0.673 ±0.047 | 0.628 ±0.050 | 0.673 ±0.036 | 86.6 ±6.6 | 87.5 ±9.0 | 87.5 ±6.0 | 80.9 ±15.4 | 88.2 ±7.2 |
| | 25% | 0.632 ±0.030 | 0.666 ±0.005 | 0.641 ±0.022 | 0.606 ±0.020 | 0.648 ±0.025 | 70.0 ±13.2 | 84.1 ±2.5 | 71.6 ±6.8 | 53.7 ±10.9 | 71.2 ±11.0 |
| | 50% | 0.616 ±0.038 | 0.669 ±0.008 | 0.626 ±0.029 | 0.595 ±0.027 | 0.635 ±0.032 | 54.6 ±19.2 | 81.1 ±4.1 | 59.0 ±10.3 | 37.0 ±11.3 | 55.4 ±16.4 |
| | 75% | 0.601 ±0.040 | 0.671 ±0.007 | 0.619 ±0.030 | 0.588 ±0.022 | 0.624 ±0.029 | 44.3 ±18.8 | 76.0 ±6.6 | 49.4 ±10.3 | 27.4 ±9.5 | 51.8 ±11.2 |
| | 100% | 0.604 ±0.030 | 0.662 ±0.015 | 0.609 ±0.028 | 0.584 ±0.024 | 0.633 ±0.019 | 43.8 ±10.4 | 70.4 ±6.6 | 45.3 ±8.7 | 23.7 ±6.6 | 53.3 ±9.9 |
| **NRGB-D** | 0% | 0.802 ±0.032 | 0.830 ±0.038 | 0.809 ±0.063 | 0.749 ±0.052 | 0.822 ±0.035 | 98.6 ±1.2 | 99.3 ±0.6 | 96.4 ±1.9 | 93.9 ±4.3 | 99.9 ±0.2 |
| | 25% | 0.742 ±0.044 | 0.810 ±0.016 | 0.757 ±0.037 | 0.706 ±0.033 | 0.749 ±0.056 | 88.5 ±9.6 | 96.6 ±2.0 | 86.8 ±7.2 | 69.0 ±14.2 | 83.5 ±13.1 |
| | 50% | 0.721 ±0.046 | 0.792 ±0.014 | 0.732 ±0.035 | 0.667 ±0.049 | 0.721 ±0.071 | 78.5 ±14.4 | 94.2 ±2.3 | 78.4 ±8.3 | 49.1 ±14.6 | 72.4 ±16.1 |
| | 75% | 0.705 ±0.054 | 0.773 ±0.011 | 0.722 ±0.036 | 0.649 ±0.046 | 0.706 ±0.071 | 73.4 ±15.6 | 91.4 ±3.1 | 75.9 ±10.0 | 37.0 ±14.7 | 65.0 ±17.1 |
| | 100% | 0.709 ±0.058 | 0.763 ±0.007 | 0.717 ±0.038 | 0.629 ±0.042 | 0.716 ±0.059 | 70.4 ±16.1 | 89.3 ±2.7 | 73.5 ±9.0 | 27.3 ±11.7 | 69.9 ±17.8 |
| **ETH3D** | 0% | 0.832 ±0.074 | 0.908 ±0.038 | 0.880 ±0.049 | 0.762 ±0.141 | 0.818 ±0.129 | 77.1 ±17.2 | 93.4 ±5.3 | 77.3 ±15.2 | 52.0 ±26.7 | 75.7 ±26.2 |
| | 25% | 0.785 ±0.096 | 0.864 ±0.079 | 0.845 ±0.083 | 0.746 ±0.126 | 0.816 ±0.111 | 67.1 ±25.4 | 93.1 ±5.6 | 75.1 ±17.1 | 50.5 ±24.1 | 76.8 ±23.5 |
| | 50% | 0.770 ±0.089 | 0.837 ±0.085 | 0.820 ±0.090 | 0.722 ±0.123 | 0.781 ±0.122 | 64.5 ±24.2 | 92.2 ±6.4 | 73.9 ±17.5 | 49.6 ±24.2 | 69.3 ±27.6 |
| | 75% | 0.729 ±0.103 | 0.818 ±0.097 | 0.793 ±0.101 | 0.709 ±0.126 | 0.774 ±0.126 | 59.8 ±27.5 | 89.9 ±8.3 | 68.8 ±22.4 | 47.7 ±23.4 | 69.8 ±29.2 |
| | 100% | 0.662 ±0.130 | 0.771 ±0.097 | 0.744 ±0.101 | 0.640 ±0.124 | 0.723 ±0.117 | 54.8 ±29.4 | 84.9 ±10.4 | 62.7 ±23.1 | 39.5 ±26.2 | 60.6 ±31.4 |

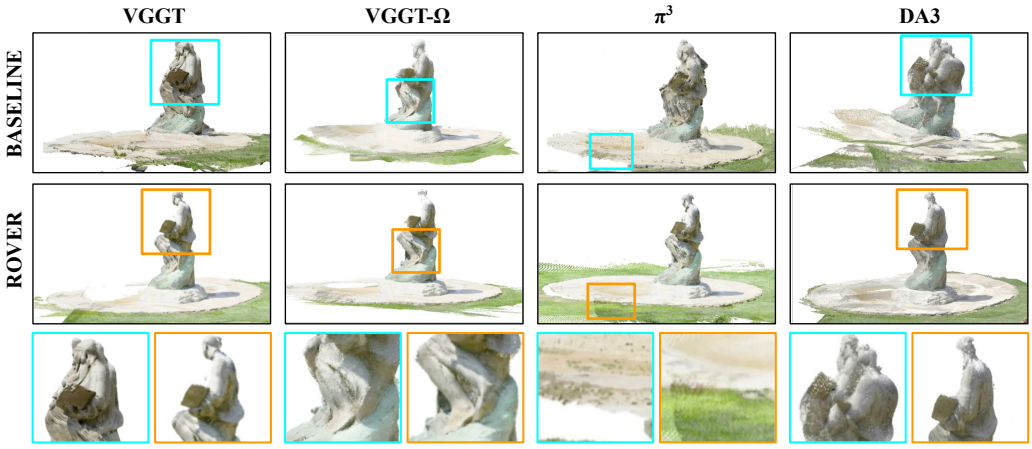



Figure 4: **Qualitative comparison.** Rotated input images to a feed-forward model produces a distorted point cloud (Baseline). Our method re-canonicalises the input images and then same model recovers geometrically faithful reconstruction. (ROVER)

### 4.4 Can ROVER Mitigate Upright Bias?

Having established the existence and origin of UprightBias, we investigate whether its effects can be mitigated entirely at inference time specifically, whether canonicalizing input orientations before reconstruction is sufficient to recover performance without any architectural modification or retraining. We compare each model in two settings: *Baseline*, where rotated inputs are provided directly to the original model, and *Ours*, where inputs are first canonicalized using ROVER and then passed to the unchanged model. Both settings are averaged over the same rotation-fraction sweep used in section 4.2. Analogous experiments with are included in the supplementary.

Table 3: **Point map estimation** on 7-Scenes, NRGB-D, and RotMV with ROVER. Results are averaged across the full rotation-fraction sweep per scene.

| | 7-Scenes | | | | | | NRGB-D | | | | | | RotMV | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | |
| **Method** | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours |
| VGGT | 0.06 ± 0.03 | **0.03** ± 0.01 | 0.13 ± 0.11 | **0.04** ± 0.01 | 0.61 ± 0.04 | **0.65** ± 0.01 | 0.05 ± 0.03 | **0.02** ± 0.00 | 0.09 ± 0.10 | **0.01** ± 0.00 | 0.72 ± 0.06 | **0.79** ± 0.01 | 0.02 ± 0.01 | **0.01** ± 0.00 | 0.04 ± 0.04 | **0.02** ± 0.02 | 0.68 ± 0.09 | **0.74** ± 0.05 |
| VGGT-Ω | 0.03 ± 0.01 | **0.02** ± 0.00 | 0.03 ± 0.01 | **0.03** ± 0.01 | 0.67 ± 0.01 | **0.67** ± 0.01 | 0.02 ± 0.01 | **0.01** ± 0.01 | **0.01** ± 0.00 | **0.01** ± 0.00 | 0.79 ± 0.02 | **0.81** ± 0.02 | **0.01** ± 0.00 | **0.01** ± 0.00 | **0.02** ± 0.01 | **0.02** ± 0.01 | 0.71 ± 0.06 | **0.72** ± 0.04 |
| $\pi^3$ | 0.07 ± 0.02 | **0.03** ± 0.01 | 0.13 ± 0.10 | **0.04** ± 0.02 | 0.62 ± 0.03 | **0.67** ± 0.01 | 0.05 ± 0.02 | **0.02** ± 0.01 | 0.09 ± 0.08 | **0.01** ± 0.01 | 0.73 ± 0.04 | **0.80** ± 0.01 | 0.04 ± 0.03 | **0.01** ± 0.01 | 0.05 ± 0.05 | **0.02** ± 0.01 | 0.70 ± 0.08 | **0.76** ± 0.05 |
| MapA. | 0.10 ± 0.02 | **0.08** ± 0.01 | 0.19 ± 0.08 | **0.12** ± 0.01 | 0.59 ± 0.03 | **0.63** ± 0.01 | 0.13 ± 0.03 | **0.08** ± 0.01 | 0.16 ± 0.13 | **0.07** ± 0.01 | 0.67 ± 0.06 | **0.73** ± 0.01 | 0.61 ± 0.25 | **0.46** ± 0.17 | 1.33 ± 0.89 | **0.95** ± 0.51 | 0.67 ± 0.11 | **0.75** ± 0.09 |
| DA3 | 0.07 ± 0.03 | **0.04** ± 0.01 | 0.10 ± 0.08 | **0.05** ± 0.02 | 0.64 ± 0.03 | **0.67** ± 0.01 | 0.06 ± 0.04 | **0.02** ± 0.01 | 0.10 ± 0.15 | **0.01** ± 0.00 | 0.73 ± 0.07 | **0.81** ± 0.02 | 0.61 ± 0.31 | **0.25** ± 0.18 | 0.83 ± 0.81 | **0.34** ± 0.40 | 0.69 ± 0.08 | **0.76** ± 0.04 |

Table 4: **Camera pose estimation** on 7-Scenes, NRGB-D, and RotMV before and after ROVER. Results are averaged across the full rotation-fraction sweep.

| | 7-Scenes | | | | | | NRGB-D | | | | | | RotMV | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | |
| **Method** | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours |
| VGGT | 77.49 ± 19.18 | **98.63** ± 2.38 | 74.92 ± 17.26 | **94.50** ± 5.76 | 53.97 ± 20.87 | **78.51** ± 6.64 | 90.49 ± 12.95 | **100.00** ± 0.00 | 90.90 ± 10.15 | **99.89** ± 0.16 | 78.22 ± 17.15 | **97.13** ± 0.95 | 88.46 ± 15.14 | **96.85** ± 3.31 | 90.94 ± 12.59 | **97.99** ± 2.30 | 72.17 ± 19.15 | **84.15** ± 8.34 |
| VGGT-Ω | 99.20 ± 2.58 | **99.91** ± 0.39 | 96.17 ± 4.34 | **98.27** ± 1.24 | 78.13 ± 7.77 | **83.63** ± 3.19 | 100.00 ± 0.00 | **100.00** ± 0.00 | 99.39 ± 0.76 | **99.96** ± 0.05 | 93.01 ± 3.94 | **98.02** ± 0.90 | 98.68 ± 2.19 | **98.86** ± 1.60 | 99.57 ± 0.79 | **99.83** ± 0.23 | 83.57 ± 7.66 | **85.60** ± 6.62 |
| $\pi^3$ | 78.60 ± 13.68 | **98.07** ± 3.00 | 76.68 ± 14.05 | **96.75** ± 3.07 | 57.05 ± 15.50 | **80.71** ± 5.41 | 91.28 ± 9.31 | **99.90** ± 0.37 | 90.49 ± 7.71 | **99.82** ± 0.44 | 79.09 ± 10.72 | **95.21** ± 1.34 | 92.92 ± 10.42 | **98.94** ± 1.48 | 93.95 ± 8.61 | **99.82** ± 0.29 | 74.62 ± 13.22 | **85.25** ± 6.60 |
| MapA. | 74.29 ± 16.49 | **98.10** ± 2.67 | 63.91 ± 14.73 | **87.00** ± 5.98 | 36.84 ± 16.90 | **62.77** ± 7.72 | 75.81 ± 21.31 | **99.90** ± 0.65 | 71.12 ± 19.26 | **98.49** ± 1.20 | 46.80 ± 22.84 | **87.22** ± 3.78 | 68.55 ± 24.05 | **98.43** ± 2.30 | 73.57 ± 18.96 | **98.00** ± 1.64 | 44.56 ± 25.06 | **80.73** ± 7.92 |
| DA3 | 82.47 ± 14.36 | **98.22** ± 2.68 | 79.14 ± 12.35 | **93.44** ± 5.67 | 58.67 ± 16.12 | **80.52** ± 5.99 | 89.51 ± 14.87 | **99.86** ± 0.42 | 88.56 ± 12.51 | **99.79** ± 0.60 | 73.37 ± 18.49 | **98.51** ± 1.62 | 80.48 ± 20.78 | **97.31** ± 4.81 | 85.23 ± 15.35 | **97.94** ± 3.53 | 60.89 ± 22.57 | **82.93** ± 9.69 |

Table 3 shows that ROVER consistently improves point-map estimation across all models and datasets. On 7-Scenes, our method provides gains across all models: Comp. errors drop by up to $3\times$ (VGGT: $0.13$ to $0.04$; $\pi^3$: $0.13$ to $0.04$), and N.C. improves consistently, with the largest gain for $\pi^3$ ($0.62$ to $0.67$). Variance is also reduced, indicating more stable predictions by ROVER. On NRGB-D, VGGT-Ω and DA3 achieve N.C. gains of $0.02$ and $0.08$ respectively, while Comp. errors fall for several models (e.g., VGGT: $0.09$ to $0.01$). On RotMV, ROVER improves point-map quality across all models: N.C. gains are observed for VGGT-Ω ($0.71$ to $0.72$), VGGT ($0.68$ to $0.74$), and DA3 ($0.69$ to $0.76$), while Accuracy improves for $\pi^3$ ($0.04$ to $0.01$) and MapA. ($0.61$ to $0.46$).

Table 4 shows that ROVER yields improvements for camera pose estimation across all datasets 7-Scenes, NRGB-D and RotMV, with large gains in $RAcc_{30}$, $TAcc_{30}$, and $AUC_{30}$ for all models. On 7-Scenes, $AUC_{30}$ improvements are observed for all models. MapAnything improved from $36.84$ to $62.77$ whereas VGGT improved from $53.97$ to $78.51$, and $\pi^3$ improved from $57.05$ to $80.71$, while variance is dramatically reduced across all models. On NRGB-D VGGT-Ω improves from $93.01$ to $98.02$ in $AUC_{30}$, and VGGT from $78.22$ to $97.13$, $\pi^3$ from $79.09$ to $95.21$, and DA3 from $73.37$ to $98.51$ after canonicalization. In RotMV $AUC_{30}$ improves for DA3 ($60.89$ to $82.93$), $\pi^3$ ($74.62$ to $85.25$), and VGGT-Ω ($83.57$ to $85.60$), while $RAcc_{30}$ gains for MapA. ($68.55 \rightarrow 98.43$), VGGT ($88.46$ to $96.85$), and $\pi^3$ ($92.92$ to $98.94$).

Figure 4 shows rotated images cause severe geometry degradation across all models, while our method re-canonicalize input orientations, enabling the same models to produce clean, geometrically faithful point clouds with better architectural details. Critically, this degradation is not a capacity limitation: it appears across architecturally diverse models including VGGT, $\pi^3$, DA3, and even VGGT-Ω, a 10B parameter model trained on $15\times$ more data than VGGT, demonstrating that neither architectural choice nor scale resolves the underlying orientation prior. We include addition qualitative results for different datasets in supplementary.

Table 5: Ablation of orientation correction strategies using VGGT-Ω on NRGB-D

| **Method** | $\Delta\theta \downarrow$ | N.C. ↑ | $AUC_{30}$ ↑ |
|---|---|---|---|
| Baseline | 57.70 ±60.20 | 0.79 ±0.02 | 93.01 ±3.94 |
| + DINOv1 | 95.90 ±49.2 | 0.730 ±0.028 | 77.52 ±7.09 |
| + Homography | 77.20 ±58.70 | 0.58 ±0.05 | 8.25 ±13.20 |
| + RotNet | 39.10 ±45.50 | 0.66 ±0.06 | 93.70 ±3.83 |
| + G.C. | 56.30 ±66.40 | 0.63 ±0.07 | 92.83 ±5.05 |
| + RotNet + G.C. | 35.80 ±50.40 | 0.68 ±0.06 | 94.43 ±3.97 |
| + ROVER | **10.30 ±15.20** | **0.81 ±0.02** | **98.03 ±1.02** |

Table 6: Performance on upright inputs (0% rotation) on NRGB-D. ROVER introduces no degradation when all upright images are used

| | N.C. ↑ | | $AUC_{30}$ ↑ | |
|---|---|---|---|---|
| **Method** | Base | Ours | Base | Ours |
| VGGT | **0.80 ±0.03** | **0.80 ±0.03** | **98.62 ±1.20** | 98.60 ±1.23 |
| VGGT-Ω | **0.83 ±0.04** | **0.83 ±0.04** | 99.27 ±0.56 | **99.28 ±0.52** |
| $\pi^3$ | **0.81 ±0.06** | **0.81 ±0.06** | **96.41 ±1.93** | **96.41 ±1.93** |
| MapA. | **0.75 ±0.05** | **0.75 ±0.05** | 93.92 ±4.30 | **93.94 ±4.30** |
| DA3 | **0.82 ±0.04** | **0.82 ±0.03** | **99.87 ±0.19** | 99.85 ±0.13 |

### 4.5 Why Does ROVER Work?

Having established that ROVER mitigates UprightBias, we investigate why its orientation-estimation strategy is effective by comparing it against alternative orientation correction methods: Homography-based matching [1] (Barath & Kukelova, 2019), RotNet (Gidaris et al., 2018), GeoCalib (Veicht et al., 2024), and their combination. Finally we additionally replace DINOv2 in ROVER with DINOv1 (Caron et al., 2021), which was trained without KoLeo regularizer to validate that the variance signal originates from the KoLeo. For each method, the estimated orientation is used to canonicalize inputs before reconstruction, and we evaluate orientation error $\Delta\theta$, point-map N.C., and camera-pose $AUC_{30}$ on NRGB-D using VGGT-Ω (results on other dataset in supplementary section A). Table 5 shows that ROVER achieves the lowest orientation error ($\Delta\theta = 10.30°$), highest N.C. (0.81) and $AUC_{30}$ (98.03), and lowest variance. RotNet reduces $\Delta\theta$ to $39.10°$ but degrades N.C. from 0.79 to 0.66, while combining RotNet with GeoCalib reduces the error to $35.80°$ but remains below ROVER in both N.C. (0.68 vs. 0.81) and $AUC_{30}$ (94.43 vs. 98.03). DINOv1 also underperforms the uncorrected baseline ($AUC_{30}$: 77.52; N.C.: 0.730), supporting the role of the KoLeo regularizer in obtaining a reliable orientation signal. ROVER thus improves both orientation estimation and downstream reconstruction without requiring reference images, a trained rotation classifier, or geometric optimization. We additionally evaluate rotated-view rejection using RobustVGGT (Han et al., 2025) (section L): outlier rotated views are detected from internal features and excluded from reconstruction. We found that RobustVGGT fails to identify non-canonical images reliably when upright images are in minority.

Lastly, we perform an experiment to examine whether ROVER affects reconstruction when all input images are already upright. We pass the same upright images through the baseline model and ROVER and compare their reconstruction performance. Table 6 shows that ROVER introduces no meaningful performance penalty in this settingon already upright inputs, with N.C. is preserved exactly across all models, and $AUC_{30}$ varying changes at most within 0.02 points, well within measurement variance. We include results for other datasets in section A.4

## 5 Conclusion and Future Work

We identified UprightBias, an overlooked failure mode in feed-forward geometric reconstruction models, where rotations of input images cause severe degradation in reconstructed scene. We traced this failure to the majority upright orientation of the training data these models rely on. Building on the observation that DINOv2 patch embeddings exhibit maximal variance at the canonical upright orientation, we introduced ROVER, a training-free, test-time canonicalization pipeline that recovers feed-forward model performance without modifying any model weights. ROVER's orientation estimate is tied to the underlying encoder thus can make mistakes in symmetric scenes. We include discussion about failure case of ROVER in supplementary. Resolving UprightBias during pretraining remains an important direction for future work.

[1] Homography based matching needs reference upright image to determine relative rotation.

### AI USE STATEMENT

In this work, we used generative AI tools for coding assistance while evaluating the feed-forward reconstruction models studied in this paper. We did not use generative AI tools to design or provide feedback on our research methodology or experiments, to support qualitative or thematic data analysis, or to interpret results. None of the following applied to this work: using generative AI to create synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims or supply key steps toward proving them, to help write proofs, to propose or refine hypotheses, to assist with translation, or to clean or reformat datasets. Additionally, we used generative AI tools to help polish writing throughout the paper, restructure the table and refine the writing, and identify inconsistencies in terminology, notation, and cross-references across the manuscript. We have reviewed AI-assisted work before using it. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*, pp. 589–604, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58535-8. doi: 10.1007/978-3-030-58536-5_35. URL https://doi.org/10.1007/978-3-030-58536-5_35.

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6290–6301, June 2022.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. Introducing hot3d: An egocentric dataset for 3d hand and object tracking. *arXiv preprint arXiv:2406.09598*, 2024.

Daniel Barath and Zuzana Kukelova. Homography from two orientation-and scale-covariant features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1091–1099. IEEE, 2019.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision*, pp. 387–404. Springer, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Claudia Cuttano, Gabriele Trivigno, Christoph Reich, Daniel Cremers, Carlo Masone, and Stefan Roth. Insid3: Training-free in-context segmentation with dinov3. In *Third Workshop on Visual Concepts*, 2026.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2183–2191, 2019.

Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2019.

Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2018.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, October 2013. URL `https://www.microsoft.com/en-us/research/publication/real-time-rgb-d-camera-relocalization/`.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022.

Jisang Han, Sunghwan Hong, Jaewoo Jung, Wooseok Jang, Honggyu An, Qianqian Wang, Seungryong Kim, and Chen Feng. Emergent outlier view rejection in visual geometry grounded transformers. *arXiv preprint arXiv:2512.04012*, 2025.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6910–6920. IEEE, 2021.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Wenbo Huang, Jinghui Zhang, Zhenghao Chen, Guang Li, Lei Zhang, Yang Cao, Fang Dong, Takahiro Ogawa, and Miki Haseyama. Otter: Mitigating background distractions of wide-angle few-shot action recognition with enhanced rwkv. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5140–5148, 2026.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 406–413. IEEE, 2014.

Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, Jiuxiang Gu, Qixing Huang, Georgios Pavlakos, and Hao Tan. Megasynth: Scaling up 3d scene reconstruction with synthesized data. 2024.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R Walter. Do you know where your camera is? view-invariant policy learning with camera conditioning. *arXiv preprint arXiv:2510.02268*, 2025.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. *CVPR*, 2023.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL `https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/`.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Jinwoo Lee, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, Minhyuk Sung, and Junho Kim. CTRL-C: Camera calibration TRansformer with Line-Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3205–3215, 2023.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Muxin Liu, Tianbo Liu, Jing Xia, Xiaoyang Lyu, Xiaoshan Wu, Bo Wang, Peng Dai, Zhongrui Wang, Shaoshuai Shi, and Xiaojuan Qi. Optigeo: Efficient monocular geometry for embodied perception in optically challenging scenes. *arXiv preprint arXiv:2608.29881*, 2026a.

Shuhong Liu, Chenyu Bao, Ziteng Cui, Yun Liu, Xuangeng Chu, Lin Gu, Marcos V. Conde, Ryo Umagami, Tomohiro Hashimoto, Zijian Hu, Tianhan Xu, Yuan Gan, Yusuke Kurose, and Tatsuya Harada. Realx3d: A physically-degraded 3d benchmark for multi-view visual restoration and reconstruction. *International Journal of Computer Vision*, 134(8):382, 2026b. doi: 10.1007/s11263-026-02963-4.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

Zhaoyang Lv, Nicholas Charron, Pierre Moulon, Alexander Gamino, Cheng Peng, Chris Sweeney, Edward Miller, Huixuan Tang, Jeff Meissner, Jing Dong, et al. Aria everyday activities dataset. *arXiv preprint arXiv:2402.13349*, 2024.

Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, et al. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *European Conference on Computer Vision*, pp. 445–465. Springer, 2024.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4981–4991, 2023.

Mateusz Michalkiewicz, Sheena Bai, Mahsa Baktashmotlagh, Varun Jampani, and Guha Balakrishnan. Not all views are created equal: Analyzing viewpoint instabilities in vision foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9113–9123, 2025.

Bharath Raj Nagoor Kani and Noah Snavely. G3t up! gravity aligned coordinate frames simplify pointmap processing. *arXiv preprint*, 2026.

David Nordström, Johan Edstedt, Georg Bökman, Jonathan Astermark, Anders Heyden, Viktor Larsson, Mårten Wadenbäck, Michael Felsberg, and Fredrik Kahl. Loma: Local feature matching revisited. In *European Conference on Computer Vision*, pp. 637–655. Springer, 2026.

David Nordstrom, Johan Edstedt, Fredrik Kahl, and Georg Bokman. Who handles orientation? investigating invariance in feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 97–106, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20076–20086. IEEE, 2023.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2109.08238.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10901–10911, 2021a.

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 *International Conference on Computer Vision*, 2021b.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Johannes Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. 06 2016. doi: 10.1109/CVPR.2016.445.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2013.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Utkarsh Singhal, Ryan Feng, Stella X. Yu, and Atul Prakash. Test-time canonicalization by foundation models for robust perception. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=JMZ7mr19AK.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, July 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141964. URL https://doi.org/10.1145/1141911.1141964.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020a.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020b.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in neural information processing systems*, 33:1513–1524, 2020.

Kaihua Tang, Mingyuan Tao, Jiaxin Qi, Zhenguang Liu, and Hanwang Zhang. Invariant feature learning for generalized long-tailed classification. In *European Conference on Computer Vision*, pp. 709–726. Springer, 2022.

Kaihua Tang, Ziqing Xia, Xiaoxu Zheng, Xiaoxue Zhang, Michael Bi Mi, Zhan Xu, and Dave Zhenyu Chen. Breaking the horizontal prior: From long-tailed orientation bias to roll-robust monocular depth estimation. *arXiv preprint arXiv:2608.00678*, 2026.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528, 2011. doi: 10.1109/CVPR.2011.5995347.

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. doi: 10.1109/34.88573.

Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. Geocalib: Learning single-image calibration with geometric optimization. In *European Conference on Computer Vision*, pp. 1–20. Springer, 2024.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020a.

Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9773–9783, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26024–26035. IEEE, 2025.

Tianhe Wu, Jian Zou, Jie Liang, Lei Zhang, and Kede Ma. Visualquality-r1: Reasoning-induced image quality assessment via reinforcement learning to rank. *Advances in Neural Information Processing Systems*, 38:88167–88190, 2026.

Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos, 2024.

Wenqi Xian, Zhengqi Li, Matthew Fisher, Jonathan Eisenmann, Eli Shechtman, and Noah Snavely. Uprightnet: Geometry-aware camera orientation estimation from single images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9974–9983, 2019.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

Yi Zhang, Weichao Qiu, Qi Chen, Xiaolin Hu, and Alan Yuille. Unrealstereo: Controlling hazardous factors to analyze stereo vision. In *2018 International Conference on 3D Vision (3DV)*, pp. 228–237. IEEE, 2018.

Yunhan Zhao, Shu Kong, and Charless Fowlkes. Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15754–15763. IEEE, 2021.

Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023.

# APPENDIX

**A Additional results** **A-2**

A.1 Point Cloud and Pose Degradation with Increasing Fraction of Rotated Images . . . A-2

A.2 Performance on Additional Datasets . . . A-2

A.3 Results on RotMV . . . A-3

A.4 Results on Upright Images . . . A-4

A.5 Comparison with Other Orientation-Correction Methods . . . A-4

A.6 Effect of ROVER on Trajectory Degradation under Rotated Images . . . A-5

A.7 Latency and Computational Overhead of ROVER . . . B-6

**B Algorithm** **B-6**

**C Orientation Variance Plots** **C-8**

**D Sensitivity to Limited Pitch Rotations** **E-9**

**E Examples of Point Cloud Degradation** **E-9**

**F Sensitivity to Yaw Rotations** **G-10**

**G The RotMV Dataset** **G-10**

**H Visual Quality Assessment and Rejection Statistics** **I-12**

**I Failure Case** **I-13**

**J Evaluation Criteria** **J-16**

**K Training Data re-balancing** **K-19**

**L Outlier Detection using RobustVGGT** **L-20**

L.1 Relational Representation Matrix . . . L-21

L.2 Sequence-Level Rotation Detection . . . L-21

L.3 Sensitivity to the Rotation Prior . . . L-22

L.4 Limitations of RobustVGGT . . . L-22

# A ADDITIONAL RESULTS

## A.1 POINT CLOUD AND POSE DEGRADATION WITH INCREASING FRACTION OF ROTATED IMAGES

To complement the results presented in Table 1, we report the corresponding degradation results for the remaining datasets (Tanks & Temples, Mip-NeRF 360, DTU and RotMV) in Table A. These additional results consistently shows how point cloud reconstruction and pose estimation performance change as the fraction of rotated input images increases, extending the analysis across all evaluated datasets.

Table A: Point-map and camera-pose estimation performance under increasing fractions of rotated input views. Feed Forward models suffers consistent degradation as fraction of rotated frames increases

| | | **Point-map Estimation (N.C.)** | | | | | **Camera Pose Estimation ($AUC_{30}$)** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Rot.** | VGGT | VGGT-Ω | $\pi^3$ | MapA. | DA3 | VGGT | VGGT-Ω | $\pi^3$ | MapA. | DA3 |
| **Tanks&Temples** | 0% | 0.665 ±0.076 | 0.724 ±0.067 | 0.707 ±0.118 | 0.628 ±0.152 | 0.707 ±0.064 | 76.4 ±35.5 | 91.1 ±16.1 | 79.6 ±28.4 | 78.3 ±22.7 | 94.2 ±8.2 |
| | 25% | 0.641 ±0.030 | 0.725 ±0.017 | 0.684 ±0.032 | 0.613 ±0.029 | 0.664 ±0.039 | 66.6 ±8.6 | 87.9 ±3.2 | 67.7 ±5.5 | 56.5 ±8.9 | 62.6 ±17.0 |
| | 50% | 0.641 ±0.050 | 0.712 ±0.018 | 0.656 ±0.038 | 0.598 ±0.038 | 0.650 ±0.046 | 57.2 ±11.7 | 83.0 ±4.2 | 56.1 ±7.9 | 37.6 ±12.5 | 47.8 ±15.8 |
| | 75% | 0.625 ±0.046 | 0.703 ±0.017 | 0.646 ±0.031 | 0.595 ±0.052 | 0.626 ±0.036 | 49.5 ±11.7 | 79.0 ±7.1 | 48.5 ±8.6 | 27.2 ±9.8 | 34.5 ±15.4 |
| | 100% | 0.604 ±0.043 | 0.685 ±0.023 | 0.618 ±0.040 | 0.572 ±0.044 | 0.618 ±0.042 | 42.7 ±9.6 | 74.4 ±7.2 | 40.7 ±6.9 | 18.8 ±6.1 | 31.6 ±10.0 |
| **Mip-NeRF 360*** | 25% | 0.712 ±0.025 | 0.711 ±0.016 | 0.704 ±0.013 | 0.660 ±0.070 | 0.671 ±0.028 | 90.1 ±8.3 | 95.0 ±1.7 | 90.8 ±2.5 | 72.4 ±10.1 | 82.4 ±11.6 |
| | 50% | 0.685 ±0.020 | 0.690 ±0.013 | 0.676 ±0.015 | 0.638 ±0.059 | 0.656 ±0.024 | 87.2 ±5.9 | 92.6 ±2.8 | 85.3 ±3.6 | 59.2 ±12.4 | 72.4 ±11.2 |
| | 75% | 0.664 ±0.018 | 0.673 ±0.013 | 0.649 ±0.013 | 0.614 ±0.050 | 0.633 ±0.022 | 83.5 ±7.0 | 89.9 ±2.5 | 80.0 ±7.5 | 49.4 ±10.5 | 65.4 ±11.0 |
| | 100% | 0.643 ±0.011 | 0.654 ±0.014 | 0.623 ±0.011 | 0.591 ±0.051 | 0.615 ±0.020 | 74.9 ±7.8 | 87.4 ±2.4 | 73.8 ±4.7 | 42.7 ±9.9 | 63.1 ±9.7 |
| **DTU** | 0% | 0.611 ±0.048 | 0.634 ±0.020 | 0.621 ±0.032 | 0.577 ±0.062 | 0.585 ±0.054 | 77.5 ±18.7 | 93.2 ±6.9 | 85.6 ±11.6 | 33.0 ±23.0 | 59.8 ±25.5 |
| | 25% | 0.630 ±0.027 | 0.629 ±0.025 | 0.625 ±0.025 | 0.625 ±0.047 | 0.619 ±0.035 | 88.6 ±14.1 | 95.1 ±2.2 | 90.1 ±7.6 | 36.5 ±15.0 | 83.1 ±12.9 |
| | 50% | 0.625 ±0.030 | 0.632 ±0.012 | 0.632 ±0.015 | 0.598 ±0.039 | 0.590 ±0.043 | 83.9 ±16.8 | 94.6 ±3.2 | 89.9 ±5.5 | 41.8 ±18.7 | 68.1 ±22.3 |
| | 75% | 0.631 ±0.023 | 0.632 ±0.017 | 0.634 ±0.011 | 0.620 ±0.034 | 0.621 ±0.038 | 91.8 ±9.1 | 94.5 ±2.6 | 90.8 ±5.2 | 57.7 ±17.2 | 74.5 ±18.8 |
| | 100% | 0.619 ±0.017 | 0.621 ±0.011 | 0.615 ±0.011 | 0.568 ±0.029 | 0.587 ±0.028 | 79.6 ±21.9 | 93.7 ±3.4 | 84.7 ±10.4 | 39.6 ±17.7 | 63.4 ±32.8 |
| **RotMV*** | 25% | 0.735 ±0.048 | 0.755 ±0.028 | 0.738 ±0.054 | 0.749 ±0.090 | 0.749 ±0.038 | 88.5 ±7.2 | 91.7 ±3.6 | 86.1 ±6.2 | 68.9 ±16.4 | 80.3 ±10.0 |
| | 50% | 0.673 ±0.050 | 0.707 ±0.028 | 0.696 ±0.047 | 0.676 ±0.083 | 0.689 ±0.061 | 74.4 ±13.2 | 85.1 ±4.4 | 77.1 ±8.2 | 46.3 ±18.6 | 58.7 ±21.2 |
| | 75% | 0.644 ±0.056 | 0.687 ±0.014 | 0.668 ±0.045 | 0.639 ±0.079 | 0.651 ±0.063 | 66.4 ±16.1 | 80.5 ±3.9 | 69.0 ±11.4 | 36.4 ±14.5 | 48.4 ±19.8 |
| | 100% | 0.626 ±0.049 | 0.670 ±0.015 | 0.654 ±0.039 | 0.603 ±0.052 | 0.659 ±0.045 | 56.6 ±17.6 | 75.5 ±5.0 | 63.9 ±10.7 | 21.4 ±11.8 | 52.3 ±14.4 |

*Mip-NeRF 360 and RotMV provide no ground-truth point clouds. For these datasets, both point-map and pose metrics are computed relative to each model's prediction on all upright (0%) input (see Section J) beacuse of its SOTA performance on upright images. We measure degradation from all upright prediction. The 0% setting would compare a prediction with itself and is hence omitted.

## A.2 PERFORMANCE ON ADDITIONAL DATASETS

We include results on the remaining datasets for comparison with ROVER and baselines, specifically MipNerf-360(Barron et al., 2022), Tanks and Temples Knapitsch et al. (2017), DTU Jensen et al. (2014) and ETH3D (Schöps et al., 2019; 2017). Table B and Table D presents point-map estimation results, whereas Table C and Table E present the corresponding camera pose estimation results. We observe ROVER helps improve the reconstruction by consistently re-canonicalizing the input images

before using the feed forward reconstruction model. In Table D and Table E, we observe ETH3D gains are consistent across all models and metrics with VGGT-Ω and $\pi^3$ showing consistent gains (95.07 to 95.26 and 81.89 to 83.80) in $AUC_{30}$, while VGGT shows increase in $TAcc_{30}$ (87.57 to 89.80). Overall, our approach consistently recovers camera pose estimation quality under arbitrary in-plane rotations, with gains scaling with the severity of baseline degradation.

Table B: Point map estimation on Tanks&Temples and Mip-NeRF 360 datasets. ROVER consistently improves performance of feed forward models by canocalizing input images.

| | **Tanks&Temples** | | | | | | **Mip-NeRF 360** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | |
| **Method** | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours |
| VGGT | 0.322 ± 0.098 | **0.247** ± 0.045 | 0.594 ± 0.336 | **0.539** ± 0.214 | 0.629 ± 0.055 | **0.648** ± 0.046 | 0.021 ± 0.006 | **0.014** ± 0.003 | 0.040 ± 0.023 | **0.019** ± 0.004 | 0.678 ± 0.034 | **0.726** ± 0.025 |
| VGGT-Ω | 0.190 ± 0.048 | **0.157** ± 0.025 | 0.143 ± 0.058 | **0.126** ± 0.032 | 0.707 ± 0.028 | **0.719** ± 0.021 | 0.017 ± 0.004 | **0.014** ± 0.002 | 0.017 ± 0.005 | **0.013** ± 0.002 | 0.684 ± 0.028 | **0.714** ± 0.020 |
| $\pi^3$ | 0.291 ± 0.100 | **0.199** ± 0.041 | 0.415 ± 0.173 | **0.347** ± 0.081 | 0.652 ± 0.049 | **0.689** ± 0.032 | 0.042 ± 0.015 | **0.026** ± 0.005 | 0.062 ± 0.035 | **0.034** ± 0.009 | 0.665 ± 0.037 | **0.711** ± 0.023 |
| MapA. | 0.340 ± 0.107 | **0.297** ± 0.051 | 0.808 ± 0.398 | **0.776** ± 0.253 | 0.595 ± 0.049 | **0.622** ± 0.033 | 0.351 ± 0.144 | **0.260** ± 0.079 | 0.568 ± 0.288 | **0.407** ± 0.196 | 0.628 ± 0.068 | **0.665** ± 0.070 |
| DA3 | 0.407 ± 0.155 | **0.275** ± 0.063 | 0.534 ± 0.335 | **0.335** ± 0.117 | 0.641 ± 0.049 | **0.674** ± 0.033 | 0.319 ± 0.169 | **0.162** ± 0.069 | 0.396 ± 0.381 | **0.156** ± 0.067 | 0.646 ± 0.035 | **0.683** ± 0.026 |

Table C: Camera pose estimation on Tanks&Temples and Mip-NeRF 360 datasets. We observe ROVER helps recover performance of feed forward models

| | **Tanks&Temples** | | | | | | **Mip-NeRF 360** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | |
| **Method** | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours |
| VGGT | 70.419 ± 16.405 | **86.028** ± 6.329 | 73.793 ± 11.929 | **84.134** ± 4.598 | 54.563 ± 14.883 | **72.567** ± 4.997 | 95.961 ± 5.195 | **100.000** ± 0.000 | 96.069 ± 4.732 | **99.872** ± 0.278 | 84.226 ± 9.689 | **93.703** ± 2.519 |
| VGGT-Ω | 98.780 ± 3.576 | **99.610** ± 1.424 | 93.049 ± 4.217 | **93.870** ± 2.607 | 81.307 ± 7.807 | **87.009** ± 4.083 | 99.956 ± 0.279 | **100.000** ± 0.000 | 99.541 ± 0.772 | **99.948** ± 0.145 | 91.373 ± 3.887 | **96.458** ± 1.191 |
| $\pi^3$ | 70.350 ± 11.942 | **87.248** ± 2.603 | 72.386 ± 9.104 | **83.846** ± 1.889 | 53.893 ± 13.126 | **74.584** ± 4.079 | 96.141 ± 5.166 | **100.000** ± 0.000 | 96.828 ± 4.049 | **99.768** ± 0.338 | 82.922 ± 8.558 | **92.765** ± 2.403 |
| MapA. | 59.961 ± 18.558 | **83.504** ± 7.109 | 63.738 ± 13.405 | **80.606** ± 6.166 | 36.130 ± 18.636 | **62.529** ± 10.936 | 87.015 ± 12.013 | **95.967** ± 5.741 | 86.318 ± 10.531 | **95.656** ± 5.312 | 56.776 ± 16.878 | **75.265** ± 9.465 |
| DA3 | 63.829 ± 19.325 | **87.285** ± 8.889 | 68.695 ± 15.722 | **86.463** ± 8.416 | 45.356 ± 21.409 | **75.814** ± 11.952 | 86.272 ± 11.855 | **98.685** ± 2.499 | 90.395 ± 9.753 | **99.036** ± 1.549 | 71.490 ± 15.075 | **91.869** ± 6.288 |

### A.3 Results on RotMV

RotMV provides a complementary evaluation to our controlled rotation synthesis benchmarks: rather than rendering rotations in camera space, it consists of real-world multi-view sequences captured at arbitrary orientations, without black-padding or rendering artifacts. This makes it a more challenging and authentic test of whether our method generalizes beyond controlled synthetic perturbations.

Table 3 and Table 4 in the main paper report point-map and camera pose estimation results respectively on RotMV. ROVER consistently improves over all base models across both tasks, confirming that the DINOv2 variance signal transfers reliably to real-world rotated captures. Notably,

Table D: Point map estimation on ETH3D and DTU datasets. ROVER consistently improves performance of feed forward models by canocalizing input images.

| | ETH3D | | | | | | DTU | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | | Acc. ↓ | | Comp. ↓ | | N.C. ↑ | |
| **Method** | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours |
| VGGT | 0.20 ± 0.07 | **0.18** ± 0.06 | 0.11 ± 0.04 | **0.08** ± 0.02 | 0.77 ± 0.04 | **0.79** ± 0.03 | 4.306 ± 5.377 | **4.222** ± 5.690 | **5.881** ± 9.698 | 6.488 ± 12.658 | **0.626** ± 0.034 | 0.624 ± 0.032 |
| VGGT-Ω | 0.07 ± 0.03 | **0.06** ± 0.02 | **0.05** ± 0.02 | **0.05** ± 0.02 | **0.85** ± 0.03 | **0.85** ± 0.03 | 4.306 ± 5.377 | **2.144** ± 1.976 | 5.881 ± 9.698 | **2.273** ± 2.620 | 0.626 ± 0.034 | **0.630** ± 0.019 |
| $\pi^3$ | 0.09 ± 0.03 | **0.08** ± 0.02 | 0.07 ± 0.02 | **0.06** ± 0.02 | 0.84 ± 0.02 | **0.85** ± 0.02 | 2.441 ± 2.464 | **2.240** ± 2.267 | 2.784 ± 4.381 | **2.493** ± 3.418 | **0.629** ± 0.022 | **0.629** ± 0.020 |
| MapA. | 0.16 ± 0.03 | **0.15** ± 0.03 | 0.39 ± 0.36 | **0.36** ± 0.29 | 0.75 ± 0.03 | **0.76** ± 0.02 | 12.201 ± 5.185 | **12.146** ± 5.140 | 19.941 ± 13.894 | **18.567** ± 12.641 | **0.602** ± 0.047 | **0.602** ± 0.043 |
| DA3 | **0.11** ± 0.02 | **0.11** ± 0.00 | 0.12 ± 0.01 | **0.10** ± 0.01 | **0.82** ± 0.01 | **0.82** ± 0.01 | 7.979 ± 5.250 | **7.441** ± 6.222 | 13.492 ± 15.661 | **10.556** ± 14.709 | 0.608 ± 0.046 | **0.614** ± 0.041 |

Table E: Camera pose estimation on ETH3D and DTU datasets. We observe ROVER helps recover performance of feed forward models

| | ETH3D | | | | | | DTU | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | | $RAcc_{30}$ ↑ | | $TAcc_{30}$ ↑ | | $AUC_{30}$ ↑ | |
| **Method** | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours | Base | Ours |
| VGGT | 93.03 ± 13.02 | **94.02** ± 13.86 | 87.57 ± 14.92 | **89.80** ± 14.96 | 70.54 ± 22.51 | **72.81** ± 22.81 | 94.376 ± 13.945 | **97.275** ± 9.516 | 94.969 ± 11.294 | **97.631** ± 7.825 | 86.101 ± 18.090 | **91.598** ± 12.317 |
| VGGT-Ω | 99.98 ± 0.08 | **100.00** ± 0.00 | 99.79 ± 0.26 | **99.82** ± 0.52 | 95.07 ± 2.80 | **95.26** ± 2.58 | 94.376 ± 13.945 | **99.790** ± 3.404 | 94.969 ± 11.294 | **99.791** ± 3.197 | 86.101 ± 18.090 | **95.274** ± 5.058 |
| $\pi^3$ | 96.35 ± 5.99 | **96.75** ± 7.70 | 95.51 ± 5.83 | **96.23** ± 6.34 | 81.89 ± 12.70 | **83.80** ± 12.70 | 98.968 ± 5.622 | **99.463** ± 3.242 | 98.770 ± 5.083 | **99.404** ± 2.926 | 88.982 ± 9.397 | **91.402** ± 6.224 |
| MapA. | 89.93 ± 12.44 | **90.24** ± 13.34 | 81.10 ± 16.54 | **81.87** ± 16.98 | 57.25 ± 16.32 | **57.72** ± 17.40 | 84.536 ± 21.317 | **90.301** ± 15.862 | 76.994 ± 21.675 | **81.775** ± 17.298 | 46.868 ± 22.298 | **50.393** ± 19.157 |
| DA3 | 87.74 ± 12.92 | **88.38** ± 16.78 | 88.16 ± 15.58 | **88.90** ± 18.02 | 79.62 ± 17.07 | **79.73** ± 20.59 | 89.745 ± 17.853 | **92.648** ± 15.361 | 89.612 ± 16.658 | **92.150** ± 14.054 | 73.804 ± 23.601 | **77.048** ± 22.030 |

the gains on RotMV are consistent with those observed on synthetic benchmarks, suggesting that its orientation estimates are robust to the additional photometric variation and capture noise present in real-world sequences. These results validate that our approach addresses UprightBias effectively without any retraining. The RotMV dataset is described in Section G.

### A.4 Results on Upright Images

To complement the upright-input ablation reported in Table 6, we extend the evaluation to Tanks & Temples, NRGB-D and DTU in Table F. ROVER imposes minimal to no degradation in point cloud quality and pose estimation when all input images are already upright. Across most model-dataset combinations, ROVER closely matches the baseline for both N.C. and $AUC_{30}$, with differences generally negligible. Pose estimation exhibits a similar trend, with ROVER occasionally even marginally outperforming the baseline. Overall, these results indicate that applying ROVER to already-upright inputs largely preserves the performance of the underlying reconstruction models.

### A.5 Comparison with Other Orientation-Correction Methods

To complement the ablation in Table 5, we also include ablation results comparing our approach's orientation detection and correction accuracy against other common orientation-correction methods

Table F: Feed forward reconstruction when all images upright: N.C. and AUC@30, mean ± std across scenes, on Tanks & Temples, NRGBD and DTU.

| | Method | N.C. ↑ | | $AUC_{30}$ ↑ | |
|---|---|---|---|---|---|
| | | Ours | Baseline | Ours | Baseline |
| Tanks&Temples | VGGT | 0.66 ± 0.08 | 0.66 ± 0.08 | 76.13 ± 35.27 | 76.41 ± 35.50 |
| | VGGT-Ω | 0.72 ± 0.07 | 0.72 ± 0.07 | 90.43 ± 16.70 | 91.10 ± 16.06 |
| | $\pi^3$ | 0.71 ± 0.13 | 0.71 ± 0.12 | 78.28 ± 30.13 | 79.56 ± 28.37 |
| | MapA | 0.60 ± 0.14 | 0.63 ± 0.15 | 73.36 ± 32.26 | 78.27 ± 22.66 |
| | DA3 | 0.70 ± 0.07 | 0.70 ± 0.07 | 94.57 ± 7.51 | 94.36 ± 8.18 |
| NRGBD | VGGT | 0.80 ± 0.03 | 0.80 ± 0.03 | 98.60 ± 1.23 | 98.62 ± 1.20 |
| | VGGT-Ω | 0.83 ± 0.04 | 0.83 ± 0.04 | 99.28 ± 0.52 | 99.27 ± 0.56 |
| | $\pi^3$ | 0.81 ± 0.06 | 0.81 ± 0.06 | 96.41 ± 1.93 | 96.41 ± 1.93 |
| | MapA | 0.75 ± 0.05 | 0.75 ± 0.05 | 93.94 ± 4.30 | 93.92 ± 4.30 |
| | DA3 | 0.82 ± 0.03 | 0.82 ± 0.04 | 99.85 ± 0.13 | 99.87 ± 0.19 |
| DTU | VGGT | 0.64 ± 0.02 | 0.65 ± 0.02 | 97.36 ± 5.80 | 99.69 ± 0.43 |
| | VGGT-Ω | 0.63 ± 0.02 | 0.65 ± 0.02 | 97.60 ± 1.21 | 99.69 ± 0.43 |
| | $\pi^3$ | 0.63 ± 0.01 | 0.64 ± 0.01 | 95.52 ± 1.66 | 96.01 ± 1.44 |
| | MapA | 0.61 ± 0.04 | 0.62 ± 0.04 | 68.45 ± 13.21 | 72.74 ± 12.38 |
| | DA3 | 0.63 ± 0.03 | 0.64 ± 0.03 | 94.99 ± 9.10 | 98.92 ± 0.94 |

in Table G for the 7-Scenes and Tanks&Temples datasets. ROVER achieves the lowest residual rotation error on both 7-Scenes and Tanks&Temples, while also obtaining the best N.C. and $AUC_{30}$. This shows that more accurate orientation correction translates directly to stronger downstream geometry and pose estimation. Homography-based orientation error correlates strongly with the number of inlier SIFT correspondences between each view and the fixed reference (first) image used to estimate its relative rotation; several view pairs in our multi-view scenes yield too few reliable correspondences, producing an unstable homography and a poor rotation estimate.

Table G: Rotation-correction front-ends on VGGT-Ω, mean ± std over all rotation levels and scenes. $\Delta\theta$ is the mean per-image residual rotation error (↓); for Baseline (no correction) it is the deviation of the applied rotation from upright.

| | **7-Scenes** | | | **Tanks&Temples** | | |
|---|---|---|---|---|---|---|
| **Method** | $\Delta\theta$ ↓ | N.C. ↑ | $AUC_{30}$ ↑ | $\Delta\theta$ ↓ | N.C. ↑ | $AUC_{30}$ ↑ |
| Baseline | 58.1±60.5 | 0.667±0.013 | 78.13±7.77 | 58.3±60.6 | 0.707±0.028 | 81.31±7.81 |
| + DINOv1 | 90.7±59.9 | 0.651±0.015 | 68.89±5.49 | 126.0±50.2 | 0.658±0.024 | 59.17±6.07 |
| + Homography | 72.5±57.3 | 0.580±0.037 | 8.64±12.56 | 74.2±59.4 | 0.590±0.047 | 7.81±10.61 |
| + RotNet | 32.5±40.0 | 0.622±0.027 | 73.15±6.56 | 43.3±43.8 | 0.636±0.056 | 77.80±7.08 |
| + G.C. | 58.6±62.5 | 0.597±0.037 | 73.85±10.52 | 57.7±65.0 | 0.626±0.058 | 79.36±9.88 |
| + RotNet + G.C. | 31.9±41.9 | 0.623±0.026 | 73.33±5.83 | 38.3±47.2 | 0.663±0.047 | 81.40±7.93 |
| + ROVER | **14.8**±21.7 | **0.668**±0.007 | **83.63**±3.19 | **15.6**±27.5 | **0.719**±0.021 | **86.92**±4.13 |

### A.6 Effect of ROVER on Trajectory Degradation under Rotated Images

As the fraction of rotated images increases, the degradation observed in VGGT-Ω is also reflected in the estimated camera trajectory as seen in Figure 2 and also in Section A.1 . We therefore evaluate the relative pose error (RPE) translation drift and absolute trajectory error (ATE) under the same progressively increasing rotation setting for VGGT-Ω both with and without ROVER on the 7-scenes dataset. As shown in Fig. Figure A, VGGT-Ω exhibits increasing trajectory error with more rotated images, whereas applying ROVER substantially suppresses this degradation and maintains trajectory estimates closer to COLMAP.

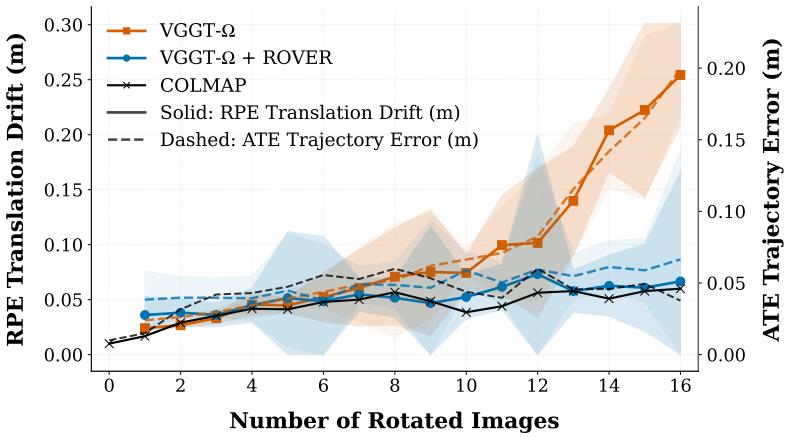



Figure A: RPE translation and ATE trajectory error with increasing fraction of rotated input images, comparing ROVER with VGGT-Ω, base model VGGT-Ω and COLMAP

### A.7 Latency and Computational Overhead of ROVER

Table H: Latency and computational overhead of ROVER across VGGT, VGGT-Ω, $\pi^3$, MapAnything and DepthAnything3

| Model | Base Latency (seconds) | ROVER Correction (seconds) | Total Latency (seconds) | Added Overhead (%) |
|---|---|---|---|---|
| VGGT | $1.769_{\pm 0.017}$ | $0.433_{\pm 0.004}$ | $2.169_{\pm 0.027}$ | $+24.5\%$ |
| VGGT-Ω | $1.400_{\pm 0.006}$ | $0.424_{\pm 0.003}$ | $1.786_{\pm 0.009}$ | $+30.3\%$ |
| $\pi^3$ | $1.296_{\pm 0.004}$ | $0.428_{\pm 0.001}$ | $1.701_{\pm 0.034}$ | $+33.0\%$ |
| MapA | $5.789_{\pm 0.038}$ | $0.430_{\pm 0.004}$ | $6.203_{\pm 0.033}$ | $+7.4\%$ |
| DA3 | $2.003_{\pm 0.010}$ | $0.431_{\pm 0.003}$ | $2.444_{\pm 0.016}$ | $+21.5\%$ |

Additionally we compute latency introduced by the ROVER in Table H for 16 images. For all of the models, the introduced latency remains less than $50\%$ of the base model latency, with the observed overhead ranging from $7.4\%$ to $33.0\%$. The ROVER correction itself incurs a nearly constant latency of approximately $0.43$ seconds across all models, irrespective of the underlying model. The relative overhead decreases for models with higher base latency, with Map-Anything incurring only $7.4\%$ overhead, demonstrating the low computational cost of ROVER.

## B Algorithm

Algorithm Algorithm 1 provides the pseudocode for controlled rotation synthesis. We first fit a 3DGS representation to each multi-view captured image set. We then sample an angle $\theta \sim \mathcal{U}(0°, 360°)$, render the corresponding rolled image, and score its reconstruction quality $q$ with VisualQuality-R1 (Wu et al., 2026). Captures falling below the predefined threshold $\tau$ are denoised with Difix3D single-step denoiser to remove artifacts, after which the remaining scenes are rendered directly. The renderings are scored again with reconstruction quality $q$ with VisualQuality-R1 to remove images with obvious artifacts. A final manual pass removes images with any residual artifacts. Orientation is then recovered through (Algorithm Algorithm 2). For every candidate orientation $\theta \in \Theta$ in a predefined discrete set $\Theta = \{0°, \delta, \ldots, 360° - \delta\}$, the input image is rotated by $\theta$ and passed through the frozen DINOv2 encoder $\phi$ to obtain the corresponding patch embeddings, $\mathbf{Z}_\theta = \phi(\mathbf{I}^\theta) = \{\mathbf{z}_1^\theta, \mathbf{z}_2^\theta, \ldots, \mathbf{z}_M^\theta\}$, where $\mathbf{I}^\theta$ denotes the image rotated by $\theta$, and $\mathbf{z}_j^\theta \in \mathbb{R}^D$ denotes the embedding corresponding to the $j$-th image patch. Each candidate orientation is scored by the variance of its patch embeddings about their centroid, $V(\theta) = \frac{1}{M}\sum_{j=1}^{M} \left\|\mathbf{z}_j^\theta - \bar{\mathbf{z}}^\theta\right\|_2^2$, where

**Algorithm 1** Rendering-Based Rotated Image Generation

**Input:** Multi-view images $\{\mathbf{I}_i\}$, poses $\{\mathbf{P}_i\}$, quality threshold $\tau$
**Output:** Rotated images $\{\mathbf{I}_i^{\theta_i}\}$

Fit 3DGS representation $\mathcal{G}$ on $\{\mathbf{I}_i, \mathbf{P}_i\}$
**for** each view $i$ **do**
  Sample $\theta_i \sim \mathcal{U}(0°, 360°)$
  Compute $\mathbf{P}_i^{\theta_i}$
  $\mathbf{I}_i^{\theta_i} \leftarrow \mathrm{Render}(\hat{\mathcal{G}}, \mathbf{P}_i^{\theta_i})$
  $q \leftarrow \text{VisualQuality-R1}(\mathbf{I}_i^{\theta_i})$
  **if** $q \leq \tau$ **then**
    $\mathbf{I}_i^{\theta_i} \leftarrow \text{Difix3D_Denoise}(\mathbf{I}_i^{\theta_i})$
    **if** $\text{VisualQuality-R1}(\mathbf{I}_i^{\theta_i}) \leq \tau$ **then**
      Exclude view $i$ from evaluation
    **end if**
  **end if**
**end for**
**for** each image $\mathbf{I} \in \{\mathbf{I}_i^{\theta_i}\}$ **do**
  Manually verify $\mathbf{I}$ and remove it from the set if visible artifacts are observed
**end for**

**Algorithm 2** ROVER: Variance-based Canonical Orientation Estimation

**Input:** Image $\mathbf{I}$, frozen DINOv2 encoder $\phi$, coarse step $\Delta_c = 15°$, fine window $\Delta_f = 7°$
**Output:** Canonicalized image $\mathbf{I}^{\mathrm{can}}$ with estimated roll $\hat{\theta}$

*// Coarse stage*
$\Theta_{\text{coarse}} \leftarrow \{k \cdot \Delta_c \mid k = 1, \ldots, \lfloor 360°/\Delta_c \rfloor\}$
**for** each $\theta \in \Theta_{\text{coarse}}$ **do**
  $V(\theta) \leftarrow \textsc{EmbedVar}^3(\mathbf{I}, \theta, \phi)$
**end for**
$\theta_c \leftarrow \arg\max_{\theta \in \Theta_{\text{coarse}}} V(\theta)$
*// Fine stage*
$\Theta_{\text{fine}} \leftarrow \{\theta_c + \delta \mid \delta \in [-\Delta_f, \Delta_f],\ \delta \in \mathbb{Z}\}$
**for** each $\theta \in \Theta_{\text{fine}}$ **do**
  $V(\theta) \leftarrow \textsc{EmbedVar}^3(\mathbf{I}, \theta, \phi)$
**end for**
$\hat{\theta} \leftarrow \arg\max_{\theta \in \Theta_{\text{fine}}} V(\theta)$
*// Canonicalization*
$\mathbf{I}^{\mathrm{can}} \leftarrow \mathrm{Rotate}(\mathbf{I}, -\hat{\theta})$
**return** $\mathbf{I}^{\mathrm{can}}, \hat{\theta}$

**Algorithm 3** EMBEDVAR: Patch Embedding Variance Computation

**Input:** Image $\mathbf{I}$, candidate orientation $\theta$, frozen DINOv2 encoder $\phi$
**Output:** Patch embedding variance $V(\theta)$

$\mathbf{Z}_\theta \leftarrow \phi(\mathrm{Rotate}(\mathbf{I}, \theta))$ $\{M$ patch embeddings $\in \mathbb{R}^D\}$
$\bar{\mathbf{z}}^\theta \leftarrow \frac{1}{M}\sum_{j=1}^{M} \mathbf{z}_j^\theta$ {Mean patch embedding}
**return** $\frac{1}{M}\sum_{j=1}^{M} \|\mathbf{z}_j^\theta - \bar{\mathbf{z}}^\theta\|_2^2$

$\bar{\mathbf{z}}^\theta = \frac{1}{M}\sum_{j=1}^{M} \mathbf{z}_j^\theta$ is the centroid of the patch embeddings. Since the backbone is pre-trained predominantly on upright imagery, patch features are most strongly differentiated at the canonical orientation and collapse toward the centroid as the rotation departs from it. The orientation estimation operator is therefore defined as $E(\mathbf{I}) = \arg\max_{\theta \in \Theta} V(\theta)$, and the maximizing candidate gives the predicted roll. We use DINOv2 embeddings to determine the canonical orientation in our approach. Previous works (Cuttano et al., 2026) points out that DINOv3 features can exhibit spurious correspondences between image pairs regardless of their semantic similarity. In particular, feature

similarities between semantically unrelated images exhibit systematic activations aligned with absolute spatial coordinates, indicating a strong positional bias. This positional bias in DINOv3 hinders its ability to accurately identify canonical rotation angle using our variance-maximization approach which makes DINOv2 as an encoder $\phi$.

## C Orientation Variance Plots

We include plot of variance of token embeddings w.r.t. the image orientation in fig. B for datasets 7-Scenes, NRGB-D, Tanks and Temples and RotMV. To analyze the orientation dependence of DINOv2 representations, we synthetically rotate each canonical image over $0°$–$360°$ in $15°$ increments and compute the same patch-token spatial variance used by our method, excluding padded regions introduced by rotation. To isolate orientation-dependent changes from image-specific differences in absolute variance, each image's variance curve is independently min–max normalized across candidate rotations before aggregation.

Figure B shows the mean normalized variance across all images, with $\pm 1$ standard deviation. We use upright images here to calculate variance. We show that for all datasets, the variance of token embeddings consistently achieves its maximum value at canonical orientation ($0°$ in this case, as we used upright images), while non-canonical rotations produce substantially lower values with weaker secondary peaks near multiples of $90°$. This supports the use of DINOv2 patch-embedding variance as a reliable signal for identifying canonical upright orientation.

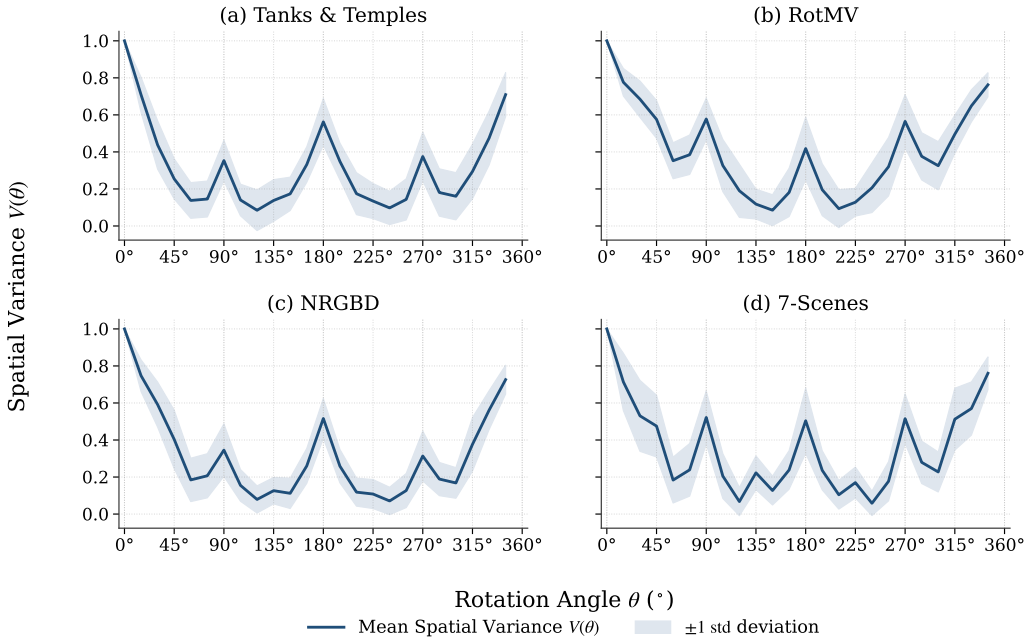



Figure B: Variance as a function of rotation angle across the RotMV, NRGB-D, 7-Scenes, and Tanks and Temples datasets. We observe that ROVER consistently helps estimate the upright orientation of the images

We also plot the variance for images rotated by $225°$ in fig. C for Tanks and Temples, NRGB-D and 7-Scenes. Each rotated image is further rotated by candidate angles over $[0°, 360°)$ in $15°$ increments, and the variance is maximized by the candidate that restores the upright orientation ($135°$). We plot the mean normalized variance against the estimated rotation $\theta = 360° -$ candidate, so the peak appears at $\theta = 225°$ for all three datasets and the correction is simply a rotation by $-\theta$. In the upright setting, the peak occurs at $0°$, where no rotation (and hence no padding) is applied, so one could argue the peak is simply caused by the absence of padding. Here, leaving the image unrotated ($\phi = 0°$) does not give the maximum; instead, the peak occurs at $\phi = 135°$, which adds

padding(because of the rotation) but restores the upright view. This shows that the variance peak is driven by the upright orientation of the image content, not by the padding.

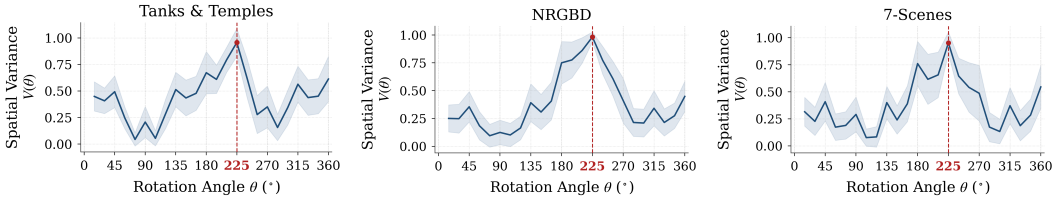



Figure C: ROVER recovers the true rotation angle. Input images are rotated by 225° from their upright orientation. We plot the normalized patch-embedding variance (mean ±1 std) against the estimated rotation $\theta = 360° - \phi$, where $\phi$ is the candidate correction. The peak (red dashed line) falls at $\theta = 225°$ for all three datasets, so rotating by $-\theta$ restores the upright view.

## D Sensitivity to Limited Pitch Rotations

We additionally evaluate sensitivity to camera rotations about the pitch axis on NRGB-D, 7-Scenes, and Tanks&Temples in Table I. Following the evaluation protocol described in Section 4.1, we evaluate the four non-zero rotation levels (25%, 50%, 75%, and 100% rotated views). Selected views are independently assigned a pitch perturbation from $[-5°, +5°]$; the corresponding camera pose is then perturbed and a new view is rendered using Section 3.1. Large perturbation leads to change in viewpoint of the scene as the camera will point to a different region in the scene unlike roll perturbation thus we restrict the perturbation to limited range. Across this limited range, performance remains close to the upright setting for most models and datasets. For instance, VGGT changes from 0.798 to 0.796 N.C. and from 98.36 to 98.17 $AUC_{30}$ on NRGB-D, while on Tanks&Temples the corresponding values change from 0.678 to 0.671 and from 91.62 to 91.20. Larger variation is observed in some cases, such as $\pi^3$ on Tanks&Temples, where N.C. decreases from 0.701 to 0.666, although pose accuracy remains comparatively stable. Overall, these results indicate sensitivity to small pitch perturbations over the evaluated range.

Table I: Point-map (N.C., ↑) and pose (AUC@30, ↑) accuracy, upright vs. rotated about the pitch axis, on NRGBD, 7-Scenes, and Tanks&Temples.

| | **NRGBD** | | | | **7-Scenes** | | | | **Tanks&Temples** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.C. ↑ | | AUC@30 ↑ | | N.C. ↑ | | AUC@30 ↑ | | N.C. ↑ | | AUC@30 ↑ | |
| **Method** | Up. | Rot. | Up. | Rot. | Up. | Rot. | Up. | Rot. | Up. | Rot. | Up. | Rot. |
| VGGT | 0.798 ± 0.031 | 0.796 ± 0.005 | 98.36 ± 1.44 | 98.17 ± 0.38 | 0.661 ± 0.038 | 0.662 ± 0.003 | 85.60 ± 6.73 | 84.68 ± 1.23 | 0.678 ± 0.054 | 0.671 ± 0.022 | 91.62 ± 14.60 | 91.20 ± 0.79 |
| VGGT-Ω | 0.827 ± 0.049 | 0.824 ± 0.007 | 99.22 ± 0.57 | 98.84 ± 0.43 | 0.670 ± 0.050 | 0.670 ± 0.003 | 88.07 ± 5.80 | 87.05 ± 1.20 | 0.684 ± 0.083 | 0.664 ± 0.016 | 92.88 ± 14.29 | 92.79 ± 0.36 |
| $\pi^3$ | 0.807 ± 0.077 | 0.805 ± 0.007 | 96.32 ± 2.74 | 95.86 ± 0.56 | 0.682 ± 0.050 | 0.683 ± 0.003 | 85.13 ± 8.09 | 84.59 ± 1.36 | 0.701 ± 0.056 | 0.666 ± 0.030 | 94.46 ± 8.71 | 94.20 ± 0.86 |
| MapA. | 0.741 ± 0.052 | 0.720 ± 0.012 | 93.86 ± 3.79 | 93.25 ± 0.84 | 0.631 ± 0.059 | 0.633 ± 0.010 | 76.22 ± 15.42 | 75.69 ± 1.82 | 0.614 ± 0.065 | 0.592 ± 0.030 | 92.48 ± 3.04 | 91.70 ± 1.29 |
| DA3 | 0.815 ± 0.043 | 0.810 ± 0.009 | 99.77 ± 0.37 | 99.74 ± 0.11 | 0.678 ± 0.041 | 0.678 ± 0.004 | 88.14 ± 5.95 | 87.44 ± 1.48 | 0.694 ± 0.057 | 0.683 ± 0.024 | 98.45 ± 1.16 | 97.87 ± 1.83 |

## E Examples of Point Cloud Degradation

We qualitatively illustrate the effect of in-plane roll on point-cloud reconstruction in Figures Figure D, Figure E Figure F across VGGT, VGGT-Ω, $\pi^3$, and DepthAnything3 (DA3). Under rotated inputs, the baseline reconstructions exhibit substantial geometric artifacts, including distorted structures, misaligned surfaces, and, most prominently, duplicated scene elements, as highlighted in the

zoomed regions. Applying ROVER consistently reduces these artifacts across the different backbones and produces substantially cleaner and more coherent point-cloud reconstructions despite the input images being arbitrarily rotated in-plane.

We additionally evaluate the setting where all input images are rotated by the same angle in Figure G, preserving their relative orientations. Surprisingly, models still fail to reconstruct the scene, demonstrating that the degradation is not caused by relative pose ambiguity between views but by a learned prior on absolute image orientation the model simply does not expect to see images that deviate from upright.

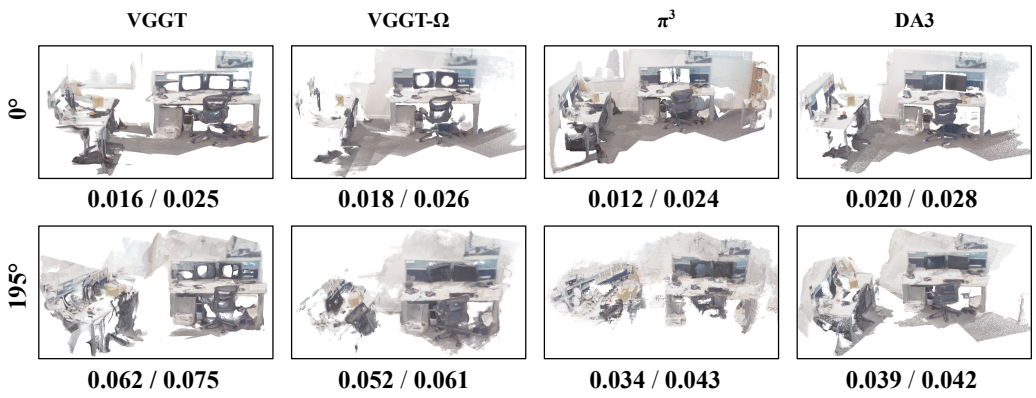



Figure G: **Degradation under uniform roll on ETH3D (*office*).** Point cloud reconstructions comparing clean upright views ($0°$, top row) with sequences where every input view is rotated identically by $195°$ (bottom row). Numbers denote Accuracy (Acc ↓) / Completeness (Comp ↓) in meters relative to ground truth . Degradation occurs despite identical relative poses between views, confirming reliance on an absolute upright prior.

## F Sensitivity to Yaw Rotations

We additionally evaluate sensitivity to camera rotations about the yaw axis on NRGB-D, 7-Scenes, and Tanks&Temples in Table J. Following the evaluation protocol described in Section 4.1, we evaluate the four non-zero rotation levels, with selected views independently assigned yaw perturbations from $[-20°, +20°]$. Large perturbation across yaw will lead to change in viewpoint of the scene which is not a manisfestation of UprightBias thus we restrict the evaluation to limited perturbation. The corresponding camera poses are perturbed and new views are rendered using Section 3.1. Overall, yaw perturbations introduce moderate degradation, particularly on Tanks&Temples. For example, $\pi^3$ decreases from $0.721$ to $0.667$ N.C., while MapAnything decreases from $0.627$ to $0.592$ N.C. and from $94.53$ to $92.57$ $\text{AUC}_{30}$.

## G The RotMV Dataset

As discussed in section 3, existing multi-view datasets consist predominantly of clean, upright images. Consequently, evaluations on these datasets are performed by introducing synthetic camera roll using the controlled rotation synthesis described in section 3.1 (algorithm 1). We present a novel benchmark consisting of seven real-world scenes (*chairs*, *foosball-table*, *building*, *rooftop*, *room*, *statue*, and *table-tennis*). We capture each scene using a camera ($4.84$ mm, $f/1.8$, $12$ MP, $4096 \times 3072$ resolution). Each scene is imaged from $16$ fixed viewpoints, and at each viewpoint, the camera is rolled in-plane through $12$ orientations spanning a full $360°$ (in $30°$ increments), resulting in $192$ images per scene.

For each viewpoint, we estimate the base camera pose from the upright ($0°$) image using VGGT (Wang et al., 2025a). The poses corresponding to the remaining $11$ roll angles are then derived from this base pose by applying the corresponding in-plane rotation to the camera orientation, while keeping the camera center and viewing direction unchanged. Thus, pose estimation is

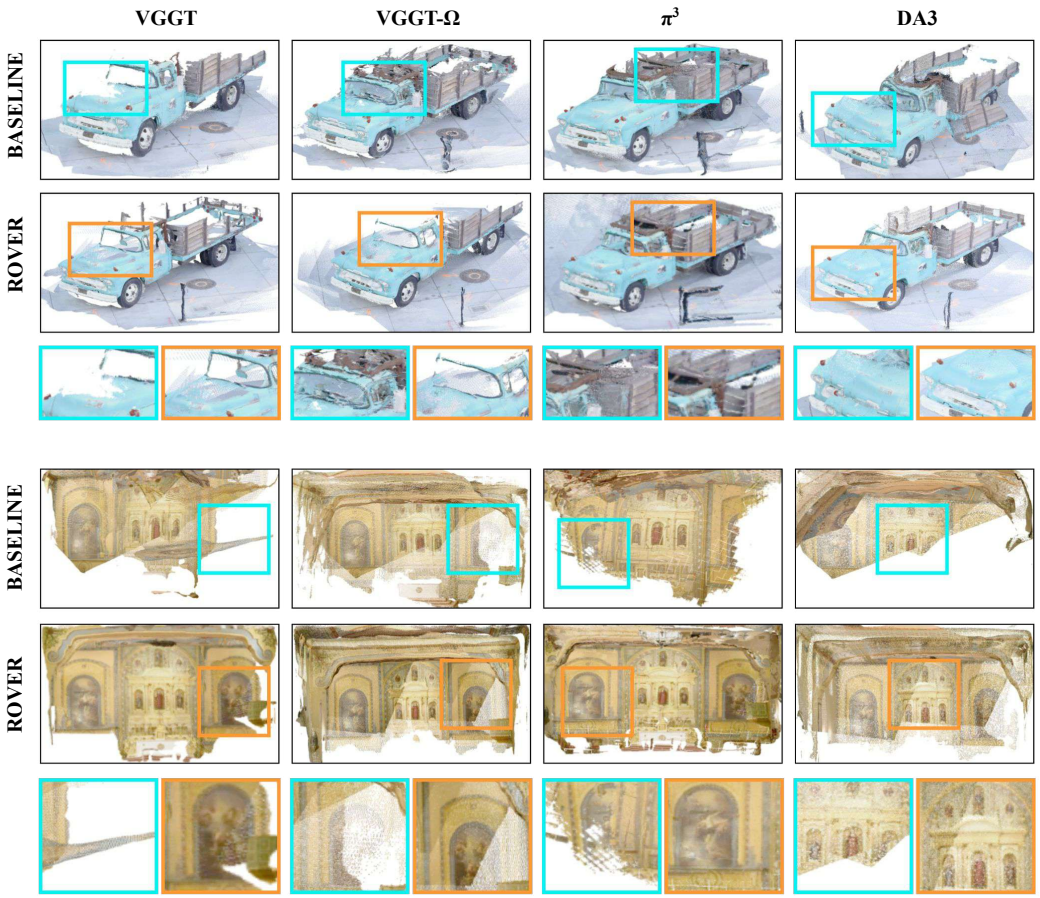



Figure D: **Qualitative comparison on the Tanks & Temples dataset** (Truck and Church scenes). Point cloud reconstructions across VGGT, VGGT-Ω, $\pi^3$, and DA3 comparing baseline and ROVER.

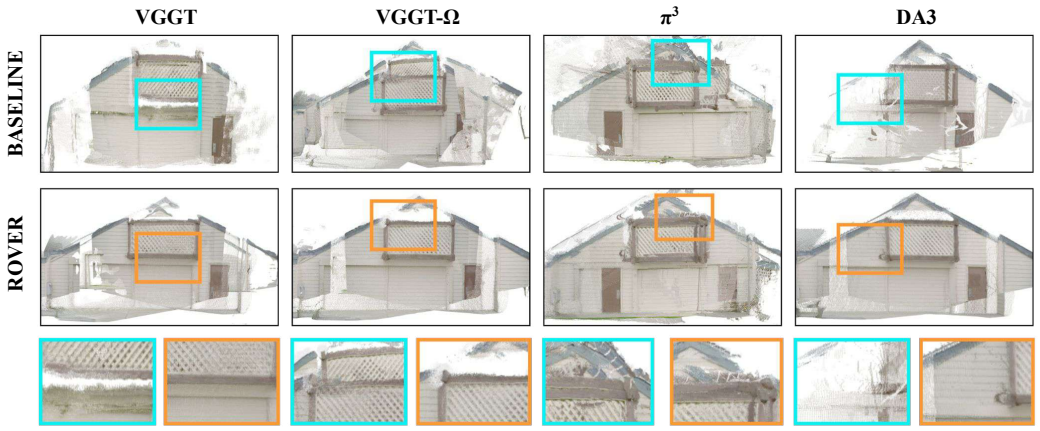



Figure E: **Qualitative comparison on the Barn scene** from Tanks & Temples dataset . Reconstruction comparisons across baseline and ROVER settings.

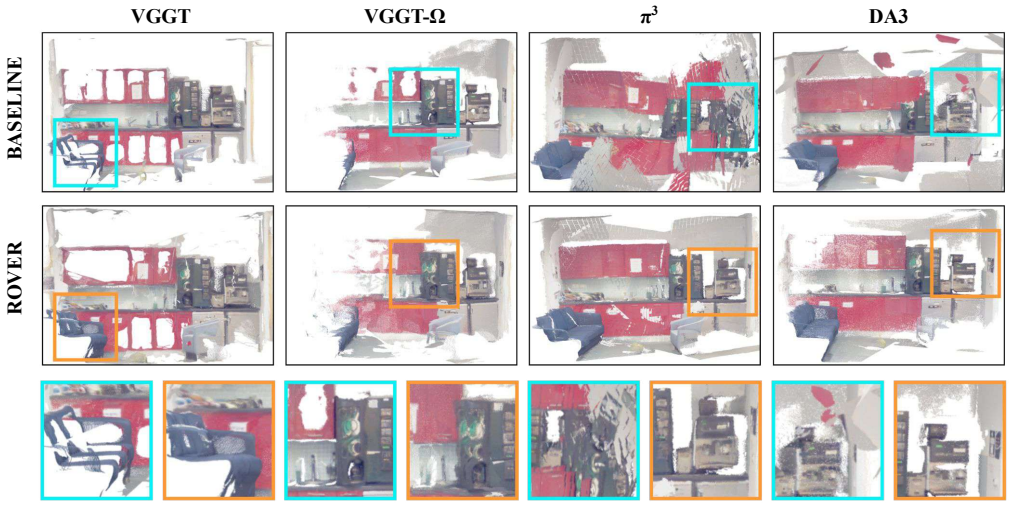



Figure F: **Qualitative comparison on the 7-Scenes dataset (Red Kitchen scene)**. Close-up comparisons showing geometric alignment and wall boundary recovery under ROVER.

Table J: Point-map (N.C., ↑) and pose ($AUC_{30}$, ↑) accuracy for upright inputs and inputs rotated about the yaw axis, on NRGBD, 7-Scenes, and Tanks&Temples.

| | NRGBD | | | | 7-Scenes | | | | Tanks&Temples | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.C. ↑ | | $AUC_{30}$ ↑ | | N.C. ↑ | | $AUC_{30}$ ↑ | | N.C. ↑ | | $AUC_{30}$ ↑ | |
| **Method** | Upright | Rotated | Upright | Rotated | Upright | Rotated | Upright | Rotated | Upright | Rotated | Upright | Rotated |
| VGGT | 0.795 ± 0.030 | 0.788 ± 0.010 | 98.65 ± 0.86 | 97.59 ± 0.82 | 0.659 ± 0.033 | 0.665 ± 0.006 | 85.80 ± 9.53 | 84.44 ± 2.25 | 0.685 ± 0.042 | 0.659 ± 0.032 | 94.18 ± 9.62 | 93.33 ± 1.63 |
| VGGT-Ω | 0.825 ± 0.048 | 0.825 ± 0.008 | 99.10 ± 0.72 | 98.46 ± 0.63 | 0.672 ± 0.040 | 0.679 ± 0.006 | 89.35 ± 5.07 | 87.78 ± 1.58 | 0.669 ± 0.080 | 0.657 ± 0.023 | 93.27 ± 13.12 | 93.52 ± 0.99 |
| $\pi^3$ | 0.817 ± 0.066 | 0.808 ± 0.011 | 96.17 ± 2.97 | 95.42 ± 0.82 | 0.674 ± 0.041 | 0.685 ± 0.008 | 87.54 ± 6.31 | 85.94 ± 1.74 | 0.721 ± 0.059 | 0.667 ± 0.040 | 97.76 ± 0.63 | 97.17 ± 1.30 |
| MapA. | 0.739 ± 0.057 | 0.716 ± 0.017 | 93.11 ± 5.13 | 91.59 ± 1.54 | 0.623 ± 0.046 | 0.636 ± 0.013 | 76.26 ± 12.07 | 73.28 ± 3.17 | 0.627 ± 0.052 | 0.592 ± 0.033 | 94.53 ± 0.98 | 92.57 ± 1.79 |
| DA3 | 0.819 ± 0.037 | 0.808 ± 0.014 | 99.75 ± 0.38 | 99.53 ± 0.25 | 0.676 ± 0.032 | 0.684 ± 0.007 | 88.10 ± 7.73 | 88.55 ± 1.48 | 0.674 ± 0.051 | 0.661 ± 0.030 | 98.58 ± 2.11 | 98.74 ± 1.36 |

performed only for the upright image at each viewpoint, and the poses for the other roll angles are constructed analytically. This enables a robust evaluation of each method under realistic in-plane camera roll, while keeping the underlying scene geometry and viewpoint fixed.

We show example images from our dataset in Figures Figure H and Figure I.

## H Visual Quality Assessment and Rejection Statistics

We render rotated images as described in algorithm 1. To reduce rendering artifacts, we use only the single-step diffusion denoiser (Difix) from Difix3D+ (Wu et al., 2025), applied independently to each rendered view as a post-processing step. We compare renders without and with this denoising step in Figure J and K. We observe that 3DGS reconstructions from dense image sets (e.g., DTU and Mip-NeRF 360) already yield high-quality renders, for which VisualQuality-R1 (Wu et al., 2026) outputs a score of $\geq 4.0$ on a scale of $[0, 5]$. Therefore, we apply the Difix denoiser only to images rendered from the remaining, sparser datasets. We then filter images with artifacts using VisualQuality-R1, passing each rotated image together with its $0^\circ$ counterpart and a prompt asking for a quality score in the range $[0, 5]$. Finally, we manually remove any remaining renders with obvious artifacts. Statistics for the generated and filtered images are given in table K, and example images from the quality filtering pipeline are shown in fig. L.




Figure H: **RotMV dataset roll sweeps: Foosball Table (Indoor).** Sampled views (rows) across in-plane roll angles ($\theta \in [0°, 300°]$, columns).

Table K: Statistics of VLM (VisualQuality-R1) based and manual rejection during dataset curation using controlled rotation synthesis.

| Dataset | Initial images | Rejected by VQ-R1 | Manually rejected | Final |
|---|---|---|---|---|
| Tanks&Temples | 43,248 | 37,248 | 1,080 | 4,920 |
| NRGBD | 26,424 | 15,864 | 1,608 | 8,952 |
| ETH3D | 10,896 | 9550 | 534 | 812 |

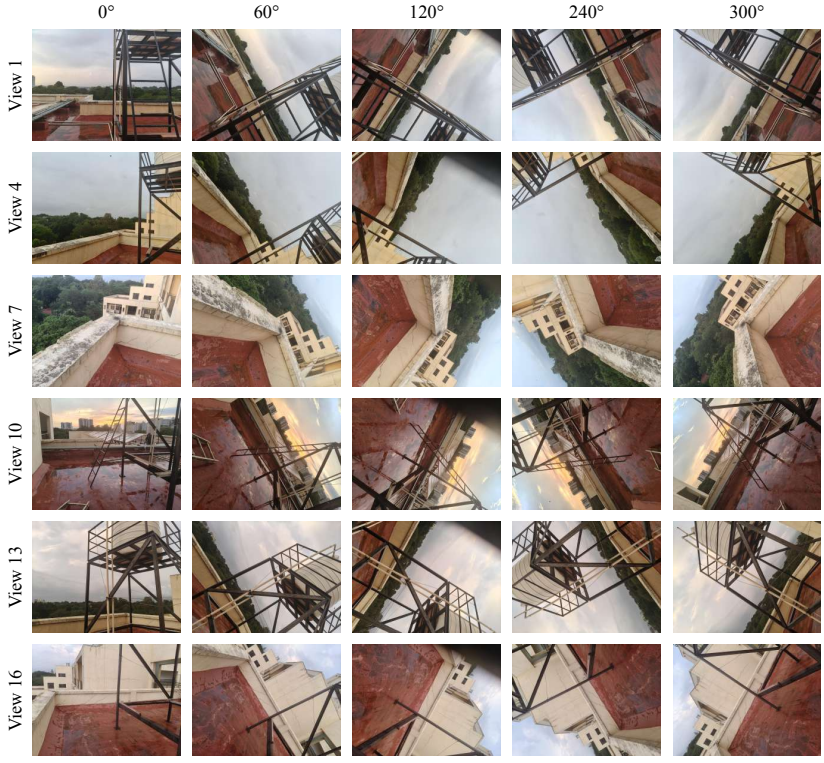
# I FAILURE CASE

Table L reports pose and point-cloud metrics on ETH3D on the scene *office*. Across other scenes, variance-based roll correction recovers the upright prior and improves the reconstruction, but the *office* scene is a consistent failure case: correction degrades *every* metric for *every* model relative to the baseline. ATE increases uniformly (e.g., $0.040$ to $0.095$ for VGGT-$\Omega$), and both Tacc@30 and Racc@30 fall sharply (e.g., $98.39$ to $63.72$ and $100.00$ to $81.57$ for DA3).

We attribute this failure to rotation to symmetry failure and primarily rotation in pitch and yaw.Figure M qualitatively compares the original upright ($0°$) views with the ROVER-corrected views obtained from the corresponding rotated inputs.




(a) Chairs (Indoor)




(b) Rooftop (Outdoor)

Figure I: **RotMV dataset roll sweeps: Chairs and Rooftop.** Sampled views (rows) across in-plane roll angles ($\theta \in [0°, 300°]$, columns).




Figure J: **Effect of Difix denoising on 3DGS-based rotated view synthesis.** Reference images (left), 3DGS-synthesized views (middle), and their denoised counterparts (right), shown for the *Courtyard*, *Electro* and *Pipes* scenes from ETH3D. Red boxes mark artifacts in the synthesized views, with corresponding regions in the denoised views to indicate that the artifacts are removed or reduced.




Figure M: Qualitative failure on the ETH3D *office* scene. Steep upward pitch and ceiling symmetries cause ROVER to introduce spurious corrective rotations to already-upright (0°) views.

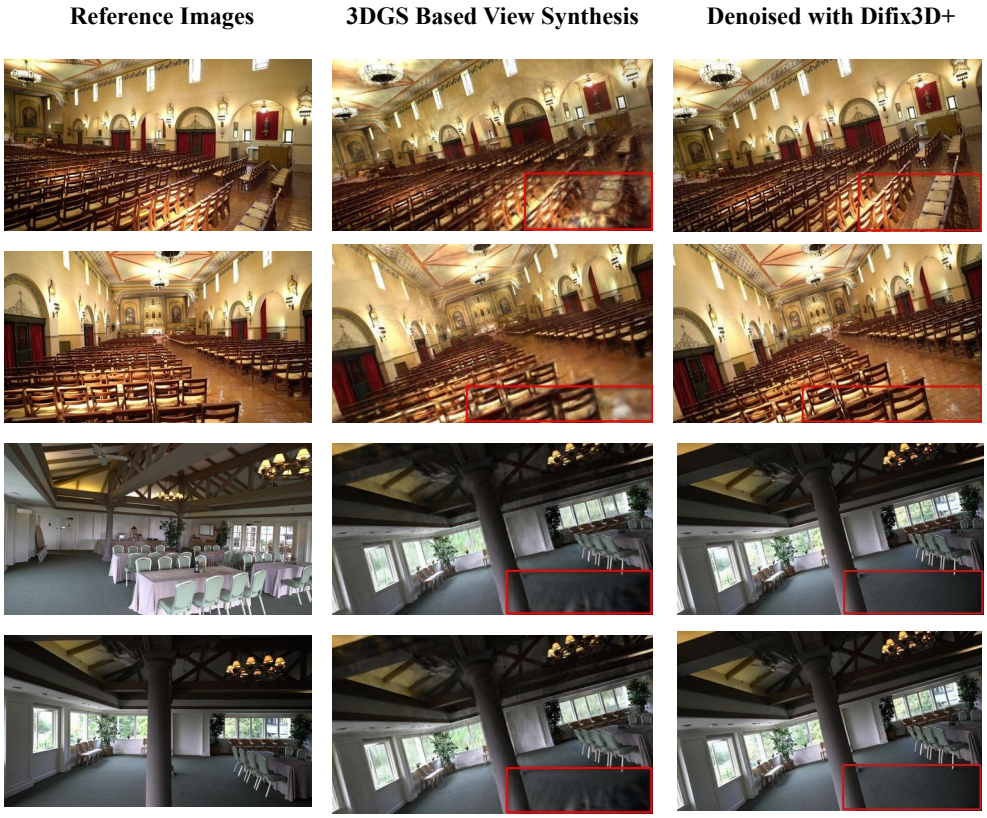



Figure K: **Effect of Difix3D denoising on 3DGS-based rotated view synthesis.** Reference images (left), 3DGS-synthesized views (middle), and their denoised counterparts (right), shown for the *Church* and *Meetingroom* scenes from Tanks and Temples. Red boxes mark artifacts in the synthesized views, with corresponding regions in the denoised views to indicate that the artifacts are removed or reduced.

Table L: Pose estimation results on the ETH3D *office* scene.

| Model | ATE ↓ | | Tacc@30 ↑ | | Racc@30 ↑ | |
|---|---|---|---|---|---|---|
| | Base | Ours | Base | Ours | Base | Ours |
| DA3 | **0.049** | 0.265 | **98.39** | 63.72 | **100.00** | 81.57 |
| MapA | **0.236** | 0.313 | **25.89** | 19.41 | **92.68** | 66.75 |
| $\pi^3$ | **0.208** | 0.326 | **54.63** | 33.56 | **92.68** | 56.40 |
| VGGT | **0.157** | 0.241 | **77.38** | 57.05 | **92.68** | 69.49 |
| VGGT-Ω | **0.040** | 0.095 | **96.18** | 86.54 | **100.00** | 87.50 |

## J Evaluation Criteria

Following the protocol described in Section B to generate multi-view scenes with random rotation across frames. We follow the evaluation protocol from Wang et al. (2025b) and evaluate camera pose and point cloud for each rotation configuration set.

**Ground-truth pose under roll.** Let $R$ and $t$ be the $SO(3)$ ground truth rotation matrix and translation vector respectively for a frame. For a roll of $\theta$ about the optical axis, let $\mathbf{R}_z(\theta)$ be the $SO(3)$

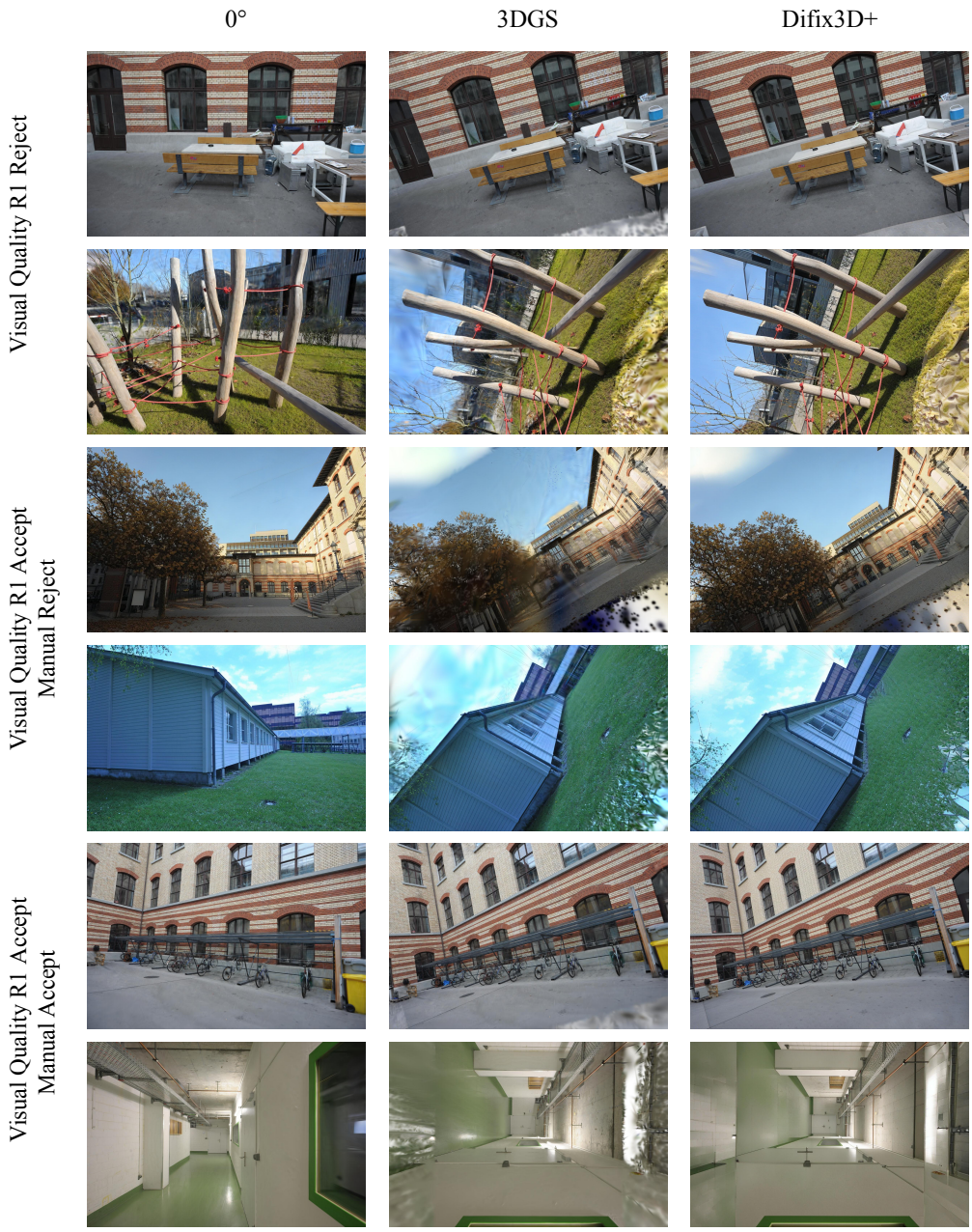



Figure L: Qualitative examples of the quality filtering pipeline across reference (0°), raw 3DGS, and Difix-denoised views, showing automated rejections (top), manual rejections (middle), and accepted views (bottom).

rotation matrix about the $z-$axis. Then, under a roll of $\theta$, the world-to-camera pose $[\mathbf{R} \mid \mathbf{t}]$ becomes $\hat{\mathbf{R}} = \mathbf{R}_z(\theta)\mathbf{R}$ and $\hat{\mathbf{t}} = \mathbf{R}_z(\theta)\mathbf{t}$. The camera centre is unchanged,

$$-\hat{\mathbf{R}}^\top \hat{\mathbf{t}} = -\mathbf{R}^\top \mathbf{R}_z(\theta)^\top \mathbf{R}_z(\theta)\, \mathbf{t} = -\mathbf{R}^\top \mathbf{t}, \tag{1}$$

so the ground-truth trajectory is identical to the upright one for every rotation budget $k$ and angle assignment; the reported ATE degradation reflects the predicted trajectory alone. Relative rotations, by contrast, are genuinely re-oriented, so rotation related metrics are evaluated against altered targets.

**Ground-truth geometry under roll.** Some datasets like ETH3D and DTU provide ground-truth geometry only for the upright (0°) configuration. Since the scene itself is unchanged by the roll,

we adjust the *support* of the ground truth rather than its content: the ground-truth depth map $\mathbf{D}$ is rotated in the image plane, $\hat{\mathbf{D}} = \mathbf{R}_z(\theta)\mathbf{D}$, regions falling outside the frame after rotation are masked out, and the remaining pixels are unprojected to 3D using the rotated camera $[\hat{\mathbf{R}} \mid \hat{\mathbf{t}}]$. This yields the ground-truth point cloud restricted to the region actually visible in the rotated view, so that predictions are never penalised for geometry outside the rolled frustum.

For datasets without ground-truth geometry (Mip-NeRF 360, RotMV) we take the model's own upright reconstruction $\hat{\mathbf{P}}_0$ as reference and report the post-alignment discrepancy of each rotated reconstruction $\hat{\mathbf{P}}_k$ against it. For our hand-captured dataset no ground-truth poses are available; we run VGGT on the upright capture to obtain reference poses and derive the ground truth for each rotated configuration by composing these with the known roll matrices as in equation 1. These two protocols measure degradation relative to the model's own upright output rather than absolute accuracy; our absolute-ground-truth results on 7-Scenes, NRGB-D, ETH3D and DTU establish the corresponding absolute claims.

**Alignment.** Feed-forward predictions are defined only up to a global similarity transform. For pose metrics we align the predicted camera centres to the ground-truth centres with Umeyama. For point cloud metrics we apply Umeyama followed by ICP to align the predicted point cloud to the ground truth, following the protocol of $\pi^3$. All metrics are computed after alignment.

**Pose metrics.** Let $\mathbf{T}_i, \hat{\mathbf{T}}_i \in SE(3)$ denote the ground-truth and predicted extrinsics of view $i$, with camera centres $\mathbf{C}_i, \hat{\mathbf{C}}_i$. After a $\mathrm{Sim}(3)$ trajectory alignment the absolute trajectory error is

$$\mathrm{ATE} = \left( \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \left\| \mathbf{C}_i - \hat{\mathbf{C}}_i \right\|_2^2 \right)^{1/2}. \tag{2}$$

For each of the $\binom{N}{2}$ image pairs $(i, j)$ we form the relative pose $\Delta\mathbf{T}_{ij} = \mathbf{T}_j \mathbf{T}_i^{-1}$ (and $\Delta\hat{\mathbf{T}}_{ij}$ analogously from $\hat{\mathbf{T}}_i, \hat{\mathbf{T}}_j$), decomposed into rotation $\Delta\mathbf{R}_{ij}$ and translation $\mathbf{t}_{ij}$. The relative rotation error is the geodesic distance

$$e_{ij}^R = \arccos\left( \frac{\mathrm{Tr}\big(\Delta\mathbf{R}_{ij}^\top \Delta\hat{\mathbf{R}}_{ij}\big) - 1}{2} \right), \tag{3}$$

and the relative translation error is the undirected angle between translation directions (antipodal directions are equivalent, since scale and sign are unobservable per pair),

$$e_{ij}^T = \arccos\left( \frac{\left| \mathbf{t}_{ij}^\top \hat{\mathbf{t}}_{ij} \right|}{\|\mathbf{t}_{ij}\| \, \|\hat{\mathbf{t}}_{ij}\|} \right). \tag{4}$$

The reported percentages are obtained by thresholding these per-pair errors: $\mathrm{RAcc}_\tau$ and $\mathrm{TAcc}_\tau$ (Relative Rotation/Translation Accuracy) are the fraction of pairs whose error falls below an angular tolerance $\tau$,

$$\mathrm{RAcc}_\tau = \frac{100}{\binom{N}{2}} \sum_{i<j} \mathbb{1}\left[ e_{ij}^R < \tau \right], \qquad \mathrm{TAcc}_\tau = \frac{100}{\binom{N}{2}} \sum_{i<j} \mathbb{1}\left[ e_{ij}^T < \tau \right],$$

reported as percentages; we use $\tau = 30^\circ$ in the tables. The reported AUC uses the *joint* per-pair error $e_{ij}^{RT} = \max\big(e_{ij}^R, e_{ij}^T\big)$, thresholded exactly as $\mathrm{RAcc}_\tau$/$\mathrm{TAcc}_\tau$ above,

$$\mathrm{Acc}_\tau^{\mathrm{joint}} = \frac{100}{\binom{N}{2}} \sum_{i<j} \mathbb{1}\left[ e_{ij}^{RT} < \tau \right]. \tag{5}$$

$\mathrm{AUC@}\tau_{\max}$ is the mean of this joint accuracy over every integer-degree threshold up to $\tau_{\max}$ — a discrete Riemann-sum approximation, at $1^\circ$ resolution, of the area under the accuracy-vs-threshold curve on $[0^\circ, \tau_{\max}]$, normalised so a perfect predictor scores 100:

$$\mathrm{AUC@}\tau_{\max} = \frac{1}{\tau_{\max}} \sum_{\tau=1}^{\tau_{\max}} \mathrm{Acc}_\tau^{\mathrm{joint}}. \tag{6}$$

We report AUC@30 (i.e. $\tau_{\max} = 30^\circ$) in the main tables.

**Point cloud metrics.** Following evaluation protocol of $\pi^3$(Wang et al., 2025b), predicted point maps are aligned to the ground truth with a closed-form $\mathrm{Sim}(3)$ transform (Umeyama), computed directly from the pixel-wise correspondence between predicted and ground-truth point maps, then refined with point-to-point ICP (correspondence threshold $0.1$–$1.0\,\mathrm{m}$, scaled to each dataset's depth range). Let $\mathcal{P}$ denote the resulting aligned predicted point cloud and $\mathcal{Q}$ the ground truth. Accuracy and completeness are the mean one-sided nearest-neighbour distances,

$$\mathrm{Acc} = \tfrac{1}{|\mathcal{P}|} \textstyle\sum_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{q}\in\mathcal{Q}} \|\mathbf{p}-\mathbf{q}\|_2, \tag{7}$$

$$\mathrm{Comp} = \tfrac{1}{|\mathcal{Q}|} \textstyle\sum_{\mathbf{q}\in\mathcal{Q}} \min_{\mathbf{p}\in\mathcal{P}} \|\mathbf{p}-\mathbf{q}\|_2, \tag{8}$$

and their average is the well known Chamfer distance. Normal consistency (N.C.) is likewise two-sided, averaging the mean absolute alignment of surface normals at nearest-neighbour correspondences in each direction,

$$\mathrm{NC}_1 = \tfrac{1}{|\mathcal{P}|} \textstyle\sum_{\mathbf{p}\in\mathcal{P}} \left|\mathbf{n}_{\mathbf{p}} \cdot \mathbf{n}_{\mathrm{nn}_{\mathcal{Q}}(\mathbf{p})}\right|, \tag{9}$$

$$\mathrm{NC}_2 = \tfrac{1}{|\mathcal{Q}|} \textstyle\sum_{\mathbf{q}\in\mathcal{Q}} \left|\mathbf{n}_{\mathbf{q}} \cdot \mathbf{n}_{\mathrm{nn}_{\mathcal{P}}(\mathbf{q})}\right|, \tag{10}$$

where $\mathrm{nn}_{\mathcal{X}}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{v}-\mathbf{x}\|_2$ denotes the nearest neighbour of $\mathbf{v}$ in point set $\mathcal{X}$ (reusing the correspondences already computed for Acc/Comp above), and $\mathbf{n}_{(\cdot)}$ denotes the surface normal estimated at each point from its local neighbourhood. We report

$$\mathrm{NC} = \tfrac{1}{2}(\mathrm{NC}_1 + \mathrm{NC}_2) \in [0,1], \tag{11}$$

with $\mathrm{NC} = 1$ indicating perfectly consistent surface orientation between prediction and ground truth, and $\mathrm{NC} = 0$ indicating orthogonal (maximally inconsistent) surfaces.

## K Training Data re-balancing

Table M presents a comprehensive audit of the training corpora of all evaluated models. Datasets used by these models are CO3D(Reizenstein et al., 2021a;b), ScanNet/ScanNet++(Dai et al., 2017; Yeshwanth et al., 2023), MegaDepth(Li & Snavely, 2018), DL3DV-10K(Ling et al., 2024), ARKitScenes(Baruch et al., 2021), Mapillary(Antequera et al., 2020), HyperSim(Roberts et al., 2021), Habitat/Replica(Ramakrishnan et al., 2021), Virtual KITTI(Gaidon et al., 2016), BlendedMVS(Yao et al., 2020), WildRGB-D(Xia et al., 2024), PointOdyssey(Zheng et al., 2023), Taskonomy(Zamir et al., 2018), Waymo OpenSun et al. (2020a), MegaSynthJiang et al. (2024), MVS-Synth(Huang et al., 2018), UnrealStereo4K(Zhang et al., 2018), MatrixCity(Li et al., 2023), Aria Series(Pan et al., 2023; Banerjee et al., 2024; Ma et al., 2024; Lv et al., 2024), Dynamic Replica(Karaev et al., 2023), Spring(Mehl et al., 2023), Structured3D(Zheng et al., 2020), CityscapesCordts et al. (2016), GTA-SfM(Cao et al., 2020), Kubric(Greff et al., 2022), TartanAir(Wang et al., 2020b) and MidAir (Fonder & Droogenbroeck, 2019) datasets. Across **27** datasets spanning indoor RGB-D scans, outdoor photo collections, driving sequences, and synthetic renderings, the overwhelming majority are upright by construction with each model trained on between 10 and 19 fully upright datasets and at most 2 partially aligned ones. This confirms that UprightBias is a data-driven phenomenon: feed-forward models inadvertently absorb a strong canonical orientation prior simply by virtue of what their training data looks like. To investigate if re-balancing the training data and finetuning the model recovers the performance we fine-tune $\pi^3$ on a re-balanced version of the DTU training set augmented with randomly rotated multi-view sequences, and evaluate on the DTU test set under the same rotated-input protocol used throughout the paper. Tables Table N and Table O report point-map and camera pose metrics respectively for the pretrained and fine-tuned models across four test scenes. The results do not support the re-balancing hypothesis. Fine-tuning yields inconsistent changes, consistent with the feature-distortion risk discussed in Section 3.2(Kumar et al., 2022; Tang et al., 2020; Kang et al., 2019). More specifically fine-tuning on rotated data disrupts the DINOv2-derived visual representations that $\pi^3$ relies on, as the rotated fine-tuning distribution diverges from the DINOv2 pretraining distribution.

Table M: Training datasets used by each feed-forward geometric model and their upright alignment status. ✓ = fully upright; ∼ = partially upright The overwhelming majority of training data across all models is upright by construction

| **Dataset** | **Capture Modality** | **Upright** | **VGGT** | **VGGT-Ω** | **DA3** | **MapA.** | $\pi^3$ |
|---|---|---|---|---|---|---|---|
| CO3D | Handheld object-centric video | ✓ | ✓ | ✓ | ✓ | | ✓ |
| ScanNet / ScanNet++ | Handheld RGB-D indoor scan | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| MegaDepth | Internet landmark photography | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DL3DV-10K | Scene-level video | ✓ | ✓ | ✓ | ✓ | ✓ | |
| ARKitScenes | iPad indoor capture | ✓ | | ✓ | ✓ | | ✓ |
| Mapillary | Street-level vehicle camera | ✓ | ✓ | ✓ | | ✓ | |
| HyperSim | Synthetic indoor rendering | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Habitat / Replica | Simulated indoor environments | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Virtual KITTI | Driving simulation | ✓ | ✓ | ✓ | ✓ | | |
| BlendedMVS | SfM outdoor/indoor photos | ✓ | ✓ | | ✓ | ✓ | ✓ |
| WildRGB-D | In-the-wild RGBD video | ✓ | ✓ | ✓ | ✓ | | ✓ |
| PointOdyssey | Synthetic ground-level motion | ✓ | ✓ | ✓ | ✓ | | |
| Taskonomy | Indoor scene photography | ✓ | | ✓ | ✓ | | ✓ |
| Waymo Open | Autonomous driving cameras | ✓ | | ✓ | ✓ | | |
| MegaSynth | Synthetic scene rendering | ✓ | | ✓ | ✓ | | |
| MVS-Synth | Synthetic multi-view stereo | ✓ | ✓ | ✓ | ✓ | ✓ | |
| UnrealStereo4K | Synthetic indoor/outdoor | ✓ | | ✓ | ✓ | ✓ | |
| MatrixCity | Synthetic city aerial/ground | ✓ | | | ✓ | | ✓ |
| Aria Series | Egocentric wearable capture | ✓ | ✓ | ✓ | | ✓ | |
| Dynamic Replica | Synthetic dynamic indoor | ✓ | | ✓ | | ✓ | |
| Spring | Synthetic outdoor driving | ✓ | | | | ✓ | |
| Structured3D | Synthetic indoor rendering | ✓ | | | ✓ | | |
| Cityscapes | Driving forward-facing camera | ✓ | | | ✓ | | |
| GTA-SfM | Game engine driving scenes | ✓ | | | | | ✓ |
| Kubric | Synthetic random viewpoints | ∼ | ✓ | | | | |
| TartanAir | Drone and aerial simulation | ∼ | | ✓ | ✓ | | ✓ |
| Mid-Air | Drone aerial footage | ∼ | | ✓ | | | ✓ |
| **Fully upright datasets used** | | | **13** | **18** | **19** | **10** | **11** |
| **Partially upright datasets used** | | | **1** | **2** | **1** | **0** | **2** |

Table N: DTU test-scene point-cloud metrics for the $\pi^3$ backbone finetuning with rebalanced dataset. Results are reported as mean ± standard deviation over 50 rotation sets per scene. We compare the pretrained model with the finetuned model.

| Scene | Accuracy ↓ | | Completeness ↓ | | Normal Consistency ↑ | |
|---|---|---|---|---|---|---|
| | Pretrained | Finetuned | Pretrained | Finetuned | Pretrained | Finetuned |
| scan1 | **0.923** ±0.311 | 2.251 ±0.279 | **0.902** ±0.281 | 1.347 ±0.161 | 0.646 ±0.009 | **0.652** ±0.006 |
| scan10 | 5.701 ±6.254 | **4.340** ±1.000 | 4.463 ±9.323 | **4.057** ±0.704 | 0.655 ±0.020 | **0.681** ±0.013 |
| scan118 | 3.384 ±2.605 | **1.467** ±0.272 | 5.969 ±9.243 | **1.493** ±0.286 | 0.593 ±0.039 | **0.608** ±0.004 |
| scan77 | **5.962** ±1.011 | 7.931 ±1.186 | **6.658** ±1.894 | 12.447 ±2.721 | **0.622** ±0.010 | 0.607 ±0.007 |

## L Outlier Detection using RobustVGGT

A natural alternative to correcting rotated views is to detect and discard them. As noted in section 4.5, we evaluate this strategy using RobustVGGT (Han et al., 2025), which flags views that are inconsistent with the rest of the sequence and excludes them before reconstruction. Unlike

Table O: DTU test-scene pose metrics for the $\pi^3$ backbone finetuning with rebalanced dataset. Results are reported as mean $\pm$ standard deviation over 50 rotation sets per scene. We compare the pretrained model with the finetuned (v2) model.

| Scene | ATE ↓ | | $AUC_{30}$ ↑ | |
|---|---|---|---|---|
| | Pretrained | Finetuned | Pretrained | Finetuned |
| scan1 | **9.161** **±3.658** | 10.550 ±3.293 | **94.832** **±2.343** | 94.073 ±1.927 |
| scan10 | 72.433 ±77.302 | **34.919** **±4.356** | **77.543** **±18.736** | 61.702 ±3.774 |
| scan118 | 79.647 ±88.556 | **17.696** **±3.658** | 81.022 ±17.060 | **86.688** **±3.347** |
| scan77 | **49.597** **±41.676** | 206.479 ±64.354 | **77.976** **±12.527** | 26.733 ±9.703 |

ROVER, which estimates the orientation of each view independently, this approach relies on relations between views. Following RobustVGGT , we identify rotated frames in a zero-shot manner directly from the model's internal feature representations. Specifically, we extract spatial patch tokens $\mathbf{X}_i \in \mathbb{R}^{N_p \times D_{\text{feat}}}$ and multi-head query-key projections $\mathbf{Q}_i, \mathbf{K}_i \in \mathbb{R}^{H \times N_p \times d_k}$ from the first global self-attention layer across an $N$-frame sequence during a single forward pass.

### L.1 Relational Representation Matrix

For each frame pair $(i, j)$, we evaluate inter-frame consistency through token similarity and global attention:

$$S_{i,j} = \frac{1}{N_p^2} \sum_{u,v=1}^{N_p} \frac{\langle \mathbf{X}_{i,u}, \mathbf{X}_{j,v} \rangle}{\|\mathbf{X}_{i,u}\|_2 \|\mathbf{X}_{j,v}\|_2}, \qquad A_{i,j} = \frac{1}{HN_p} \sum_{h=1}^{H} \sum_{u=1}^{N_p} \text{Softmax}\left( \frac{\mathbf{Q}_{i,u}^{(h)} (\mathbf{K}_j^{(h)})^\top}{\sqrt{d_k}} \right). \quad (12)$$

Both matrices are min-max normalized to $[0, 1]$ across all frame pairs using a small constant $\epsilon = 10^{-6}$ for numerical stability, yielding normalized matrices $\mathbf{S}'$ and $\mathbf{A}'$. We construct the joint symmetric affinity matrix as $\mathbf{W} = \frac{1}{2}(\mathbf{S}' + \mathbf{A}') \in \mathbb{R}^{N \times N}$.

### L.2 Sequence-Level Rotation Detection

To detect orientation anomalies, we define a sequence reference profile $\mathbf{r} \in \mathbb{R}^N$ by taking the column-wise median across frames, $r_j = \text{median}_i(W_{i,j})$, and compute each frame's mean absolute deviation:

$$d_i = \frac{1}{N} \sum_{j=1}^{N} |W_{i,j} - r_j|. \quad (13)$$

Frames are sorted in descending order of deviation: $\mathbf{d}_{\text{sorted}} = [d_{\rho(1)}, \ldots, d_{\rho(N)}]$. We partition the sequence under two orientation priors:

• **Minority Rotation (Gap-Based):** Assuming rotated frames form an anomalous minority ($s \leq \lfloor N/2 \rfloor$), the decision boundary is placed at the largest consecutive deviation drop:

$$s^* = \arg \max_{1 \leq s \leq \lfloor N/2 \rfloor} (d_{\rho(s)} - d_{\rho(s+1)}). \quad (14)$$

The top $s^*$ frames $\{\rho(1), \ldots, \rho(s^*)\}$ are classified as **rotated**, while the remaining $N - s^*$ frames are labeled as **upright**.

• **Majority Rotation (Variance Minimization):** When rotated frames dominate, the reference profile $\mathbf{r}$ reflects the rotated orientation, making upright frames appear as high-deviation outliers. We partition the sequence into candidate upright frames $G_{\text{up}} = \{d_{\rho(1)}, \ldots, d_{\rho(s)}\}$ and rotated frames $G_{\text{rot}} = \{d_{\rho(s+1)}, \ldots, d_{\rho(N)}\}$. Sweeping $s \in [1, \min(8, N-1)]$, the optimal split $s^*$ minimizes the within-class variance:

$$\sigma_w^2(s) = \frac{s}{N} \sigma^2(G_{\text{up}}) + \frac{N-s}{N} \sigma^2(G_{\text{rot}}). \quad (15)$$

The top $s^*$ frames are isolated as **upright**, and the remaining $N - s^*$ frames are classified as **rotated**.

### L.3 SENSITIVITY TO THE ROTATION PRIOR

As shown in Table Table P, detection is substantially more reliable when rotated frames form the minority (average Global F1 of $0.811$ vs. $0.543$ under the majority setting). Precision and recall both degrade when canonical reference frames become scarce, demonstrating the necessity of an upright reference anchor for unsupervised image identification

### L.4 LIMITATIONS OF ROBUSTVGGT

The approach described in RobustVGGT drops the anomalous frames and predicts a sparse point cloud. Further, it would predict cameras only for the chosen frames. The results in Table P highlight two limitations of rejection based approach in RobustVGGT for handling rotated frames. First, detection depends on upright frames serving as a reference, and becomes unreliable when such frames are scarce. Second, even perfect detection discards the rotated views, reducing coverage of the scene. ROVER avoids both: it estimates the orientation of each view from its own DINOv2 features, independent of the fraction of rotated views, and retains all views after canonicalization. This allows our approach to recover reconstruction quality.

Table P: Rotation detection performance under minority and majority rotated settings.

| | | **Minority Rotated** | | | **Majority Rotated** | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Model** | **Prec. ↑** | **Rec. ↑** | **F1 ↑** | **Prec. ↑** | **Rec. ↑** | **F1 ↑** |
| DTU | VGGT | 0.9180 | 0.8127 | 0.8621 | 0.3676 | 0.3452 | 0.3560 |
| | VGGT-Ω | 0.9144 | 0.8012 | 0.8541 | 0.5239 | 0.4179 | 0.4649 |
| | $\pi^3$ | 0.9067 | 0.8024 | 0.8514 | 0.4168 | 0.3726 | 0.3935 |
| ETH3D | VGGT | 0.7673 | 0.6183 | 0.6848 | 0.3699 | 0.2847 | 0.3217 |
| | VGGT-Ω | 0.7604 | 0.6135 | 0.6791 | 0.3608 | 0.3017 | 0.3286 |
| | $\pi^3$ | 0.7545 | 0.6413 | 0.6933 | 0.4014 | 0.6423 | 0.4941 |
| Mip-NeRF 360 | VGGT | 0.9442 | 0.8844 | 0.9133 | 0.8756 | 0.8368 | 0.8558 |
| | VGGT-Ω | 0.9075 | 0.8394 | 0.8721 | 0.8895 | 0.7655 | 0.8228 |
| | $\pi^3$ | 0.9121 | 0.8594 | 0.8849 | 0.8906 | 0.8130 | 0.8500 |
| **Average** | | **0.865** | **0.764** | **0.811** | **0.566** | **0.531** | **0.543** |