Flow3r: Factored Flow Prediction for Visual Geometry Learning

Zhongxiao Cong Qitao Zhao Minsik Jeon Shubham Tulsiani
Carnegie Mellon University
{zcong, qitaoz, minsikj, stulsian}@cmu.edu

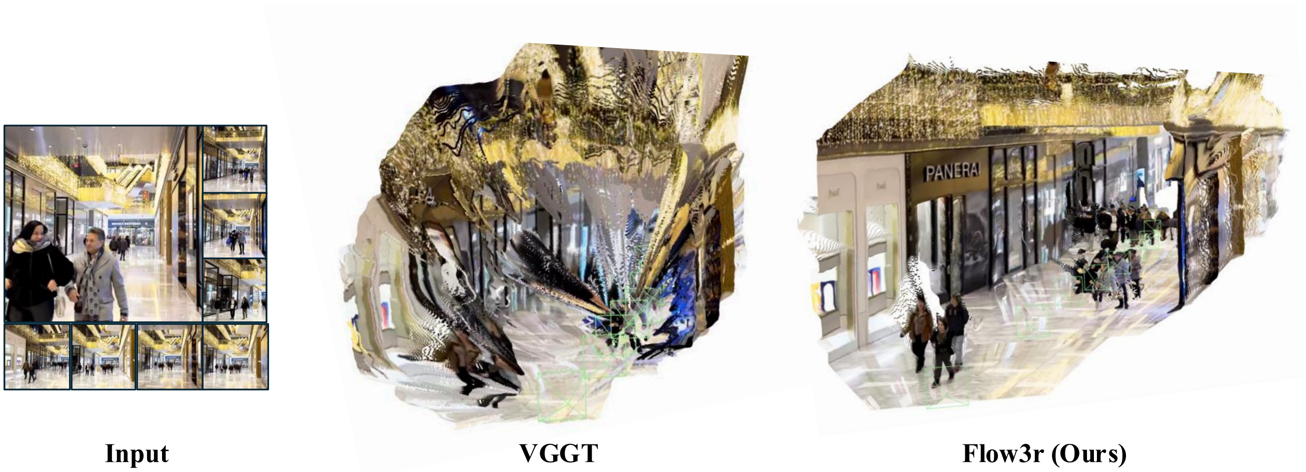

Figure 1. **Flow3r** leverages *unlabeled videos* via flow supervision for visual geometry learning, enabling more accurate reconstruction in challenging real-world dynamic scenes compared to supervised baselines that can only leverage dense geometry and pose supervision.

Abstract

We propose Flow3r, a scalable framework for visual geometry learning that leverages flow prediction to guide learning using unlabeled monocular videos. Current 3D/4D reconstruction systems primarily rely on dense geometry and pose supervision, and cannot easily generalize to diverse dynamic real-world scenes. In this work, we propose a mechanism to augment training directly from unlabeled videos, leveraging dense 2D correspondences (or 'flow') between arbitrary image pairs as supervision. Our key insight is that a factored flow prediction module that computes between two images using 'geometry latents' from one image and the 'pose latent' from other can guide visual geometry learning. We first highlight the benefits and scalability of flow supervision in controlled settings and then leverage large-scale unlabeled data to improve off-the-shelf visual geometry models. We evaluate Flow3r across diverse 3D benchmarks and demonstrate competitive or state-of-the-art performance, even surpassing supervised models trained with more labeled data.

1. Introduction

The task of 'visual geometry inference' *i.e.* recovering the 3D structure of a (static or dynamic) scene from multi-view input images has undergone a paradigm shift – evolving from classical optimization-based methods [1, 13, 21, 27, 30, 36] to recent data-driven predictors [31, 33, 34, 41] that can directly output the geometry and pose corresponding to the input images. The success of such efficient feedforward systems, however, has crucially relied on multi-view training data with dense geometry and camera pose supervision. Unfortunately, this supervision may not be easily available across all settings of interest *e.g.* for dynamic scenes in-the-wild or domains like egocentric videos, and existing visual geometry prediction methods do not generalize well to such scenarios. More broadly, unlike self-supervised learning objectives common for training LLMs and vision transformers, the reliance on dense geometry and pose labels prevents truly large-scale visual geometry learning.

In this work, we take a step towards scalable learning of multi-view models and present Flow3r, a framework to guide visual geometry learning from *unlabeled* videos *i.e.* without any explicit geometry or pose supervision. Instead, Flow3r leverages a readily available supervisory signal that is a cornerstone of classical (and recent) optimization-based mutli-view methods – (dense) *correspondences across images*. In particular, we are inspired by the progress in inferring dense correspondences or pixel tracks for generic image pairs and videos, and seek to unlock scalable learning by incorporating such (2D) flow as auxiliary supervision for (3D) visual geometry models. The key technical question we seek to answer is: ***'how can flow be effectively leveraged for supervising visual geometry prediction?'***.

We are not the first to consider flow supervision for guiding visual geometry learning. Indeed, the seminal VGGT work [31] adds a 'tracking' module that uses local features from two images to predict a flow between them, and uses this as an auxiliary training objective. However, as we show later, this merely encourages the corresponding features to be visually discriminative but does not directly aid the learning of pose or geometry. Our key insight is that to guide geometry learning, the design of the ***flow prediction module should be asymmetric***. We build on the observation that for static scenes, the flow between a source and a target image can be induced only via the geometry of the source image (pointmaps in a global coordinate) and the camera pose of the target. Building on this, we propose to incorporate a ***factored flow prediction module*** in visual geometry models. Specifically, such models typically compute 'local' patchwise features that later predict geometry as well as a global per-image token that infers camera pose. Our flow prediction module is designed to compute flow between two images using ***only the global pose token for the target along with the patchwise tokens for the source***.

We find that our factored flow prediction helps better supervise pose and geometry learning compared to the symmetric design adopted by previous works, while also allowing robust prediction and applications to dynamic scenes unlike a projection-based flow inference. Flow3r integrates such factored flow prediction and leverages $800k$ unlabeled videos as supervision in addition to existing (labeled) 3D datasets. We show that this allows Flow3r to outperform prior visual-geometry systems, in particular improving over them in in-the-wild dynamic videos where labeled data is scarce. More broadly, we believe that by allowing extracting supervisory signal from unlabeled videos (although leveraging off-the-shelf 2D flow prediction), Flow3r represents a step towards large-scale visual geometry learning without large-scale supervision.

2. Related Work

Dense correspondence Learning. Dense correspondence learning focuses on estimating pixel-level matches across views, which serves as the basis for recovering camera motion and 3D structure. Early work primarily focused on local optical flow [15, 22], providing per-pixel motion between adjacent frames but failing under large viewpoint changes. Building upon these ideas, recent methods learn geometry-aware dense correspondences that remain stable across wide baselines. For example, DKM [11] refines coarse predictions through hierarchical warping to capture fine-grained geometric alignment, while RoMa [12] leverages pretrained visual features to achieve semantically consistent matching across diverse scenes. Transformer-based architectures such as UFM [39] further unify dense matching and optical-flow reasoning, yielding high-quality correspondences that generalize across domains. Beyond pairwise alignment, dense correspondence learning has expanded to long-range video tracking [9, 10, 14], with recent models such as CoTracker [16] jointly reasoning over multiple frames to achieve state-of-the-art occlusion-robust tracking. Our approach builds on dense correspondence learning but goes beyond matching: we introduce a factored flow formulation that explicitly links source geometry and target camera pose. This factorization allows the model to infer geometry-aware flow that is consistent with 3D structure, enabling accurate reconstruction and motion estimation in challenging in-the-wild dynamic scenes.

Correspondence-driven Reconstruction. Building on (dense) correspondence estimation, these methods recover 3D structure and camera motion directly from learned matches. Classical Structure-from-Motion [27] pipelines estimate camera poses and scene structure from sets of images by detecting local features, computing pairwise correspondences, and jointly optimizing camera parameters and 3D points through global bundle adjustment. Extending correspondence-based reconstruction to dynamic scenes, visual SLAM systems track features across frames to jointly estimate camera trajectories and scene geometry. Recent approaches, including Robust-CVD [17], CasualSAM [40] and MegaSAM [20], further incorporate monocular depth priors or single-view geometric supervision to enhance robustness under motion and occlusion. However, these methods remain optimization-based, requiring per-video refinement and lacking feed-forward efficiency.

Feed-forward Visual Geometry Learning Recent efforts aim to replace traditional optimization pipelines with feed-forward networks that directly predict visual geometry from images. DUSt3R [34] first demonstrated that dense pointmaps can be estimated from image pairs within a shared coordinate system, enabling efficient two-view reconstruction. MASt3R [18] further improves this paradigm by introducing a learned matching head for better correspondence reasoning, while DiffusionSfM [41] and VGGT [31] generalize to multi-view settings, jointly estimating camera parameters and scene structure. Subsequent works such as MonST3R [38], CUT3R [33], and StreamVGGT [43] extend this formulation to dynamic

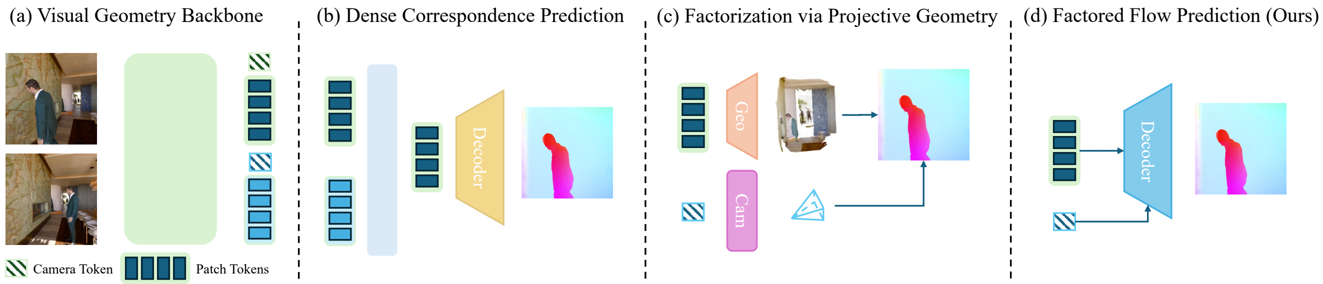
(a) Visual Geometry Backbone
(b) Dense Correspondence Prediction
(c) Factorization via Projective Geometry
(d) Factored Flow Prediction (Ours)
Decoder
Geo
Cam
Decoder
Camera Token
Patch Tokens

Figure 2. **Mechanisms for flow prediction.** (a) The visual geometry backbone first extracts the camera token and patch token of each input. (b) Existing dense correspondence heads [31] predict flow directly from patch features using appearance cues. (c) Flow can also be obtained by explicitly projecting predicted 3D points into another view via decoded camera parameters. However, this projection-based formulation is restricted to static scenes and sensitive to geometric errors. (d) Our factored flow approach conditions source-view geometry latents on target-view camera latents and decodes dense correspondences directly in latent space, providing a geometry-aware and robust flow prediction mechanism that naturally extends to dynamic scenes.

scenes, learning temporally consistent geometry across video frames. However, these models rely on labeled 3D or camera data, which are not easily scalable. In contrast, our method enables scalable feed-forward learning of dynamic visual geometry using factored flow prediction, allowing training on unlabeled real-world videos.

3. Approach

Our approach, Flow3r, aims to scale up visual geometry learning through supervising factored flow prediction, which enables the use of in-the-wild data without requiring ground-truth geometry annotations (*e.g.*, camera poses and depth maps). In this section, we first review the commonly adopted paradigm for visual geometry learning under full supervision of camera poses, depths and pointmaps (Sec. 3.1). We then introduce our factored flow prediction formulation in Sec. 3.2, which decodes flow by combining the camera token from one view with the patch tokens from the other view in an image pair, allowing flow supervision to guide visual geometry learning. We further show that this formulation naturally extends to dynamic scenes. Finally, we describe the overall supervision signals and network architecture in Sec. 3.3.

3.1. Preliminaries: Visual Geometry Networks

State-of-the-art visual geometry networks (*e.g.*, VGGT [31] and π^3 [35]) take as input a set of images $\{I_1, I_2, \dots, I_N\}$ and infer 3D scene geometry through a unified multi-view transformer. For instance, VGGT first encodes the input images into latent patch tokens using an off-the-shelf vision backbone [23]:

$$\mathbf{X}_i = f_{\text{enc}}(I_i), \quad \mathbf{X}_i \in \mathbb{R}^{P \times D}, \tag{1}$$

where P and D denote the number of patch tokens and their feature dimension, respectively.

It then initializes a set of learnable camera tokens $\{\mathbf{c}_i \in \mathbb{R}^D\}$ and appends them to the patch features from each image, forming joint per-view tokens:

$$\mathbf{T}_i = [\,\mathbf{c}_i\,;\,\mathbf{X}_i\,], \quad \mathbf{T}_i \in \mathbb{R}^{(1+P)\times D}. \tag{2}$$

A multi-view transformer then performs cross-view reasoning over all tokens to enable joint understanding of scene structure and camera motion:

$$\{\mathbf{c}'_i, \mathbf{X}'_i\}_{i=1}^N = f_{\text{multi-view}}\big(\{\mathbf{T}_i\}_{i=1}^N\big), \tag{3}$$

where $\mathbf{c}'_i$ and $\mathbf{X}'_i$ denote the updated camera and patch tokens after cross-view aggregation, illustrated in Fig. 2 (a).

Finally, the updated latent representations are decoded into explicit geometric properties – camera poses, depths, and pointmaps.

$$[\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i, \hat{\mathbf{K}}_i] = f_{\text{cam}}(\mathbf{c}'_i), \tag{4}$$

$$\hat{\mathbf{D}}_i, \hat{\mathbf{P}}_i = f_{\text{depth}}(\mathbf{X}'_i), f_{\text{point}}(\mathbf{X}'_i). \tag{5}$$

These outputs are supervised using ground-truth labels, typically obtained from running Structure-from-Motion systems [1, 27]. Modern visual geometry networks are commonly trained on large-scale mixtures of datasets, which require extensive annotation and curation efforts. Yet, further scaling such models remains non-trivial.

3.2. Learning Visual Geometry via Factored Flow

Flow3r leverages *flow* (dense pixel correspondences) as an additional supervision signal for visual geometry learning, enabling the use of in-the-wild data without ground-truth camera or depth annotations and thereby allowing more scalable training. The key technical question we answer is about *how* one can effectively leverage such supervision, and we first detail different possible alternatives before describing our proposed mechanism:

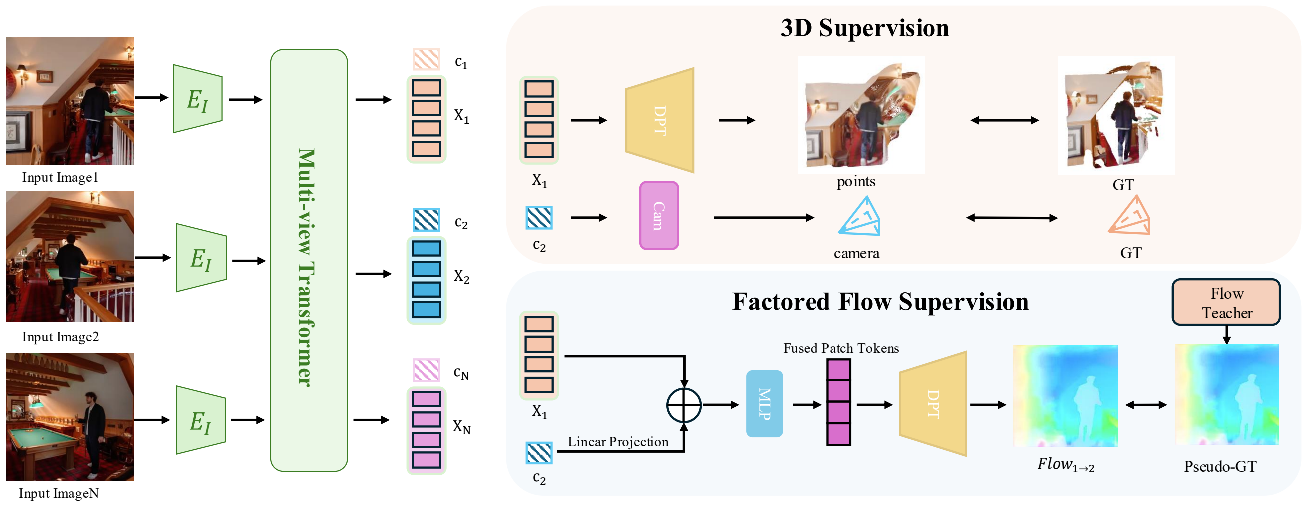

Figure 3. **Overview of Flow3r.** Flow3r predicts visual geometry using factored flow supervision, enabling scalable geometry learning from unlabeled videos. Each input image is encoded and processed by the multi-view transformer to produce camera tokens and patch tokens. For data with dense geometry and pose labels, we directly supervise the patch tokens and camera tokens with the corresponding labels. For unlabeled datasets without geometry and supervision, we predict flow between two frames in a factorized manner, supervised by an off-the-shelf 2D flow prediction model [39]. To obtain the factored flow, we fuse the patch features of one frame with the camera features of the other, and decode the fused representation through the DPT head to produce dense flow predictions.

Flow as Visual Correspondence. One possible approach is to infer flow using local features for each input image (Fig. 2 (b)). Indeed, prior multi-view models [31] adopt this design and leverage flow as an auxiliary training signal. However, we find that this does not necessarily facilitate the *joint learning of scene geometry and camera motion*. This is because only patch tokens (from Eq. 3) are involved, and the predicted flow need not be informed by camera motion or geometry, but merely by visually discriminative features.

Flow from Explicit Camera and Scene Geometry. We note that classical projective geometry provides an alternative mechanism to infer flow between images. Considering a static scene with predicted pointmap $\hat{\mathbf{P}}_i$ from view i and the camera parameters for view j, the flow between these images can be analytically computed by projecting $\hat{\mathbf{P}}_i(\mathbf{u}_i)$ into camera j:

$$\hat{\mathbf{F}}_{i \mapsto j}(\mathbf{u}_i) = \hat{\mathbf{u}}_{i \mapsto j} = \pi\big(\hat{\mathbf{K}}_j(\hat{\mathbf{R}}_j\,\hat{\mathbf{P}}_i(\mathbf{u}_i) + \hat{\mathbf{t}}_j)\big), \tag{6}$$

This explicit flow computation mechanism, grounded in projective geometry, provides an ideal factorization that jointly promotes learning of both scene geometry and camera motion, as illustrated in Fig. 2 (c). However, it has two key limitations: (1) the pure projection in Eq. 6 cannot account for scene motion, restricting the model to learning from only static scenes; and (2) supervising flow from decoded cameras and depths can introduce instability, as errors in either component may lead to large inaccuracies in the resulting flow.

Factored Flow Prediction. Our approach builds on an insight from the projection-based flow computation – that asymmetric flow computation using geometry from source and pose from target can better guide visual geometry learning. However, instead of the analytical computation, we propose to leverage a learned *factored flow prediction* mechanism to infer flow from pose and geometry latent representations (see (c) in Fig. 2).

More specifically, given the geometry latents $\mathbf{X}'_i$ from view i and camera latents $\mathbf{c}'_j$ from view j (the outputs of multi-view transformer in Eq. 3), Flow3r leverages a flow prediction head to predict the flow $\hat{\mathbf{F}}_{i \mapsto j}$:

$$\hat{\mathbf{F}}_{i \mapsto j} = \Phi_{\text{flow}}(\mathbf{X}'_i, \mathbf{c}'_j), \tag{7}$$

where Φ_{flow} denotes a learned flow prediction module. Specifically, it uses the camera latents from view j to modulate the geometry latents from view i, which are subsequently decoded by a DPT head to produce a flow field.

This approach bypasses the need to decode explicit geometric elements for flow computation and therefore improves robustness and enables end-to-end training. Moreover, it also implicitly allows handling dynamic scenes where the flow field no longer corresponds to a pure geometric projection but instead reflects a combination of camera motion and scene motion.

3.3. Overall Architecture and Learning Objectives

Flow3r builds upon a standard supervised visual geometry learning framework with a permutation-equivariant design. Our model jointly optimizes the camera, depth, and point heads to predict their corresponding geometric quantities. Beyond full supervision using dense geometric labels, Flow3r further incorporates factored flow prediction on both labeled and unlabeled data. For datasets without ground-truth geometry, we rely on pseudo-ground-truth correspondences from a strong online teacher, UFM [39]. Taken together, this unified formulation enables scalable visual geometry learning across labeled and unlabeled data, and across both static and dynamic scenes. An overview of our method is provided in Fig. 3.

Permutation-equivariant Formulation. To disentangle visual geometry learning from predefined coordinate frames and improve robustness, we adopt the permutation-equivariant design of π^3 [35], which predicts camera parameters and local geometry without relying on a fixed reference frame. We further extend this formulation to directly predict *global* pointmaps and incorporate flow supervision.

Flow Supervision from Labeled and Unlabeled Data. We supervise the proposed flow head with a robust regression loss, following RoMa [12]. Given the predicted projected coordinates $\hat{\mathbf{u}}_{i \to j}$ and the ground-truth correspondences $\mathbf{u}^{gt}_{i \to j}$, we first compute the displacements between the predicted projection and the source pixel location:

$$\boldsymbol{\phi}_{i \to j} = \hat{\mathbf{u}}_{i \to j} - \mathbf{u}_i, \quad \boldsymbol{\phi}^{gt}_{i \to j} = \mathbf{u}^{gt}_{i \to j} - \mathbf{u}_i. \tag{8}$$

Then the flow loss is defined as:

$$\mathcal{L}_{\text{flow}} = \frac{1}{\sum_{i \in I} C[i]^{gt}} \sum_{i \in I} C[i]^{gt} \, \ell_{\text{robust}} \left(\| \boldsymbol{\phi}_{i \to j}[i] - \boldsymbol{\phi}^{gt}_{i \to j}[i] \|_2 \right),$$

where $C[i]^{gt}$ denotes the ground-truth covisibility mask, and ℓ_{robust} is a generalized Charbonnier loss that emphasizes inlier correspondences.

Camera and Geometry Supervision from Labeled Data. Flow3r predicts per-image cameras, pixel-aligned depth maps, global pointmaps and confidence maps that indicate model uncertainty. For camera supervision, we adopt the same relative rotation loss $\mathcal{L}_{\text{rot}}$ as π^3 [35], but replace their relative translation loss with a loss $\mathcal{L}_{\text{center}}$ directly on the predicted camera centers after computing optimal alignment, which we empirically find to perform better (see appendix for details). The depth head is supervised using the confidence-weighted regression loss introduced in DUSt3R [34], which we denote as $\mathcal{L}_{\text{depth}}$.

We introduce a permutation-equivariant training objective for supervising the global pointmaps where we estimate the best rigid transform that aligns the predicted pointmap $\hat{\mathbf{P}}$ to the ground truth $\mathbf{P}^{gt}$ and computes an ℓ_2 loss on the aligned points:

$$\mathcal{L}_{\text{point}} = \frac{1}{K} \min_{\mathbf{R} \in SO(3),\, \mathbf{T} \in \mathbb{R}^3} \sum_k \| \mathbf{R} \hat{\mathbf{P}}_k + \mathbf{T} - \mathbf{P}^{gt}_k \|_2.$$

Model Variant	Data (# Seqs)	RRA@30↑	RTA@30↑	CD↓	MSE ↓
3d-sup	ScanNet++ (1k)	0.7500	0.6929	0.030	0.088
flow-projective	+ ScanNet++ (1k)	0.6700	0.4572	0.033	0.088
flow-tracking	+ ScanNet++ (1k)	0.7438	0.7021	0.030	0.089
flow-factored	+ ScanNet++ (1k)	0.7700	**0.7366**	**0.026**	**0.078**
3d-sup++	+ ScanNet++ (1k)	**0.8033**	0.7333	**0.026**	0.079

Table 1. **Does factored flow prediction help visual geometry learning on static scenes?** On ScanNet++ [37], our factored flow prediction model (flow-factored) significantly outperforms the no-flow baseline (3d-sup), while outperforming other alternatives that leverage flow supervision. It even performs highly comparable with the fully-3D-supervised baseline (3d-sup++).

Model Variant	Data (# Seqs)	RRA@30↑	RTA@30↑	CD ↓	MSE ↓
3d-sup	OmniWorld (1k)	66.01	62.37	0.105	0.637
flow-projective	+ SpatialVID (3k)	61.23	56.12	0.158	0.710
flow-tracking	+ SpatialVID (3k)	68.56	62.95	0.107	0.628
flow-factored	+ SpatialVID (3k)	76.26	68.84	0.103	0.598
flow-factored+	+ SpatialVID (10k)	78.45	68.82	0.077	0.560
flow-factored++	+ SpatialVID (20k)	**81.12**	**71.21**	**0.075**	**0.532**
3d-sup++	+ OmniWorld (3k)	78.68	70.26	0.080	0.565

Table 2. **Does factored flow prediction help dynamic visual geometry learning?** We train seven model variants on OmniWorld [42] and SpatialVID [32], where OmniWorld provides 3D supervision and SpatialVID offers flow supervision. Consistent with our findings on static scenes, flow-factored with factored flow prediction considerably outperforms the no-flow baseline (3d-sup) and other flow-supervised alternatives. Also, factored flow prediction brings consistent gains by using more data.

4. Experiments

We validate the effectiveness of the proposed factored flow prediction strategy in Flow3r. In Sec. 4.1, we compare our factored prediction paradigm against alternative designs and no-flow baselines, showing that incorporating flow supervision through our formulation consistently improves visual geometry learning for both static and dynamic scenes. In Sec. 4.2, as an effort to demonstrate scalability, we adapt an off-the-shelf visual geometry network (*i.e.*, VGGT [31]), integrate our factored flow prediction head, and scale up training using unlabeled dynamic video data.

4.1. Flow Supervision Improves Visual Geometry

We investigate whether incorporating flow supervision into a no-flow baseline trained under full 3D supervision (*i.e.*, camera poses, depths, and pointmaps) can improve visual geometry learning. Also, we further ask: *Can flow supervision help reduce (or even close) the performance gap when 3D supervision is insufficient?* We also compare our factored prediction formulation with other alternatives.

Experimental Designs. First, we include two models trained with full 3D supervision with different numbers of training sequences (denoted as 3d-sup and 3d-sup++). Next, building upon the no-flow baseline 3d-sup, we introduce three additional variants that incor-

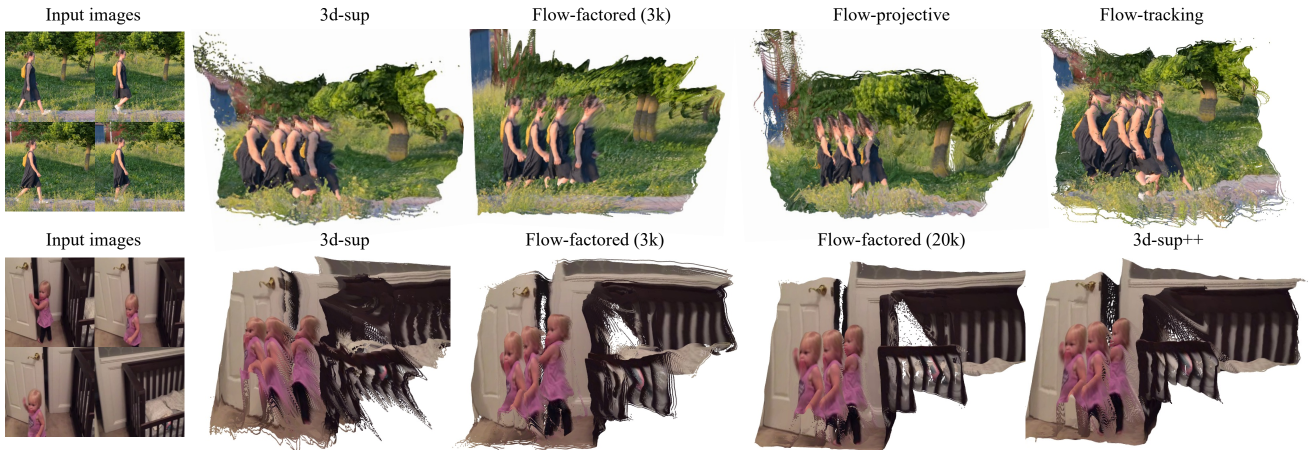

Figure 4. **Factored flow prediction aids visual geometry learning.** Compared with the baseline (3d-sup) and alternative formulations that use flow supervision (flow-projective, flow-tracking), Flow3r (flow-factored) yields more accurate dynamic-scene geometry and further improves with additional training data. This shows the effectiveness of factored flow prediction for geometry learning.

Methods	Kinetics700				EPIC-KITCHENS				Sintel				Bonn			
	RPE trans↓	RPE rot↓	**MSE↓**	**f-score@5↑**	RPE trans↓	RPE rot↓	**MSE↓**	**f-score@5↑**	RPE trans↓	RPE rot↓	**MSE↓**	**f-score@5↑**	RPE trans↓	RPE rot↓	**MSE↓**	**f-score@5↑**
VGGT	0.038	1.392	0.347	0.479	0.049	3.025	1.220	0.617	0.086	0.222	0.595	0.011	0.096	**6.021**	0.082	0.884
Flow3r*	0.033	1.360	0.258	0.623	0.053	3.262	0.230	0.606	0.126	1.753	0.620	0.278	0.095	6.246	0.082	0.894
Flow3r	**0.023**	**0.999**	**0.257**	**0.626**	**0.041**	**3.005**	0.218	**0.620**	**0.066**	1.154	**0.482**	**0.346**	0.094	6.175	**0.076**	**0.908**
π^{3*}	0.026	1.178	0.259	0.613	0.046	3.212	0.222	0.612	0.126	1.200	0.600	0.279	**0.093**	6.213	0.079	0.895
π^3	**0.023**	1.006	0.267	0.585	0.043	3.025	**0.200**	**0.620**	**0.066**	**1.122**	0.523	0.317	0.095	6.240	**0.076**	0.905
DUSt3R	0.063	9.343	0.366	0.533	0.110	8.492	0.312	0.528	0.179	15.166	0.622	0.271	0.113	6.384	0.116	0.800
CUT3R	0.027	1.988	0.303	0.573	0.081	4.709	0.338	0.493	0.128	1.998	0.676	0.217	0.095	6.349	0.088	0.899

Table 3. **Comparison on dynamic datasets.** Best, second-best, and third-best results are highlighted in light red, orange, and yellow, respectively. Flow3r outperforms other methods in both camera pose estimation and scene reconstruction, demonstrating its effectiveness.

porate flow supervision using different formulations: (1) flow-projective, computes flow explicitly from predicted camera poses and pointmaps via projective geometry; (2) flow-tracking, adopts a VGGT-style [31] tracking head based on pairwise patch features; (3) flow-factored applies our proposed factored flow prediction formulation.

Static Scenes. For static scenes, we conduct experiments on ScanNet++ [37], where model variants with flow supervision apply flow loss only to the additional training sequences used in 3d-sup++, ensuring that all model variants (except 3d-sup) are trained with the same total number of sequences.

Dynamic Scenes. For dynamic scenes, we experiment on OmniWorld [42] for 3D supervision. For those with flow supervision, we leverage a large-scale dynamic dataset, SpatialVID [32], for applying flow loss. To investigate the scaling behavior of the proposed factored flow prediction, we additionally include two expanded variants of flow-factored that use even more sequences for flow supervision, denoted as flow-factored+ and flow-factored++.

For these experiments, we measure both camera pose metrics and geometric metrics. Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA) measure the accuracy of predicted camera rotations and translations, while Chamfer Distance (CD) and Mean-Square Error (MSE) assess the geometric correctness of the reconstructed 3D structure.

Results on Static Scenes. The results in Tab. 1 show that flow-factored outperforms both flow-supervised alternatives (flow-projective and flow-tracking) as well as the no-flow baseline, achieving higher camera pose accuracy and better geometric quality. Notably, flow-tracking provides almost no improvement in pose accuracy and geometric quality, suggesting that supervising flow prediction from pairwise patch features does not meaningfully benefit visual geometry learning. Moreover, flow-projective even degrades performance on both pose metrics and geometry quality, indicating that supervising flow computed from explicit camera and geometry predictions may suffer from instability and thus harm learning. Compared to 3d-sup++, the no-flow baseline trained with full 3D supervision, flow-factored achieves comparable pose accuracy and geometry quality while even slightly improving camera center accuracy and reconstruction MSE.

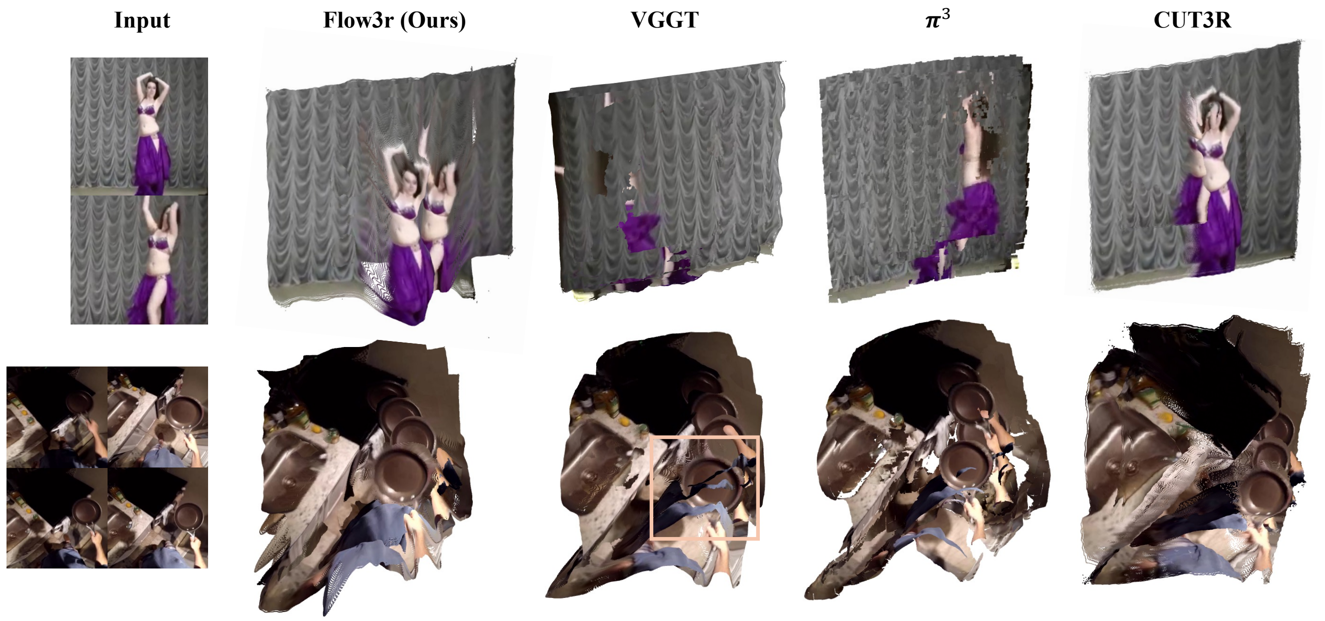

Figure 5. **Qualitative results on in-the-wild videos.** While other methods fail to reconstruct the scene accurately and often align it to moving object (top row), `Flow3r` robustly recovers dynamic scenes from in-the-wild videos, even under complex motion.

These results demonstrate that our proposed factored flow prediction serves as an effective approach of scaling visual geometry learning on static scenes, especially when dense 3D labels are insufficient.

Results on Dynamic Scenes and Scaling. The results on dynamic scenes are presented in Tab. 2, and they exhibit a performance pattern consistent with our observations on static scenes: our factored flow prediction significantly improves over the no-flow baseline (`3d-sup`), whereas flow supervision via VGGT's tracking head provides negligible gains, and supervising flow computed from explicit camera and geometry predictions continues to degrade performance. Also, the results reveal that increasing the number of training sequences to a larger scale (*e.g.*, 10× or 20× the amount used in the `3d-sup` no-flow baseline) yields consistent improvements. Notably, the `flow-factored++` variant – trained with 20K unlabeled dynamic video sequences in addition to 1K 3D-labeled sequences – surpasses `3d-sup++`, which uses 3K labeled sequences for full 3D supervision. These results demonstrate that supervising flow through our factored prediction formulation can effectively scale visual geometry learning, leveraging large quantities of unlabeled, dynamic video data.

Visualization. In Fig. 4, we qualitatively compare our method with baselines under full 3D supervision and those with different flow formulations. Notably, `flow-factored` significantly improves reconstruction quality over the no-flow baseline (`3d-sup`) while outperforming other flow-supervised model variants. Also, leveraging more data also brings non-trivial gains in reconstruction quality and the resulting model performs comparably with or even surpasses the baseline with largest number of training sequences under full 3D supervision (`3d-sup++`).

4.2. Scaling Visual Geometry Learning

Built upon the findings from Sec. 4.1, we scale the training of an off-the-shelf large visual geometry network (*i.e.*, VGGT [31]) by leveraging our factored flow prediction strategy with unlabeled dynamic data. In the following, we first describe the experimental setups, our training strategies, and baselines. Then, we present the resulting performance along with our analysis.

Datasets. We train our model on a diverse mixture of labeled 3D datasets and large-scale unlabeled video data. Our 3D supervision set consists of eight widely-used multi-view reconstruction datasets – CO3Dv2 [25], Habitat [26], ARKitScenes [3], ScanNet [7], ScanNet++ [37], MegaDepth [19], BlendedMVS [19], and StaticThings3D[28] – which provide ground-truth camera poses and geometry for supervised learning. To further scale correspondence supervision, we incorporate three unlabeled 4D video datasets: Kinetics-700 [5], SpatialVID [32] (high-quality dynamic sequences), and EPIC-Kitchens [8]. These dynamic datasets offer extensive appearance and motion diversity.

Training Overview. Our model training proceeds in two stages. First, we initialize our model from a pretrained VGGT [31] checkpoint and remove the first camera token to

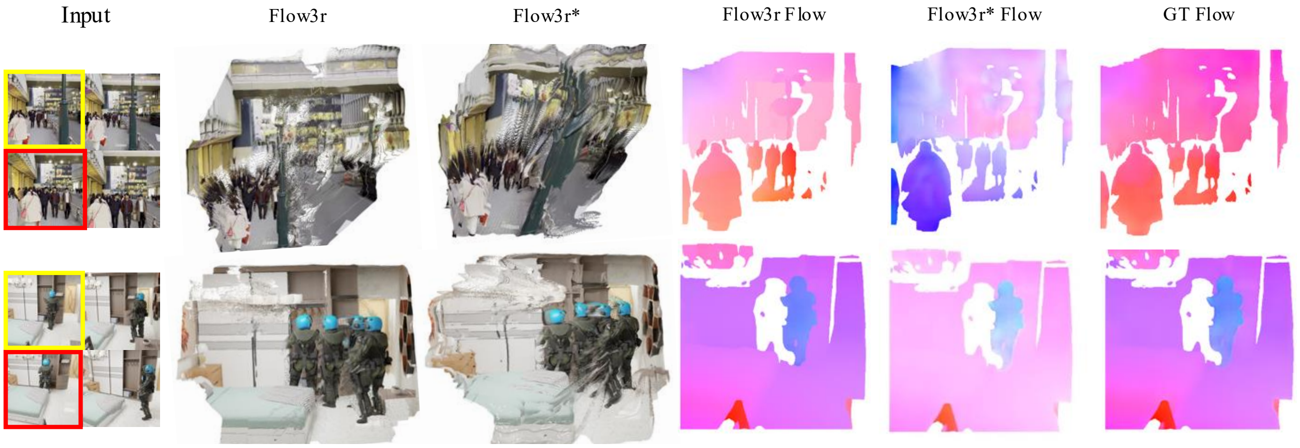

Figure 6. **Comparision on utilizing large-scale unlabeled dataset.** Compared with `Flow3r*`, `Flow3r` more accurately predicts dense flow and geometry on dynamic datasets, demonstrating the effectiveness of using large-scale unlabeled data via factored flow prediction.

enable permutation-equivariant prediction, while appending our factored flow prediction head to the model. In this stage, we use all labeled 3D data to train VGGT backbone and the newly-added flow prediction head separately (*i.e.*, the gradients from flow supervision would not affect the VGGT backbone). We denote this model by `Flow3r*`. In the second stage, we add additional unlabeled video datasets for training, and unfreeze the whole model, performing end-to-end finetuning for the whole model. The resulting model is denoted by `Flow3r`.

Baselines and Metrics. We compare `Flow3r` (and `Flow3r`) with CUT3R [33], VGGT [31], and π^3 [35]. Since the official π^3 checkpoint was trained with unreleased dynamic-scene data, we re-implement π^3 and train a model (denoted as π^3) using the same training data as our method to ensure a fair comparison. CUT3R is trained on a considerably larger pool of data (30+ datasets spanning diverse domains), whereas VGGT and our π^{3*} baseline are trained on the same data as `Flow3r`.

We evaluate performance using pose accuracy and reconstruction metrics in four dynamic datasets: Kinetics700 [5], Epic-Kitchens [8], Sintel [4] an Bonn [24], using MegaSAM [20] to compute 'ground truth' from dense videos on the first two. Following prior work [6, 38], we report Relative Pose Error, including RPE (trans) and RPE (rot). We assess 3D geometry using mean squared error (MSE), which captures overall geometric fidelity, and F-score to evaluate the accuracy–completeness trade-off.

Results. We report our evaluations in Tab. 3. For both pose estimation and scene reconstruction, `Flow3r` consistently outperforms baselines that use comparable training data, *e.g.*, VGGT and π^{3*}. Although the official π^3 model is trained on more data, `Flow3r` performs comparably on most metrics and even outperforms π^3 on a few metrics, *e.g.*, pose accuracy on Epic-Kitchens and reconstruction quality on Sintel, demonstrating the benefit from leveraging unlabeled video data via our factored flow prediction. We include qualitative results on in-the-wild videos in Fig. 5, where `Flow3r` infers cleaner and more accurate scene structure than baselines. We also observe that `Flow3r` consistently outperforms `Flow3r*` by a large margin, demonstrating the effectiveness of scaling with large amounts of unlabeled data. A visual comparison between the two models in Fig.6 further shows significant improvements in both the predicted flow fields and the scene geometry.

5. Discussion

In this work, we present `Flow3r` and demonstrate that it effectively leverages in-the-wild unlabeled data by introducing factored flow prediction, advancing visual geometry learning beyond existing fully supervised methods. While our approach opens up new possibilities, several challenges remain. First, `Flow3r` relies on off-the-shelf models to provide pseudo-ground-truth flow supervision, and there can be domains where such 2D prediction fails, limiting the performance upper bound of `Flow3r`. Second, although our factored flow formulation elegantly handles dynamic scenes and enables flow supervision to improve the learning of both camera motion and scene geometry, `Flow3r` may struggle under complex scenes with multiple moving independently components. Finally, our current experiments operate at a moderate scale (∼800K video sequences for flow supervision), and scaling to truly large-scale settings (10-100M videos) presents an exciting but unexplored direction. While this is out of scope for our work due to computational constraints, we envision `Flow3r`'s formulation serving as a building block for future large-scale learning methods.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 1, 3

[2] 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*, pages 6290–6301, 2022. 2

[3] 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. In *NeurIPS D&B*, 2021. 7

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 8

[5] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset, 2019. arXiv preprint arXiv:1907.06987. 7, 8

[6] Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. Leap-vo: Long-term effective any point tracking for visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19844–19853, 2024. 8

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 7, 2

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 7, 8

[9] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 2

[10] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 2

[11] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023. 2

[12] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 2, 5

[13] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *European conference on computer vision*, pages 368–381. Springer, 2010. 1

[14] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022. 2

[15] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2

[16] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European conference on computer vision*, pages 18–35. Springer, 2024. 2

[17] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 2

[18] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 2

[19] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 7

[20] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10486–10496, 2025. 2, 8

[21] Shaohui Liu, Yidan Gao, Tianyi Zhang, Rémi Pautrat, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Robust incremental structure-from-motion with hybrid features. In *European Conference on Computer Vision*, pages 249–269. Springer, 2024. 1

[22] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, pages 674–679, 1981. 2

[23] 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. In *TMLR*, 2024. 3, 1

[24] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. 2019. 8

[25] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 7, 2

[26] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, 2019. 7

[27] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 2, 3

[28] Philipp Schröppel, Jan Bechtold, Artemij Amiranashvili, and Thomas Brox. A benchmark and a baseline for robust multi-view depth estimation. In *3DV*, 2022. 7

[29] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 2

[30] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 1

[31] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 1, 2, 3, 4, 5, 6, 7, 8

[32] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations, 2025. arXiv preprint arXiv:2509.09676. 5, 6, 7

[33] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 1, 2, 8

[34] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2, 5

[35] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning, 2025. arXiv preprint arXiv:2306.17140. 3, 5, 8, 1, 2

[36] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 1

[37] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 5, 6, 7

[38] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 2, 8

[39] Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, et al. Ufm: A simple path towards unified dense correspondence with flow. *arXiv preprint arXiv:2506.09278*, 2025. 2, 4, 5

[40] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 2

[41] Qitao Zhao, Amy Lin, Jeff Tan, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Diffusionsfm: Predicting structure and motion via ray origin and endpoint diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6317–6326, 2025. 1, 2

[42] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling, 2025. arXiv preprint arXiv:2509.12201. 5, 6

[43] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. 2

Flow3r: Factored Flow Prediction for Visual Geometry Learning

Supplementary Material

Overview

The supplementary material includes sections as follows:

- **Section A**: Implementation details.
- **Section B**: Camera and geometry supervision from labeled data.
- **Section C**: More quantitative results.
- **Section D**: More qualitative comparisons of predicted geometry against baseline methods.

A. Implementation Details

In this section, we provide implementation and training details for all experiments of `Flow3r`.

Small Model. For ablation studies in Sec. 4.1, we use a compact model that preserves the overall architecture of `Flow3r` but with reduced capacity for efficiency. It employs a DINOv2 [23] backbone with a 224×224 input resolution to extract patch features, followed by a 12-layer multi-view transformer in which each layer contains one frame-wise self-attention block and one global self-attention block, following the structure of VGGT [31]. To perform factored flow prediction, we modulate the target-view camera latent into the source-view patch tokens through a lightweight MLP, and then decode the fused latent representation using a DPT head to obtain dense flow fields. Despite being significantly smaller, this model maintains architectural compatibility with the full variant and supports all ablation analysis.

Full Model. The full model adopts the same general design as our small model but increases the model capacity for higher accuracy. It uses the 518×518 DINOv2 [23] backbone and expands the multi-view transformer to 24 layers, while keeping the decoder heads (camera, depth, pointmap, and flow) identical in structure to those used in the small model. This configuration provides stronger representational power and serves as the default architecture for all large-scale experiments in Sec. 4.2.

Training Procedure. We train `Flow3r` in two stages. In the first stage (`Flow3r*`), we initialize the model from a pretrained VGGT checkpoint and train the camera, depth, and pointmap heads for 100k steps on all labeled 3D datasets. We use Adam with a learning rate of 1e-5 and a per-GPU batch size of 2 across 8 A100 GPUs (global batch size 16). We also apply gradient accumulation every 24 steps. The number of input views is randomly sampled between 2 and 8. After this supervised stage, we freeze the backbone and train only the factored flow head for an additional 50k steps. This flow-only phase uses a different configuration, with 8 A100 GPUs, a per-GPU batch size of 8 (no gradient accumulation), and the same learning rate.

In the second stage (`Flow3r`), we incorporate unlabeled dynamic-video datasets, unfreeze the entire model, and train end-to-end for another 100k steps across 8 H100 GPUs using the same per-GPU batch size of 2, the same randomized view sampling strategy, and a slightly higher learning rate of 2e-5. This two-stage procedure stabilizes geometry learning under full supervision before introducing large-scale flow supervision from unlabeled videos.

B. Camera and Geometry Supervision from Labeled Data.

This section introduces details for camera and geometry supervision from labeled 3D data.

Scene Scale Normalization. To avoid tying the representation to any particular reference frame, we compute losses for geometry quantities on *normalized* predictions. Specifically, given a set of predicted 3D points $\{\hat{\mathbf{P}}_i\}_{i=1}^{K}$, we first compute their centroid $\bar{\mathbf{P}}$ and define the scene scale s as the mean Euclidean distance of all points to this centroid:

$$\bar{\mathbf{P}} = \frac{1}{K}\sum_{i=1}^{K}\hat{\mathbf{P}}_i, \qquad s = \frac{1}{K}\sum_{i=1}^{K}\left\|\hat{\mathbf{P}}_i - \bar{\mathbf{P}}\right\|_2. \tag{9}$$

We then divide all predicted 3D quantities by this scale. This mean-distance value itself serves as the absolute scene scale that the model implicitly learns, and normalizing by it removes the need for the network to resolve any additional global scale during supervision. After this step, the camera, depth, and pointmap losses only need to account for their relative geometry, since the overall scale has already been fixed by our definition.

Camera Supervision. Camera head takes implicit camera features as input, and predicts both intrinsic and extrinsic camera parameters. The output includes the translation vector, rotation quaternion, and field-of-view parameters. We adopt the relative rotation loss as described in π^3 [35], which minimizes the geodesic distance (angular error) between the predicted relative rotation:

$$\mathcal{L}_{\text{rot}}(i,j) = \arccos\left(\frac{\operatorname{Tr}\left(\mathbf{R}_{i\leftarrow j}^{\top}\hat{\mathbf{R}}_{i\leftarrow j}\right) - 1}{2}\right). \tag{10}$$

For translation supervision, we replace π^3 [35]'s relative translation loss with an aligned camera center loss, which we empirically find to perform better. Specifically, for the predicted camera centers $\{\hat{\mathbf{c}}_i\}$, we compute an optimal rigid transform $(\mathbf{R}^*, \mathbf{t}^*)$ that minimizes $\sum_i \|\mathbf{R}^*\hat{\mathbf{c}}_i + \mathbf{t}^* - \mathbf{c}_i\|_2^2$, and align the predicted centers as $\mathbf{c}_i^{\text{aligned}} = \mathbf{R}^*\hat{\mathbf{c}}_i + \mathbf{t}^*$. The camera center loss is defined as:

$$\mathcal{L}_{\text{center}} = \frac{1}{N}\sum_i \left\|\mathbf{c}_i^{\text{aligned}} - \mathbf{c}_i\right\|_1,$$

where N denotes the number of views in the input.

Geometry Supervision. We also predict dense pixel-aligned depth maps, pointmaps and confidence maps which indicate the uncertainty of the model in various regions. To achieve global permutation-equivariance, we estimate an optimal rigid transform that aligns the predicted pointmaps $\hat{\mathbf{P}}$ to ground truth $\mathbf{P}^{gt}$ and compute an ℓ_2 loss on the aligned points:

$$\mathcal{L}_{\text{point}} = \frac{1}{K}\min_{\mathbf{R}\in SO(3),\, \mathbf{T}\in\mathbb{R}^3} \sum_k \|\mathbf{R}\hat{\mathbf{P}}_k + \mathbf{T} - \mathbf{P}_k^{gt}\|_2.$$

To prevent coordinate drift over training, we regularize the mean position of all predicted points to stay centered around the origin:

$$\mathcal{L}_{\text{reg}} = \left\|\frac{1}{K}\sum_k \hat{\mathbf{P}}_k\right\|_2. \tag{11}$$

The depth head is supervised using the confidence-weighted regression loss introduced in DUSt3R [34]. Specifically, for each pixel i, we regress the predicted depth $\hat{D}_i$ to the ground-truth depth D_i using the predicted uncertainty Σ_i^D as a per-pixel confidence weight. The depth loss is:

$$\mathcal{L}_{\text{depth}} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\|\Sigma_i^D \odot (\hat{D}_i - D_i)\right\|_2 - \alpha\log\Sigma_i^D\right), \tag{12}$$

The total loss for supervising camera and geometric predictions is therefore:

$$\mathcal{L}_{\text{supervised}} = \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{center}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{point}} + \beta\mathcal{L}_{\text{reg}}. \tag{13}$$

C. Additional Quantitative Results

Beyond the main results, we provide additional quantitative evaluations on several more datasets (see Tables 4, 5, 6, 7, 8, 9, 10, 11). These include the datasets from the main text, for which we report all relevant metrics, as well as several representative static-scene benchmarks such as 7-Scenes [29], ScanNet [7], CO3Dv2 [25], and NRGBD [2]. For camera evaluation, we follow the standard protocols for each setting: static-scene datasets are evaluated using Relative Rotation Accuracy (RRA), Relative Translation Accuracy (RTA), and the Area Under the Curve which combines the first two metrics, while dynamic-scene datasets are evaluated using Absolute Trajectory Error (ATE), Relative Pose Error for translation (RPE trans), and Relative Pose Error for rotation (RPE rot). For geometry evaluation, we adopt the same metrics for both static and dynamic datasets, reporting Accuracy, Completeness, Chamfer Distance, Mean-Squared Error, and F-score at 2% and 5% thresholds.

Across these results, we observe that `Flow3r` results in consistent improvements particularly on dynamic datasets where data with dense supervision is limited (although π^3 [35] uses additional synthetic data). On static-scene benchmarks, `Flow3r` is also competitive to the baseline (*e.g.* better on Scannet and 7-scenes), although we find that the performance on object-centric data (Co3D) does drop, perhaps because our unlabeled data is scene-centric. Overall, considering all datasets together, `Flow3r` shows a general improvement over the baselines, including methods trained with substantially more labeled data (*e.g.* the non-public synthetic data for training π^3 [35]).

D. More Qualitative Comparisons of Predicted Geometry against Baseline Methods

We additionally test our model on a broad set of in-the-wild videos spanning both static and dynamic scenes, including everyday scenarios with people, animals, vehicles, and complex background clutter. As shown in Fig.8, across these diverse examples, our method consistently produces competitive or better 3D geometry than baseline methods. We further provide full point-cloud visualizations in video form on the supplementary website.

Methods	**Kinetics700**								
	ATE↓	RPE trans↓	RPE rot↓	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	0.027	0.038	1.392	0.088	0.120	0.104	0.347	0.258	0.479
Flow3r*	0.022	0.033	1.360	0.066	**0.063**	**0.065**	0.258	**0.413**	0.623
Flow3r	**0.015**	**0.023**	**0.999**	0.064	0.077	0.070	**0.257**	0.403	**0.626**
π^{3*}	0.017	0.026	1.178	0.066	**0.063**	**0.065**	0.259	0.400	0.613
π^3	0.016	**0.023**	1.006	**0.059**	0.097	0.078	0.267	0.347	0.585
DUSt3R	0.045	0.063	9.343	0.083	0.106	0.095	0.366	0.317	0.533
CUT3R	0.019	0.027	1.988	0.070	0.076	0.073	0.303	0.352	0.573

Table 4. **Comparison on Kinetics700.**

Methods	**EPIC-KITCHENS**								
	ATE↓	RPE trans↓	RPE rot↓	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	0.032	0.049	3.025	0.061	0.069	0.065	0.220	0.415	0.617
Flow3r*	0.036	0.053	3.262	0.063	0.073	0.068	0.230	0.400	0.606
Flow3r	**0.030**	**0.041**	**3.005**	**0.058**	0.062	0.060	0.218	**0.461**	**0.620**
π^{3*}	0.033	0.046	3.212	0.063	0.065	0.064	0.222	0.429	0.612
π^3	0.032	0.043	3.025	0.069	**0.058**	**0.059**	**0.200**	0.459	**0.620**
DUSt3R	0.077	0.110	8.492	0.100	0.092	0.096	0.312	0.385	0.528
CUT3R	0.056	0.081	4.709	0.085	0.095	0.090	0.338	0.297	0.493

Table 5. **Comparison on EPIC-KITCHENS.**

Methods	**Sintel**								
	ATE↓	RPE trans↓	RPE rot↓	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	0.090	0.086	1.220	0.217	0.227	0.222	0.595	0.005	0.011
Flow3r*	0.093	0.126	1.753	0.230	0.200	0.215	0.620	0.134	0.278
Flow3r	**0.057**	**0.066**	1.154	**0.182**	**0.174**	**0.178**	**0.482**	**0.155**	**0.346**
π^{3*}	0.073	0.126	1.200	0.200	0.200	0.200	0.600	0.138	0.279
π^3	**0.060**	**0.066**	**1.122**	0.189	0.191	0.190	0.523	0.141	0.317
DUSt3R	0.152	0.179	15.166	0.255	0.224	0.240	0.622	0.124	0.271
CUT3R	0.111	0.128	1.998	0.221	0.269	0.245	0.676	0.111	0.217

Table 6. **Comparison on Sintel.**

Methods	**Bonn**								
	ATE↓	RPE trans↓	RPE rot↓	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	0.011	0.096	**6.021**	0.016	0.028	0.021	0.082	0.644	0.884
Flow3r*	0.017	0.095	6.246	0.019	**0.022**	0.021	0.082	0.692	0.894
Flow3r	**0.016**	0.094	6.175	0.016	**0.022**	**0.019**	**0.076**	**0.708**	**0.908**
π^{3*}	0.017	**0.093**	6.213	0.019	0.023	0.021	0.079	0.691	0.895
π^3	**0.016**	0.095	6.240	**0.014**	0.025	**0.019**	**0.076**	0.669	0.905
DUSt3R	0.055	0.113	6.384	0.029	0.037	0.033	0.116	0.546	0.800
CUT3R	0.021	0.095	6.349	0.018	0.023	0.021	0.088	0.658	0.899

Table 7. **Comparison on Bonn.**

Methods	**7-scenes**								
	RRA@30↑	RTA@30↑	AUC@30↑	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	100.0	86.51	69.77	0.052	0.056	0.054	0.182	0.395	0.665
Flow3r*	100.0	88.10	68.23	0.042	**0.036**	**0.039**	**0.157**	0.490	**0.750**
Flow3r	100.0	**90.67**	71.75	0.042	0.039	0.040	0.166	**0.496**	**0.750**
π^{3*}	100.0	88.09	68.26	0.042	0.040	0.041	0.167	0.492	**0.750**
π^3	100.0	87.69	**71.80**	**0.039**	0.045	0.042	0.169	0.473	0.737
DUSt3R	100.0	76.59	55.07	0.047	0.053	0.050	0.170	0.457	0.714
CUT3R	100.0	81.15	59.72	0.055	0.049	0.052	0.183	0.457	0.695

Table 8. **Comparison on 7-scenes.**

Methods	**NRGBD**								
	RRA@30↑	RTA@30↑	AUC@30↑	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	100.0	**99.21**	**93.13**	0.015	0.010	**0.012**	0.032	0.833	**0.964**
Flow3r*	100.0	98.01	87.50	0.023	0.012	0.018	0.046	0.758	0.920
Flow3r	100.0	98.02	87.50	0.023	0.012	0.017	0.046	0.758	0.920
π^{3*}	100.0	98.02	87.25	0.022	0.013	0.017	0.029	0.755	0.914
π^3	100.0	**99.21**	92.88	**0.014**	**0.009**	**0.012**	**0.027**	**0.868**	0.960
DUSt3R	100.0	93.25	76.04	0.031	0.024	0.027	0.063	0.662	0.865
CUT3R	100.0	95.63	76.90	0.042	0.019	0.030	0.075	0.549	0.808

Table 9. **Comparison on NRGBD.**

Methods	**Scannet**								
	RRA@30↑	RTA@30↑	AUC@30↑	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	100.0	93.71	71.37	**0.015**	0.019	**0.017**	**0.053**	**0.769**	0.931
Flow3r*	100.0	93.46	71.05	0.018	**0.016**	**0.017**	0.056	0.761	0.934
Flow3r	100.0	**94.82**	**72.11**	0.018	0.017	**0.017**	0.055	0.761	**0.935**
π^{3*}	100.0	93.47	71.06	0.019	0.017	0.018	0.056	0.760	0.933
π^3	99.75	91.14	69.39	0.016	0.022	0.019	0.058	0.762	0.930
DUSt3R	100.0	57.14	31.56	0.031	0.032	0.032	0.092	0.591	0.831
CUT3R	99.39	71.39	42.30	0.053	0.034	0.043	0.132	0.499	0.740

Table 10. **Comparison on Scannet.**

Methods	**Co3Dv2**								
	RRA@30↑	RTA@30↑	AUC@30↑	**Acc.**↓	**Comp.**↓	**CD**↓	**MSE**↓	**f-score@2%**↑	**f-score@5%**↑
VGGT	98.41	97.27	87.62	**0.022**	0.051	0.036	**0.151**	**0.707**	**0.874**
Flow3r*	98.29	96.68	82.96	0.030	**0.048**	0.039	0.186	0.633	0.849
Flow3r	98.54	96.81	80.52	0.036	0.051	0.043	0.213	0.556	0.812
π^{3*}	98.56	97.21	82.89	0.027	0.051	0.039	0.170	0.653	0.859
π^3	**98.82**	**97.49**	**90.53**	**0.022**	0.052	0.037	**0.151**	**0.707**	**0.874**
DUSt3R	97.07	90.91	66.83	0.036	0.069	**0.033**	0.223	0.567	0.783
CUT3R	93.85	90.68	68.11	0.047	0.082	0.064	0.278	0.507	0.737

Table 11. **Comparison on Co3Dv2.**

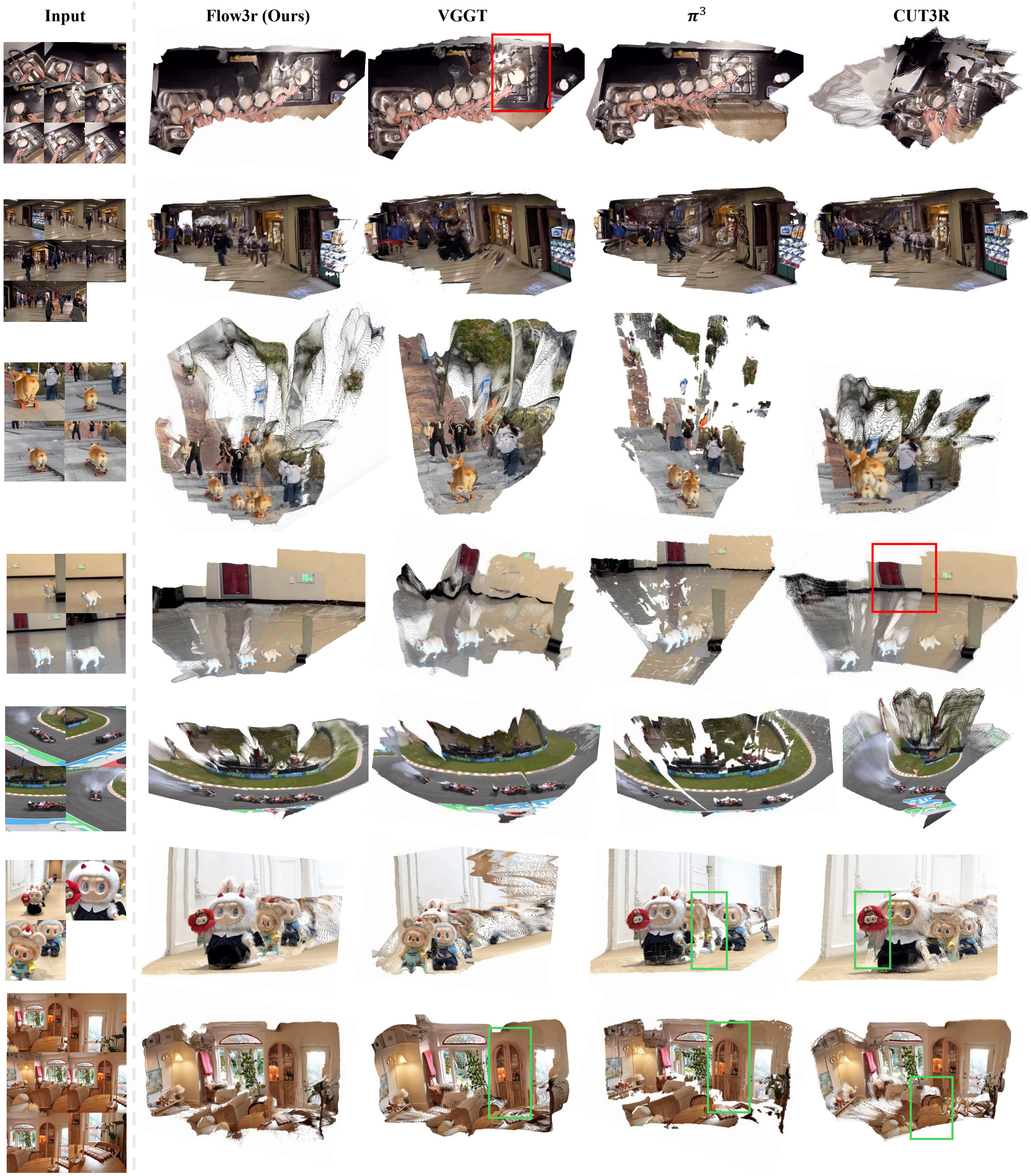

Figure 7. **More qualitative results.** The top six examples are dynamic scenes, where Flow3r tends to produce stable geometry under motion, though some challenging cases still show noticeable artifacts.

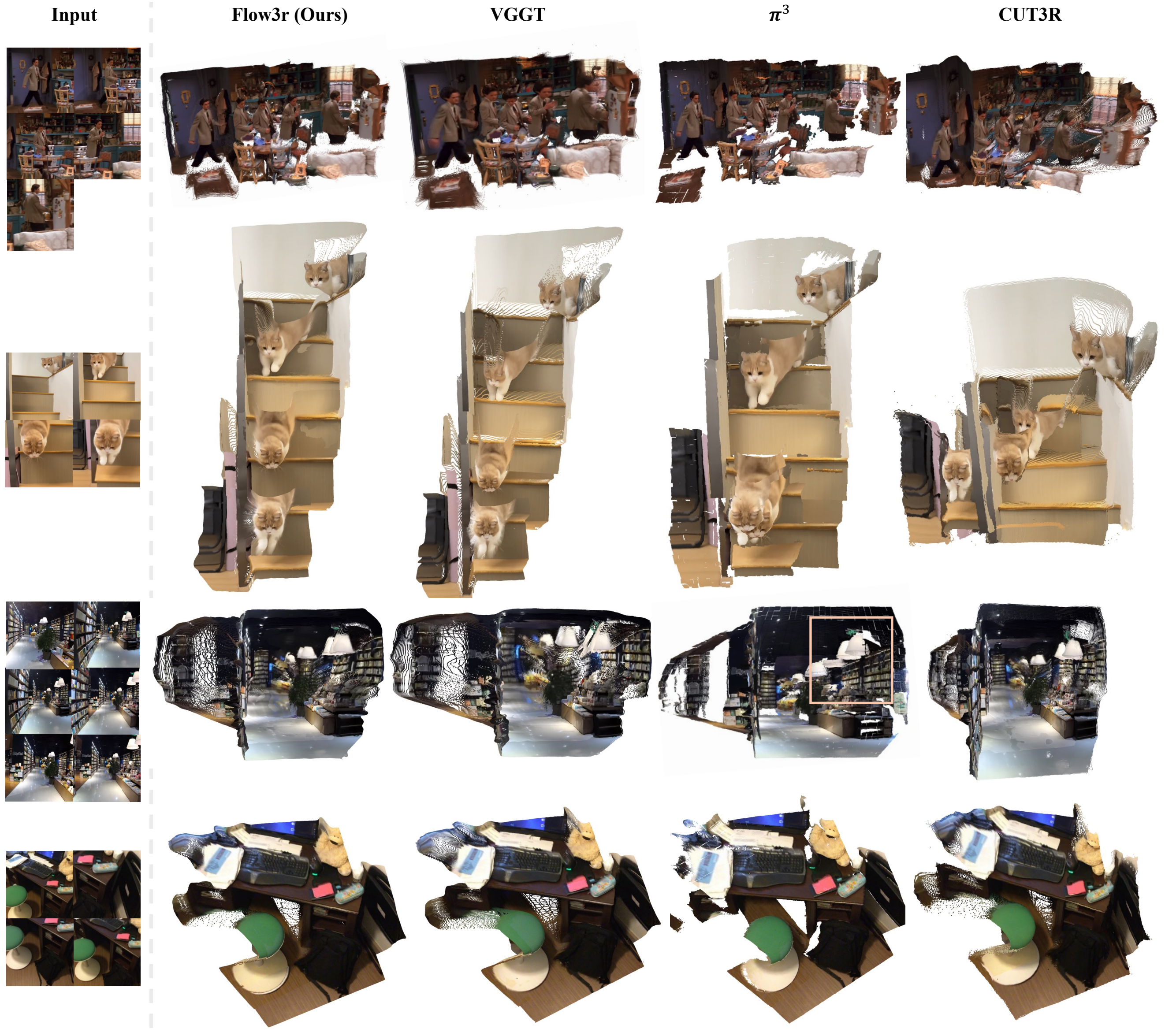

Figure 8. **More qualitative results.** The first two examples show dynamic scenes, where Flow3r better preserves both the moving object and the surrounding background geometry. In particular, in the second dynamic scene (the cat descending the stairs), Flow3r maintains more stable geometry for the cat while also keeping the staircase and nearby structures less distorted. The last two examples are static scenes, where the differences across methods are smaller. For more comprehensive comparisons, please refer to the videos in the supplementary webpage.