DROID-SLAM in the Wild

Moyang Li[1] Zihan Zhu[1] Marc Pollefeys[1,2] Daniel Barath[1]
[1]ETH Zurich [2]Microsoft

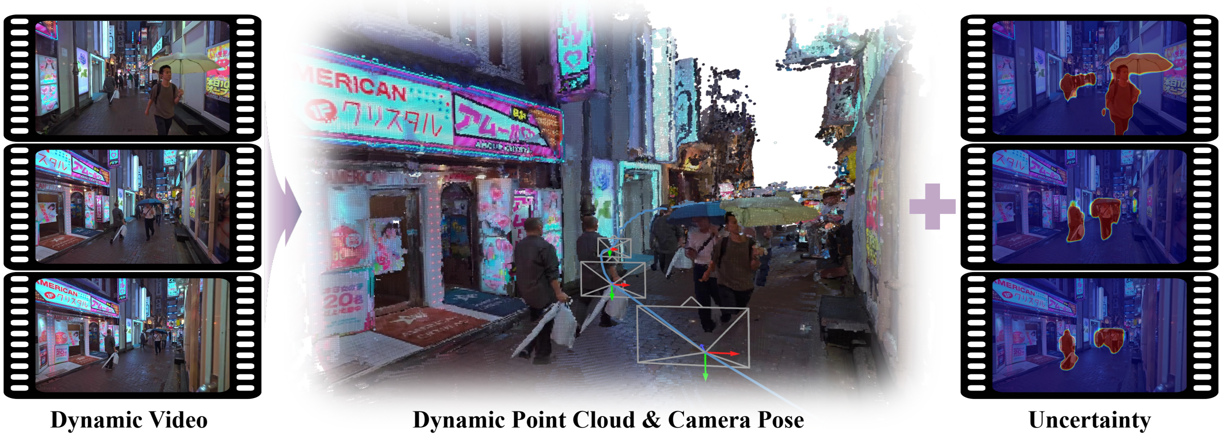

Figure 1. **DROID-W.** Given a casually captured *in-the-wild* video, our method estimates accurate dynamic uncertainty, camera trajectory, and scene structure, where existing SLAM baselines fail. *Left*: frames of the input video. *Middle*: reconstructed dynamic point clouds with estimated camera poses. *Right*: overlay of optimized uncertainty on the corresponding input frames.

Abstract

We present a robust, real-time RGB SLAM system that handles dynamic environments by leveraging differentiable Uncertainty-aware Bundle Adjustment. Traditional SLAM methods typically assume static scenes, leading to tracking failures in the presence of motion. Recent dynamic SLAM approaches attempt to address this challenge using predefined dynamic priors or uncertainty-aware mapping, but they remain limited when confronted with unknown dynamic objects or highly cluttered scenes where geometric mapping becomes unreliable. In contrast, our method estimates per-pixel uncertainty by exploiting multi-view visual feature inconsistency, enabling robust tracking and reconstruction even in real-world environments. The proposed system achieves state-of-the-art camera poses and scene geometry in cluttered dynamic scenarios while running in real time at around 8 FPS. The source code will be publicly released. See more results on our project page: moyangli00.github.io/droid-w.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental task in computer vision, with broad applications in autonomous driving [3, 12], robotics [1, 31, 69], and embodied intelligence [5, 15, 24]. Despite remarkable progress, achieving reliable SLAM in real-world environments is challenging. Dynamic and non-rigid objects often compromise pose estimation and 3D reconstruction, limiting the robustness and applicability of SLAM systems in practice.

Although this task has been extensively studied, many existing methods [10, 34–36, 48, 49] still assume a static environment and ignore non-rigid motion, which results in errors in both camera tracking and scene reconstruction. Some recent works [4, 18, 44, 55, 57] attempt to handle dynamic scenes by detecting or segmenting moving objects and masking out those regions. However, they rely heavily on prior knowledge of dynamic objects, which limits their robustness in complex and diverse real-world environments.

Recently, uncertainty-aware methods [25, 39, 66, 67] have attracted increasing attention for handling scene dynamics without relying on predefined motion priors. These approaches typically employ a shallow multi-layer perceptron (MLP) to estimate pixel-wise uncertainty from DINO [37] features and optimize the predictor through an online update. However, these approaches rely on constructing a perfectly static neural implicit [33] or Gaussian Splatting [21] map to optimize uncertainty. Consequently, their performance remains limited in complex real-world environments, where dynamic and cluttered scenes pose significant challenges for stable scene representation.

To address these limitations, we propose **DROID-W**, a novel dynamics-aware SLAM system that adapts prior deep visual SLAM system DROID-SLAM [48] to dynamic environments. We incorporate uncertainty optimization into the differentiable bundle adjustment (BA) layer to iteratively update dynamic uncertainty, camera poses, and scene geometry. The pixel-wise uncertainty of the frame is updated by leveraging multi-view visual feature similarity. In contrast with prior approaches, our uncertainty estimation is not constrained by high-quality geometric mapping or predefined motion priors. In addition, we introduce the DROID-W dataset, capturing diverse and unconstrained outdoor dynamic scenes, and further include *YouTube* clips for truly in-the-wild evaluation. In contrast to the saturated indoor benchmarks prevalent in prior works, our sequences feature challenging real-world settings with various object dynamics. Experimental results demonstrate that our approach achieves robust uncertainty estimation in real-world environments, leading to state-of-the-art camera tracking accuracy and scene geometry reconstruction while running in *real time* at approximately 8 FPS.

2. Related Works

Traditional Visual SLAM. Many existing traditional visual SLAM methods [9, 10, 34, 35, 48, 49] assume a static environment, which often leads to feature mismatching and degrades both tracking accuracy and mapping quality. To mitigate the disruption caused by object motion, some prior works [22, 23] implicitly handle dynamic elements through penalizing large frame-to-frame residuals during optimization. Other methods [38, 45] identify dynamic areas based on frame-to-model alignment residuals. StaticFusion [45] employs keypoint clustering and frame-to-model alignment to detect regions with large residuals, introducing a penalization term to constrain the map to static regions. ReFusion [38] adopts a TSDF [8] representation and removes uncertain regions with large depth residuals to maintain a consistent background map.

A complementary line of approaches [4, 40, 41, 60, 68] exploits object detection and segmentation to explicitly filter out dynamic regions. DynaSLAM [4] and DS-SLAM [60], both built upon ORB-SLAM2 [34], employ segmentation networks [2, 14] to detect moving objects and reconstruct a static background. Detect-SLAM [68] integrates the SSD detector [30] and propagates the moving probability of keypoints to reduce latency caused by object detection. Co-Fusion [40] and MaskFusion [41] extend to the object level, jointly segmenting, tracking, and reconstructing multiple independently moving objects. FlowFusion [63] instead leverages optical flow residuals to highlight dynamic regions.

NeRF- and GS-based SLAM. Recent advances in Neural Radiance Fields (NeRF) [33] have garnered substantial attention for their integration into SLAM systems, owing to their dense representation and photorealistic rendering capabilities. The pioneering work iMAP [47] introduces the first neural implicit SLAM framework, achieving high-quality dense mapping. However, iMAP [47] suffers from the loss of fine details and catastrophic forgetting, as it represents the entire scene in a single MLP. To overcome these limitations, NICE-SLAM [70] incorporates hierarchical feature grids to enhance scalability and reconstruction fidelity. Subsequent methods [19, 42, 51, 59, 64, 71] further improve the efficiency and robustness of such SLAM systems. More recently, the emergence of 3D Gaussian Splatting (3DGS) [21] inspired numerous SLAM approaches [13, 16, 20, 32, 43, 58] that adopt Gaussian primitives. However, these methods typically assume predominantly static environments, which limits their applicability in real-world scenarios with dynamic objects.

To overcome this limitation, several dynamic NeRF-based [18, 26, 44, 56] and GS-based SLAM systems [27, 29, 55, 57, 66, 67] have been proposed. Most of them [26, 27, 29, 55] rely on object detection or semantic segmentation to mask out dynamic regions, but struggle to handle undefined or unseen object classes. To address this, DynaMoN [44] introduces an additional CNN to predict motion masks from forward optical flow, while RoDyn-SLAM [18] and DG-SLAM [57] combine semantic segmentation with warping masks to improve motion mask estimation. WildGS-SLAM [66] and UP-SLAM [67] employ uncertainty modeling to handle scene dynamics. They utilize a shallow MLP to estimate per-pixel motion uncertainty from DINOv2 [37] features, as these features are robust to appearance variations and can represent abundant semantic information. The uncertainty MLP is optimized under the supervision of photometric and depth losses between input and rendered images. Furthermore, UP-SLAM [67] extends high-dimensional visual features into the 3DGS feature space and introduces a similarity loss as additional uncertainty constraints.

However, the optimization of uncertainty in these methods remains tightly coupled with scene representation, leading to performance degradation in complex environments where mapping struggles. In contrast, our approach adopts visual feature similarity between frames to estimate dynamic uncertainty, demonstrating robustness and effectiveness in challenging real-world environments.

Feed-forward Approaches. Recent feed-forward reconstruction and pose estimation methods have achieved remarkable progress. DUSt3R [54] and VGGT [52] demonstrate strong performance in scene geometry estimation. MonST3R [62] extends DUSt3R [54] to dynamic environments by estimating the dynamic mask from optical flow and pointmaps. Easi3R [6] introduces a training-free 4D reconstruction framework that isolates motion information

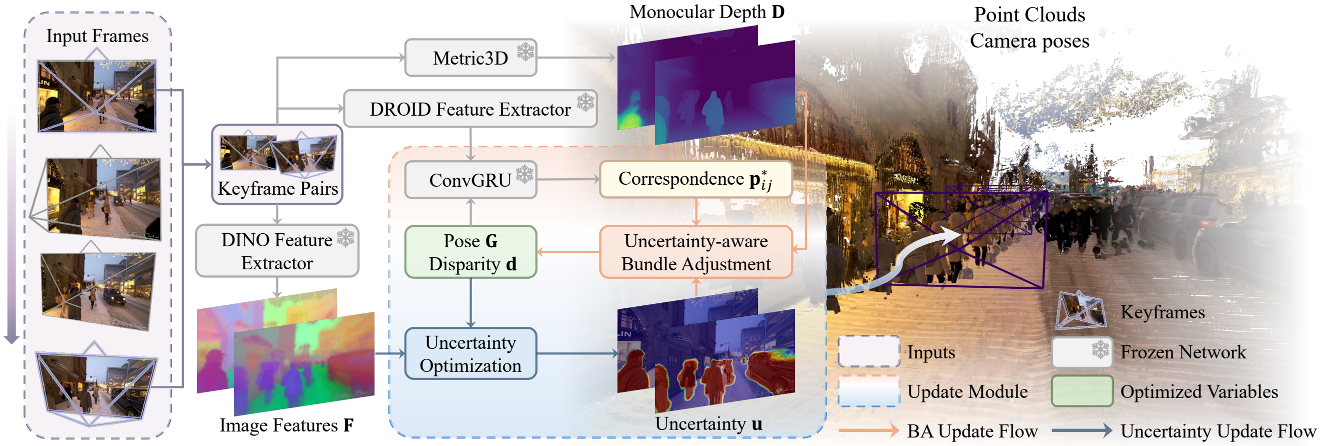

Figure 2. **System Overview.** The proposed DROID-W takes a sequence of RGB images as inputs and simultaneously estimates camera poses while recovering scene geometry. It alternatingly performs pose-depth refinement and uncertainty optimization in an iterative manner. The proposed uncertainty-aware dense bundle adjustment weights reprojection residuals with per-pixel uncertainty $\mathbf{u}$ to mitigate the influence of dynamic distractors. In addition, we use predicted monocular depth $\mathbf{D}$ as regularization of bundle adjustment, to improve its robustness under highly dynamic environments. For the uncertainty optimization module, we first extract DINOv2 [37] features from the input images and then iteratively update the dynamic uncertainty map by leveraging multi-view feature consistency. Specifically, feature consistency is measured by the cosine similarity between features of image $\mathbf{I}_i$ and its corresponding features in image $\mathbf{I}_j$, where the rigid-motion correspondences $\mathbf{p}_{ij}$ are derived using the current pose and depth estimates.

from the attention maps of DUSt3R [54]. However, these methods are restricted to short sequences. CUT3R [53] and TTT3R [7] further advance feed-forward reconstruction by handling long sequences in an online continuous manner. Despite these approaches achieving visually convincing geometry estimation, purely feed-forward pipelines often struggle to recover accurate camera trajectories and metrically consistent structure compared to SLAM-style systems. In contrast, our method, grounded in a visual SLAM framework, yields more accurate camera trajectories and reconstructions.

3. Proposed Method

Our approach adapts prior deep visual SLAM DROID-SLAM [48] by introducing a differentiable Uncertainty-aware Bundle Adjustment (UBA) that explicitly models per-pixel uncertainty to handle dynamic objects. Given RGB sequences from cluttered real-world scenes, our system optimizes camera poses, depth, and uncertainty to achieve robust tracking and accurate geometry estimation.

Next, we will first summarize the key components of DROID-SLAM designed for static environments (Sec. 3.1). We then present our proposed differentiable Uncertainty-aware Bundle Adjustment (Sec. 3.2) and dynamic uncertainty update (Sec. 3.3) modules. Finally, we introduce the proposed overall dynamic SLAM system (Sec. 3.4). The overview of **DROID-W** is shown in Fig. 2.

3.1. Preliminaries

DROID-SLAM leverages a differentiable bundle adjustment (BA) layer to update camera poses and depths in an iterative manner. For each RGB image in the input sequence $\{\mathbf{I}_t\}_{t=0}^N$, it maintains two state variables: camera pose $\mathbf{G}_t \in SE(3)$, inverse depth $\mathbf{d}_t \in \mathcal{R}^{\frac{H}{8}\times\frac{W}{8}}$. In addition, it constructs the frame-graph $(\mathcal{V}, \mathcal{E})$ to represent co-visibility across frames, where an edge $(i,j) \in \mathcal{E}$ means that the images $\mathbf{I}_i$ and $\mathbf{I}_j$ overlap. The set of camera poses $\{\mathbf{G}_t\}_{t=0}^N$ and inverse depths $\{\mathbf{d}_t\}_{t=0}^N$ are iteratively updated through the differentiable BA layer, operating on a set of image pairs $(\mathbf{I}_i, \mathbf{I}_j)$.

Differential Bundle Adjustment. For each pair of images $(\mathbf{I}_i, \mathbf{I}_j)$, we can derive the rigid-motion correspondence as:

$$\mathbf{p}_{ij} = \Pi_c\Big(\mathbf{G}'_{ij} \circ \Pi_c^{-1}(\mathbf{p}_i, \mathbf{d}'_i)\Big), \tag{1}$$

where Π_c denotes the camera projection function, and $\mathbf{G}'_{ij}$ is the relative pose between frames i and j. Variable $\mathbf{p}_i \in \mathcal{R}^{\frac{H}{8}\times\frac{W}{8}\times 2}$ represents a grid of pixel coordinates in frame i. DROID-SLAM predicts the 2D dense correspondence $\mathbf{p}_{ij}^* \in \mathcal{R}^{\frac{H}{8}\times\frac{W}{8}\times 2}$ and confidence map $\mathbf{w}_{ij} \in \mathcal{R}^{\frac{H}{8}\times\frac{W}{8}\times 2}$ in an iterative manner. The differentiable BA jointly refines camera poses and inverse depths by minimizing dense correspondence residuals as follows:

$$\mathbf{E}(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathcal{E}} \left\|\mathbf{p}_{ij}^* - \mathbf{p}_{ij}\right\|_{\mathbf{\Sigma}_{ij}}^2, \qquad \mathbf{\Sigma}_{ij} = \operatorname{diag}(\mathbf{w}_{ij}).$$

where $\|\cdot\|_\Sigma$ denotes Mahalanobis distance that weights the residuals according to the confidence map predicted by DROID-SLAM. The pose and disparity are optimized using the Gauss-Newton algorithm as follows:

$$\begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^\top & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\xi} \\ \Delta\mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \tag{2}$$

$$\begin{aligned} \Delta\boldsymbol{\xi} &= [\mathbf{B} - \mathbf{E}\mathbf{C}^{-1}\mathbf{E}^\top]^{-1}(\mathbf{v} - \mathbf{E}\mathbf{C}^{-1}\mathbf{w}), \\ \Delta\mathbf{d} &= \mathbf{C}^{-1}(\mathbf{w} - \mathbf{E}^\top\Delta\boldsymbol{\xi}). \end{aligned} \tag{3}$$

where $(\boldsymbol{\Delta\xi}, \boldsymbol{\Delta}\mathbf{d})$ represents pose and disparity update. Matrix $\mathbf{C}$ is diagonal as each term in Eq. (2) depends only on a single depth value, thus it can be inverted by $\mathbf{C}^{-1} = 1/\mathbf{C}$.

3.2. Uncertainty-aware Bundle Adjustment

Dynamic objects violate the rigid-motion assumption, yielding unreliable residuals that destabilize the BA layer of DROID-SLAM. To address this, we introduce a per-pixel dynamic uncertainty $\mathbf{u}_t \in \mathcal{R}^{\frac{H}{8} \times \frac{W}{8}}$ that downweights inconsistent correspondences during optimization. Intuitively, $\mathbf{u}_t$ acts as a confidence term penalizing high residuals caused by dynamic objects. Thus, we define uncertainty-aware Mahalanobis distance term $\|\cdot\|_{\Sigma^{\text{uncer}}_{ij}}$ as follows:

$$\boldsymbol{\Sigma}^{\text{uncer}}_{ij} = \text{diag}\,(\mathbf{w}_{ij} \cdot \frac{1}{\mathbf{u}'_i}). \tag{4}$$

However, jointly optimizing pose, depth, and uncertainty via Gauss-Newton algorithms is computationally prohibitive. We thus adopt an interleaved optimization strategy that alternates between pose-depth refinement and uncertainty optimization. The pose-depth refinement is performed by minimizing the following uncertainty-aware energy function:

$$\hat{\mathbf{E}}(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathcal{E}} \left\|\mathbf{p}^*_{ij} - \mathbf{p}_{ij}\right\|^2_{\boldsymbol{\Sigma}^{\text{uncer}}_{ij}}. \tag{5}$$

3.3. Uncertainty Optimization

For the optimization of dynamic uncertainty, we measure multi-view inconsistency via the similarity of DINOv2 [37] features across image pairs rather than the reprojection residuals in Eq. (5). Reprojection error can become unreliable under large dynamic motion, while 2D visual feature similarity yields a more stable and semantically meaningful measure for multi-view inconsistency.

Uncertainty Cost Function. For each pair of images $(\mathbf{I}_i, \mathbf{I}_j)$, 2D visual features $(\mathbf{F}_i, \mathbf{F}_j)$ are first extracted using FiT3D [61], a refined DINOv2 model. For each pixel $\mathbf{p}_i$ in frame i, we compute its rigid-motion correspondence $\mathbf{p}_{ij}$ in frame j via Eq. (1). We then obtain corresponding feature $\mathbf{F}_{ij}$ and uncertainty $\mathbf{u}_{ij}$ through bilinear interpolation. Multi-view consistency of the image pair is measured by cosine similarity between the DINOv2 features $(\mathbf{F}_i, \mathbf{F}_{ij})$. The dynamic objects in the environment with multi-view inconsistency are expected to have high uncertainty. Thus, we formulate the following similarity loss:

$$\mathbf{E}_{\text{sim}}(\mathbf{u}') = \sum_{(i,j)\in\mathcal{E}} \frac{1 - \frac{\mathbf{F}_i \cdot \mathbf{F}_{ij}}{\|\mathbf{F}_i\|_2 \|\mathbf{F}_{ij}\|_2}}{\mathbf{u}'_i \cdot \mathbf{u}'_{ij}}. \tag{6}$$

Here, we optimize bidirectional uncertainties for each image pair to decouple inter-frame dynamics.

To avoid the trivial solution of $\mathbf{u}' \to +\infty$, we regularize the uncertainty with a logarithmic prior:

$$\mathbf{E}_{\text{prior}}(\mathbf{u}') = \sum_i \log(\mathbf{u}'_i + 1.0). \tag{7}$$

Here, we add a bias term 1.0 to the uncertainty to prevent the prior loss from being negative.

Thus, the total uncertainty cost function is defined as:

$$\mathbf{E}_{\text{uncer}}(\mathbf{u}') = \mathbf{E}_{\text{sim}}(\mathbf{u}') + \gamma_{\text{prior}} \mathbf{E}_{\text{prior}}(\mathbf{u}'). \tag{8}$$

Uncertainty Regularization. Direct optimization of pixel-wise uncertainty may suffer from spatial inconsistency and overfitting to noise due to various dynamic motion. To address this, we learn a local affine mapping followed by the Softplus activation function from DINOv2 features to uncertainties. Thus, the uncertainty is obtained via $\mathbf{u} = \text{Softplus}(\boldsymbol{\theta} \cdot \mathbf{F})$. This affine mapping plays the role of a regularization term within the small local window, which is different from the decoder in prior works [39, 66].

Optimization. To avoid the inverse computation of the large Hessian matrix, we optimize uncertainty using Gradient Descent with weight decay instead of the Newton algorithm. All backpropagation operations are implemented in CUDA to ensure efficiency. The learnable parameters $\boldsymbol{\theta}$ of the affine mapping layer are updated as the following Jacobians:

$$\begin{aligned}
\boldsymbol{g}_t &= \sum_{i=0}^{N} \frac{\partial \mathbf{E}_{\text{uncer}}}{\partial \mathbf{u}'_i} \cdot \frac{\partial \mathbf{u}'_i}{\partial \boldsymbol{\theta}_{t-1}} \\
&= \sum_{i=0}^{N} \frac{\partial \mathbf{E}_{\text{uncer}}}{\partial \mathbf{u}'_i} \cdot \frac{1}{1 + \exp(-\boldsymbol{\theta}_{t-1} \cdot \mathbf{F}_i)} \cdot \mathbf{F}_i, \\
\boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} - \lambda \cdot \boldsymbol{g}_t - \eta \cdot \boldsymbol{\theta}_{t-1}.
\end{aligned} \tag{9}$$

For more details about the gradient derivations, please refer to the supplementary material.

3.4. SLAM System

Following DROID-SLAM, we accumulate 12 keyframes with sufficient motion to initialize the SLAM system. DROID-SLAM initializes the disparities as the constant value of 1, which can cause inaccurate tracking in high-dynamic scenes. Thus, we adopt the metric monodepth $\mathbf{D}_t \in \mathcal{R}^{\frac{H}{8} \times \frac{W}{8}}$ predicted by Metric3D [17] to penalize the disparity and improve accuracy. Thus, the cost function of BA with depth regularization is defined as follows:

$$\mathbf{E}^{+}(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathcal{E}} \left\|\mathbf{p}^*_{ij} - \mathbf{p}_{ij}\right\|^2_{\boldsymbol{\Sigma}^{\text{uncer}}_{ij}} + \gamma_d \sum_i \left\|\mathbf{d}'_i - \mathbf{D}_i\right\|^2.$$

After the initialization, we process incoming keyframes in an incremental manner. For newly added keyframes, we follow DROID-SLAM to perform local bundle adjustment in a sliding window and adopt depth regularization. For both initialization and frontend tracking stages, we optimize poses, disparities, and uncertainties. After frontend tracking, we perform global BA over all keyframes to refine camera poses and disparities. We freeze the dynamic-uncertainty parameters during global BA, since the affine transformation is intended to regularize uncertainty locally within the sliding window rather than at global scale.

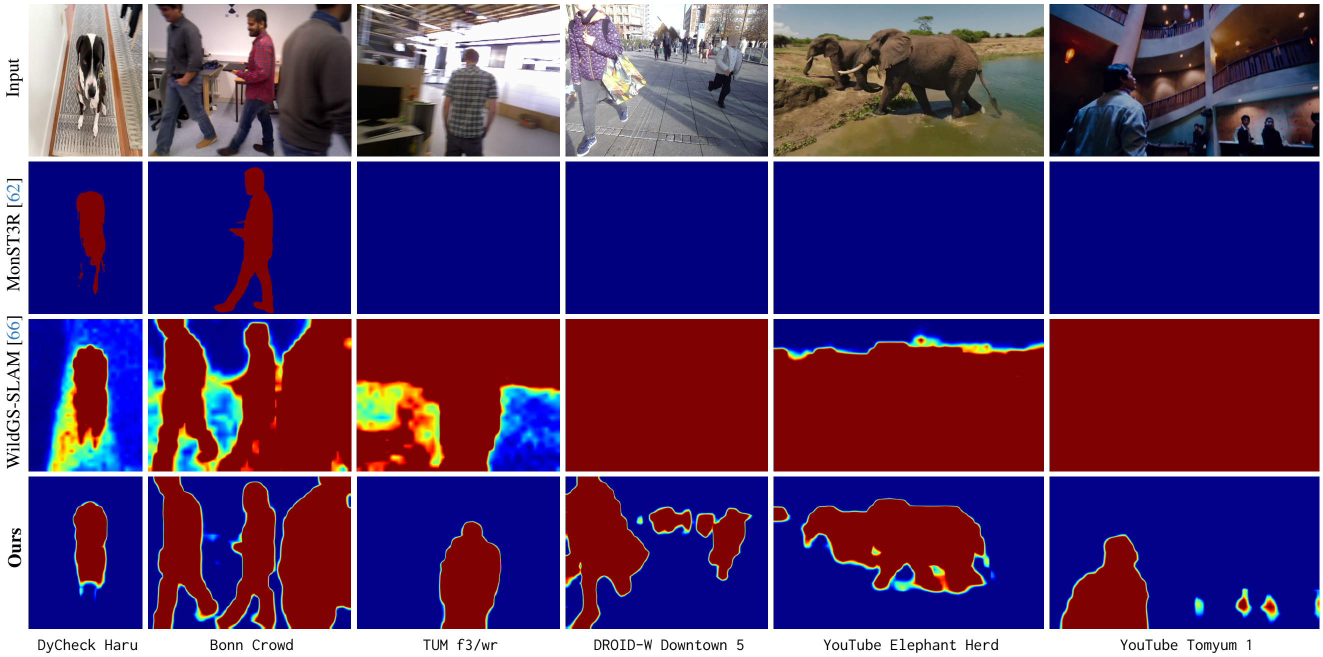

Figure 3. **Uncertainty Estimation.** WildGS-SLAM [66] and our approach estimate dynamic uncertainty, whereas MonST3R [62] predicts a binary motion mask. Our approach produces more accurate and spatially consistent uncertainty estimations across all challenging sequences.

4. Experiments

Datasets. We evaluate our approach on the Bonn RGB-D Dynamic dataset [38], TUM RGB-D dataset [46], and DyCheck [11] dataset. To further assess performance in ***unconstrained, outdoor*** settings, we introduce the DROID-W dataset, captured using a Livox Mid-360 LiDAR rigidly mounted with an RGB camera. The dataset comprises 7 sequences (`Downtown 1-7`) with RGB frames at a resolution of 1200×1600, ground-truth camera poses, and synchronized IMU and LiDAR measurements. Since satellite-based localization is unavailable for `Downtown 1-2`, we use FAST-LIVO2 [65] trajectories as ground truth, whereas the remaining sequences rely on RTK ground truth.

Additionally, we test on 6 dynamic videos downloaded from *YouTube*. The sequences span 8 seconds to 30 minutes, featuring diverse object motion and cluttered scenes. Sequences exceeding 5 minutes are partitioned into non-overlapping 5-minute segments due to resource bottlenecks of SLAM on a single GPU. For each video, the camera intrinsics are estimated with MonST3R [62] using 20 frames.

Baselines. We conduct comparisons with both ***SLAM-style*** and recent ***feed-forward*** methods. For SLAM-style methods, existing methods can be categorized into four groups: (a) *Classic SLAM*: DSO [10], ORB-SLAM2 [34], and DROID-SLAM [48]; (b) *Classic dynamic SLAM*: ReFusion [38] and DynaSLAM [4]; (c) *NeRF-/GS-based SLAM in static environments*: NICE-SLAM [70], and Splat-SLAM [43]; (d) *NeRF-/GS-based SLAM in dynamic environments*: DG-SLAM [57], RoDyn-SLAM [18], DDN-SLAM [26], DynaMoN [44], UP-SLAM [67], and ADD-SLAM [55]. For feed-forward approaches, we compare with MonST3R [62] and the very recent TTT3R [7].

Metrics. We use the Absolute Trajectory Error (ATE) to evaluate camera tracking accuracy. For the DyCheck dataset [11], we follow MegaSaM [28] and normalize the ground-truth camera trajectories to unit length, as the sequence lengths in this dataset vary significantly. Following DROID-SLAM, our approach performs optimization only for keyframes. To evaluate full trajectories, we recover non-keyframe poses through SE(3) interpolation followed by a pose graph update. For all methods, we align the estimated camera trajectory with the ground-truth camera trajectory through Sim(3) Umeyama alignment [50]. In addition to tracking accuracy, for each method, we report the average run-time by dividing the number of input frames by the total time.

4.1. Experimental Results

Quantitative Results. Camera tracking results on four benchmarks are reported in Tables 1, 2, 3, and 4. Table 1 indicates that our approach achieves the best camera tracking accuracy across all baselines on the Bonn RGB-D Dynamic dataset [38] due to effective uncertainty optimization. As shown in Table 2, WildGS-SLAM [66] exhibits a noticeable performance drop compared to DROID-SLAM [48] on low-dynamic sequences (`f3/sr`, `f3/shs`). This gap mainly stems from the unreliable uncertainty estimation, caused by challenging mapping in visually complex environments. In contrast, our method achieves comparable tracking accuracy

Method	Balloon	Balloon2	Crowd	Crowd2	Person	Person2	Moving	Moving2	**Avg.**
RGB-D									
NICE-SLAM [70]	24.4	20.2	19.3	35.8	24.5	53.6	17.7	8.3	22.74
ReFusion [38]	17.5	25.4	20.4	15.5	28.9	46.3	7.1	17.9	22.38
RoDyn-SLAM [18]	7.9	11.5	-	-	14.5	13.8	-	12.3	N/A
DynaSLAM (N+G) [4]	3.0	2.9	1.6	3.1	6.1	7.8	23.2	3.9	6.45
ORB-SLAM2 [34]	6.5	23.0	4.9	9.8	6.9	7.9	3.2	3.9	6.36
DG-SLAM [57]	3.7	4.1	-	-	4.5	6.9	-	3.5	N/A
DDN-SLAM (RGB-D) [26]	**1.8**	4.1	1.8	2.3	4.3	3.8	2.0	3.2	2.91
UP-SLAM [67]	2.8	2.7	-	-	4.0	3.6	-	3.2	N/A
ADD-SLAM [55]	2.7	**2.3**	-	-	**2.4**	3.7	-	**2.1**	N/A
RGB-only									
Splat-SLAM [43]	8.8	3.0	6.8	F	4.9	25.8	1.7	3.0	N/A
TTT3R [7]	21.5	15.4	9.8	7.7	30.0	21.4	33.4	41.2	22.55
DSO [10]	7.3	21.8	10.1	7.6	30.6	26.5	4.7	11.2	14.98
MonST3R [62]	7.2	6.0	6.6	6.9	9.8	16.1	3.5	6.7	7.85
DROID-SLAM [48]	7.5	4.1	5.2	6.5	4.3	5.4	2.3	4.0	4.91
DynaMoN (MS&SS) [44]	2.8	2.7	3.5	2.8	14.8	**2.2**	**1.3**	2.7	4.10
DynaMoN (MS) [44]	6.8	3.8	6.1	5.6	**2.4**	3.5	1.4	2.6	4.02
WildGS-SLAM [66]	2.8	2.4	1.6	2.2	3.9	3.1	1.7	2.5	2.52
DROID-W (Ours)	2.6	2.5	**1.3**	**1.9**	3.4	2.9	1.5	2.4	**2.31**

Table 1. **Tracking Performance on the Bonn RGB-D Dynamic Dataset [38]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first**, second, and third. "-" indicates sequences without reported results in the original papers or unavailable code. "F" denotes tracking failure. For MonST3R [62], we use the same keyframes as our method and perform evaluation in a window-wise manner with a window size of 20 and an overlap ratio of 0.5 to reduce memory consumption.

Method	f2/dp	f3/ss	f3/sx	f3/sr	f3/shs	f3/ws	f3/wx	f3/wr	f3/whs	**Avg.**
RGB-D										
NICE-SLAM [70]	88.8	1.6	32.0	59.1	8.6	79.8	86.5	244.0	152.0	83.60
ORB-SLAM2 [34]	**0.6**	0.8	1.0	2.5	2.5	40.8	72.2	80.5	72.3	30.36
ReFusion [38]	4.9	0.9	4.0	13.2	11.0	1.7	9.9	40.6	10.4	10.73
DynaSLAM (N+G) [4]	0.7	**0.5**	1.5	2.7	1.7	0.6	1.5	3.5	2.5	1.69
DG-SLAM [57]	3.2	-	1.0	-	-	0.6	1.6	4.3	-	N/A
RoDyn-SLAM [18]	-	-	-	-	4.4	1.7	8.3	-	5.6	N/A
DDN-SLAM (RGB-D) [26]	-	-	1.0	-	1.7	1.0	1.4	3.9	2.3	N/A
UP-SLAM [67]	1.3	-	0.9	-	-	0.7	1.6	-	2.6	N/A
ADD-SLAM [55]	-	-	-	-	**1.3**	0.5	1.4	-	**1.6**	N/A
RGB-only										
TTT3R [7]	113.1	3.1	5.8	6.4	24.9	2.0	24.7	15.9	23.1	24.33
MonST3R [62]	33.9	0.8	28.3	5.1	36.8	1.6	19.1	16.6	32.8	19.45
DSO [10]	2.2	1.7	11.5	3.7	12.4	1.5	12.9	13.8	40.7	11.15
DDN-SLAM (RGB) [26]	-	-	1.3	-	3.1	2.5	2.8	8.9	4.1	N/A
Splat-SLAM [43]	0.7	**0.5**	0.9	2.3	1.5	2.3	1.3	3.9	2.2	1.71
DynaMoN (MS) [44]	**0.6**	**0.5**	0.9	**2.1**	1.9	1.4	1.4	3.9	2.0	1.63
DynaMoN (MS&SS) [44]	0.7	**0.5**	0.9	2.4	2.3	0.7	1.4	3.9	1.9	1.63
DROID-SLAM [48]	**0.6**	**0.5**	0.9	2.2	1.4	1.2	1.6	4.0	2.2	1.62
WildGS-SLAM [66]	1.4	**0.5**	**0.8**	2.4	2.0	**0.4**	1.3	3.3	**1.6**	1.51
DROID-W (Ours)	1.2	**0.5**	**0.8**	2.2	1.4	0.5	**1.2**	**2.7**	**1.6**	**1.34**

Table 2. **Tracking Performance on TUM RGB-D Dataset [46]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first**, second, and third. "-" indicates sequences without reported results in the original papers or unavailable code. Our approach consistently leads to the best or second-best results on the sequences, on average, outperforming *all* baselines.

to DROID-SLAM on low-dynamic scenes and significantly outperforms it on high-dynamic sequences by effectively handling motion-induced inconsistencies.

The DyCheck dataset is characterized by motion and scene diversity across indoor and outdoor scenarios. Table 3 demonstrates that WildGS-SLAM often fails to achieve accurate camera tracking due to the difficulty of scene reconstruction in these complex settings and erroneous uncertainty estimation, whereas our method remains stable and accurate. On scene haru, where a moving dog dominates the view, our accurate uncertainty estimation suppresses dynamic regions. Consequently, fewer reliable background features remain to support tracking, which degrades our performance. On average, our proposed method outperforms all baselines. Table 4 presents the experimental results on the proposed large-scale outdoor dataset DROID-W. Our method shows superior per-

Method	apple	backpack	block	creeper	handwavy	haru	mochi	paper	pillow	spin	sriracha	teddy	**Avg.**
RGB-D													
NICE-SLAM [70]	0.186	0.149	0.099	0.166	0.059	F	0.042	0.062	0.171	0.211	0.073	0.060	N/A
DynaSLAM (N+G) [4]	0.981	0.045	0.731	1.709	0.796	0.322	1.263	F	0.713	0.322	1.098	0.296	N/A
RGB-only													
MonST3R [62]	1.236	0.013	1.141	0.324	0.125	0.263	0.089	0.037	1.118	0.118	0.060	0.279	0.400
TTT3R [66]	0.915	0.026	0.507	0.699	0.298	0.042	0.233	0.070	0.712	0.353	0.215	0.125	0.350
Splat-SLAM [43]	0.038	0.005	0.078	0.052	0.024	0.078	0.211	**0.011**	0.262	0.007	0.005	0.048	0.068
WildGS-SLAM [7]	0.043	0.006	0.047	0.029	0.016	0.085	**0.017**	0.013	0.378	**0.005**	0.005	0.027	0.056
DROID-SLAM [48]	**0.036**	0.008	0.156	**0.015**	0.016	**0.005**	0.018	**0.011**	0.179	0.009	0.010	0.061	0.044
DROID-W (Ours)	0.040	**0.004**	**0.037**	0.018	**0.015**	0.116	0.024	0.012	**0.140**	**0.005**	**0.004**	**0.019**	**0.036**

Table 3. **Tracking Performance on DyCheck Dataset [11]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first** , second , and third . "-" indicates sequences without reported results in the original papers or unavailable code. "F" means tracking failure. Our approach demonstrates the effectiveness and robustness in highly-textured, diverse environments, where prior methods relying on object segmentation or Gaussian mapping for uncertainty optimization often fail.

Method	Downtown 1	Downtown 2	Downtown 3	Downtown 4	Downtown 5	Downtown 6	Downtown 7	**Avg.**
TTT3R [7]	6.31	8.86	3.69	6.42	7.82	8.29	6.31	6.815
Splat-SLAM [43]	0.25	19.14	1.15	0.97	0.78	4.17	**0.05**	3.789
DROID-SLAM [48]	0.22	8.79	0.79	0.39	1.76	**0.06**	**0.05**	1.724
WildGS-SLAM [7]	**0.10**	0.83	0.61	0.38	0.74	0.89	0.52	0.580
DROID-W (Ours)	0.11	**0.37**	**0.37**	**0.32**	**0.51**	0.07	**0.05**	**0.258**

Table 4. **Tracking Performance on DROID-W Dataset** (ATE RMSE ↓ [m]). Best results are highlighted as **first** , second .

Method	Dynamic	Bonn [38]	TUM [46]	DyCheck [11]
DROID-SLAM [48]	✗	**19.89**	**26.97**	**17.50**
WildGS-SLAM [66]	✓	0.22	0.32	0.18
DROID-W (Ours)	✓	8.22	10.96	8.59

Table 5. **Runtime Comparisons** (average FPS ↑). All evaluations are conducted on an RTX 3090 GPU with a 16-core CPU.

Method	ATE RMSE [cm]
a. w/o Uncertainty-aware BA	5.13
b. w/o monocular depth	3.30
c. w/o uncertainty decouple	2.57
d. w/o affine mapping	2.47
e. w/o weight decay	2.34
Full	**2.31**

Table 6. **Ablation Studies on Bonn RGB-D Dataset [38].** Details about each configuration are described in Sec. 4.2.

formance over prior works under this extremely challenging condition. Feed-forward approaches such as MonST3R [62] and TTT3R [7] suffer from substantially higher tracking errors across all benchmarks compared to optimization-based SLAM systems.

Runtime analysis is in Table 5. We compare with DROID-SLAM and WildGS-SLAM, the most recent state-of-the-art baseline for monocular dynamic SLAM. Our system achieves a **40**× speedup over WildGS-SLAM and maintains real-time performance at approximately 8 FPS. Our approach is slightly slower than DROID-SLAM due to monocular depth estimation and DINOv2 [37] feature extraction. Overall, these results highlight the effectiveness, robustness, and efficiency of our uncertainty-aware formulation compared with existing SLAM-style and feed-forward baselines.

Qualitative Comparisons. Fig. 3 presents comparisons of the estimated uncertainty maps. We observe that our approach delivers the most accurate dynamic uncertainty estimates, whereas WildGS-SLAM produces erroneous results near moving objects and severely incorrect predictions on challenging sequences. As shown in Fig. 3, TUM RGB-D dataset features motion blur, partial overexposure, and cluttered indoor scenes that easily degrade mapping quality. Our introduced sequences offer diverse object motion and scene configuration, which poses challenges to high-quality geometric reconstruction. WildGS-SLAM exhibits degraded performance in these challenging sequences with low-quality imagery and highly textured backgrounds, where erroneous Gaussian reconstruction leads to unstable uncertainty estimates. MonST3R [62] heavily depends on the alignment of dynamic point clouds predicted by the pretrained model, which often results in incomplete or missed detections of moving objects due to limited generalizability.

In contrast, our method yields spatially coherent, semantically consistent uncertainty maps. It sharply delineates dynamic regions and maintains stable confidence in static areas across challenging scenarios, demonstrating the robustness of our uncertainty optimization.

Finally, we compare the reconstruction quality in challenging YouTube sequences. Fig. 4 illustrates that DROID-SLAM produces inaccurate point clouds in dynamic scenes, as moving distractors lead to unreliable reprojection residuals and disrupt pose estimation. The reconstructions of DROID-SLAM exhibit scale drift (`St. Moritz 1`), erroneous geometry (`St. Moritz 3`), and noisy distractors (`Tokyo Walking 1 & 2`). WildGS-SLAM struggles to reconstruct

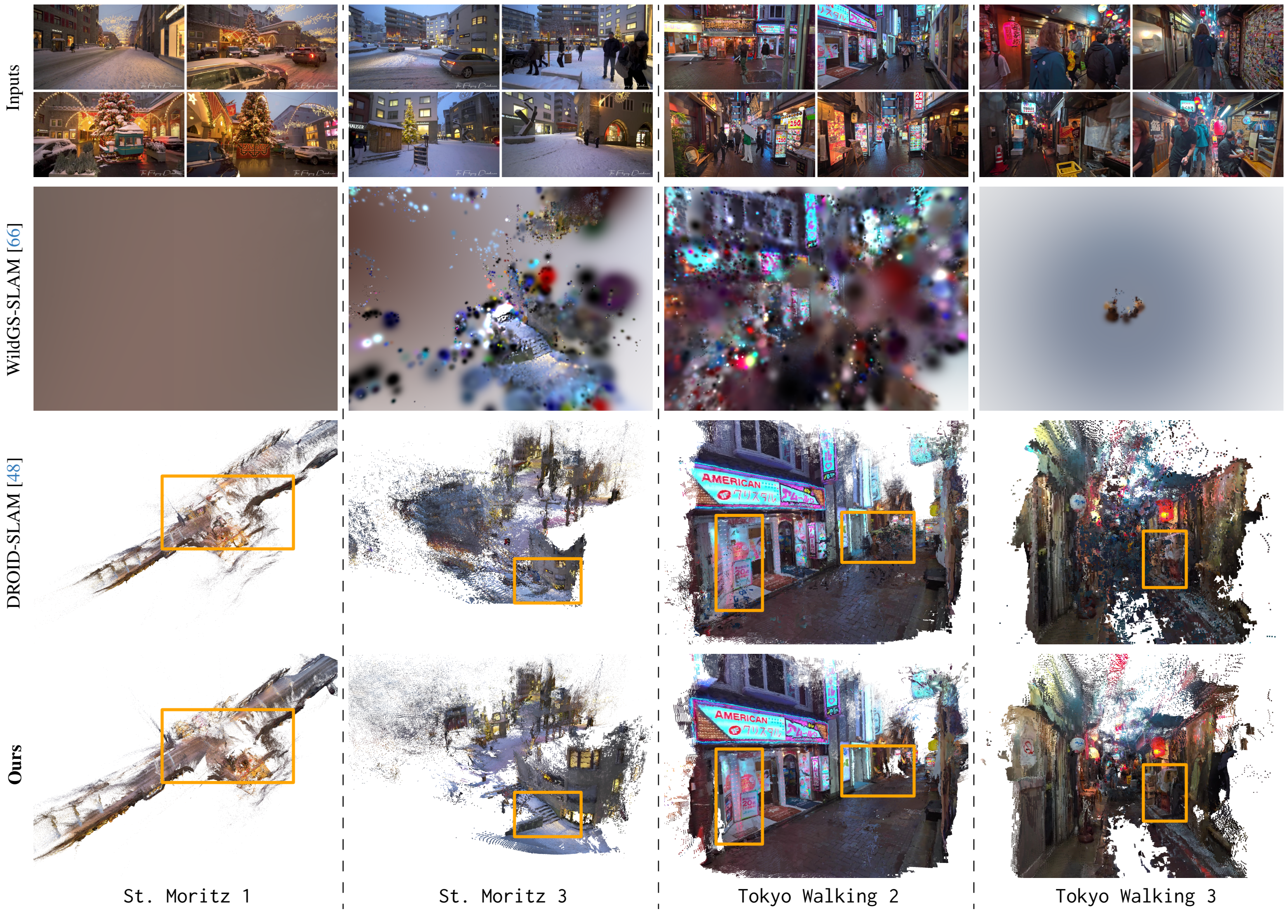

Figure 4. **3D Reconstruction Comparisons on YouTube Sequences.** We compare 3D reconstruction quality of DROID-SLAM [48], WildGS-SLAM [66], and our method. Point clouds from DROID-SLAM and ours are visualized directly, while Gaussian renderings from WildGS-SLAM are displayed using the 3DGS viewer. WildGS-SLAM fails on most sequences. DROID-SLAM shows obvious scale drift (`St. Moritz 1`), inaccurate geometry (`St. Moritz 3`), and noisy distractors (`Tokyo Walking 2 & 3`) under challenging dynamic environments. Our approach produces accurate and consistent reconstructions across highly dynamic and visually challenging real-world sequences.

Gaussian maps under these conditions, resulting in near-complete failures on all sequences. In contrast, our method yields geometrically accurate and temporally consistent point clouds, maintaining stable reconstruction quality even in challenging outdoor scenarios.

4.2. Ablation Study

We conduct ablations of the main modules in Table 6. In the *a. w/o Uncertainty-aware BA* setting, we disable the uncertainty update and weight the reprojection term solely by the confidence map. For the experiments *c. w/o uncertainty decouple*, we modify the similarity loss in Eq. (6) as follows:

$$\mathbf{E}^{*}_{\text{sim}}(\mathbf{u}') = \sum_{(i,j)\in\mathcal{E}} \frac{1 - \frac{\mathbf{F}_i \cdot \mathbf{F}_{ij}}{\|\mathbf{F}_i\|_2 \|\mathbf{F}_{ij}\|_2}}{{\mathbf{u}'_i}^2}. \tag{10}$$

In experiments *d. w/o affine mapping*, uncertainties are updated directly rather than by optimizing parameters of the affine mapping. Removing the affine mapping introduces temporal and spatial inconsistencies in uncertainty estimation, leading to degraded performance. In case of *e. w/o weight decay*, the lack of regularization term for affine mapping will cause instability, thereby leading to performance drop on some scenes. As shown in Table 6, the full system consistently outperforms all variants, validating the effectiveness of each component.

5. Conclusion

In this paper, we presented a novel monocular dynamic SLAM system. Our system optimizes dynamic uncertainty within a differentiable bundle adjustment framework by leveraging multi-view feature similarity. Extensive experiments demonstrate that our effective uncertainty optimization enables robust camera tracking and accurate geometric reconstruction across challenging real-world scenarios, where prior methods often struggle. Code will be public.

Limitations. Our uncertainty optimization relies on frame-to-frame alignment, which can lead to inaccurate uncertainty estimation during SLAM initialization when pose estimates are still unreliable. Incorporating reconstruction priors could improve the robustness of the initialization stage.

References

[1] Philip Arm, Gabriel Waibel, Jan Preisig, Turcan Tuna, Ruyi Zhou, Valentin Bickel, Gabriela Ligeza, Takahiro Miki, Florian Kehl, Hendrik Kolvenbach, et al. Scientific exploration of challenging planetary analog environments with a team of legged robots. *Science robotics*, 8(80):eade9548, 2023. 1

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 2

[3] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert systems with applications*, 165:113816, 2021. 1

[4] Berta Bescos, José M. Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes. *IEEE Robotics and Automation Letters (RA-L)*, 2018. 1, 2, 5, 6, 7

[5] Junting Chen, Guohao Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers. *arXiv preprint arXiv:2305.16925*, 2023. 1

[6] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. *arXiv preprint arXiv:2503.24391*, 2025. 2

[7] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025. 3, 5, 6, 7

[8] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *ACM Trans. on Graphics*, 1996. 2

[9] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 2

[10] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 1, 2, 5, 6

[11] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022. 5, 7

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 1

[13] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgbd gs-icp slam. In *European Conference on Computer Vision*, pages 180–197. Springer, 2024. 2

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[15] Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025. 1

[16] Jiarui Hu, Xianhao Chen, Boyin Feng, Guanglin Li, Liangjing Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In *European Conference on Computer Vision*, pages 93–112. Springer, 2024. 2

[17] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 4

[18] Haochen Jiang, Yueming Xu, Kejie Li, Jianfeng Feng, and Li Zhang. Rodyn-slam: Robust dynamic dense rgb-d slam with neural radiance fields. *IEEE Robotics and Automation Letters (RA-L)*, 2024. 1, 2, 5, 6

[19] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. *arXiv preprint arXiv:2211.11704*, 2022. 2

[20] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics*, 2023. 1, 2

[22] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 2100–2106. IEEE, 2013. 2

[23] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE international conference on robotics and automation*, pages 3748–3754. IEEE, 2013. 2

[24] Anusha Krishnan, Shaohui Liu, Paul-Edouard Sarlin, Oscar Gentilhomme, David Caruso, Maurizio Monge, Richard Newcombe, Jakob Engel, and Marc Pollefeys. Benchmarking egocentric visual-inertial slam at city scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25207–25217, 2025. 1

[25] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. In *Advances in Neural Information Processing Systems (NIPS)*, 2024. 1

[26] Mingrui Li, Jiaming He, Guangan Jiang, and Hongyu Wang. Ddn-slam: Real-time dense dynamic neural implicit slam with joint semantic encoding. *arXiv preprint arXiv:2401.01545*, 2024. 2, 5, 6

[27] Yanyan Li, Youxu Fang, Zunjie Zhu, Kunyi Li, Yong Ding, and Federico Tombari. 4d gaussian splatting slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25019–25028, 2025. 2

[28] 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. *arXiv preprint arXiv:2412.04463*, 2024. 5

[29] Haosong Liu, Long Wang, Haiyong Luo, Fang Zhao, Runze Chen, Yushi Chen, Mingyu Xiao, Jiaquan Yan, and Dan Luo. Sdd-slam: Semantic-driven dynamic slam with gaussian splatting. *IEEE Robotics and Automation Letters*, 2025. 2

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[31] Xu Liu, Jiuzhou Lei, Ankit Prabhu, Yuezhan Tao, Igor Spasojevic, Pratik Chaudhari, Nikolay Atanasov, and Vijay Kumar. Slideslam: Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation. *arXiv preprint arXiv:2406.17249*, 2024. 1

[32] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 1, 2

[34] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 2017. 1, 2, 5, 6

[35] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015. 2

[36] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 1

[37] 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. 1, 2, 3, 4, 7

[38] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2019. 2, 5, 6, 7

[39] Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 4

[40] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017. 2

[41] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 10–20. IEEE, 2018. 2

[42] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 2

[43] Erik Sandström, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Luc Van Gool, Martin R Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. *arXiv preprint arXiv:2405.16544*, 2024. 2, 5, 6, 7

[44] Nicolas Schischka, Hannah Schieber, Mert Asim Karaoglu, Melih Görgülü, Florian Grötzner, Alexander Ladikos, Daniel Roth, Nassir Navab, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic neural radiance fields. *arXiv e-prints*, pages arXiv–2309, 2023. 1, 2, 5, 6

[45] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2018. 2

[46] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2012. 5, 6, 7

[47] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2

[48] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 3, 5, 6, 7, 8

[49] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36:39033–39051, 2023. 1, 2

[50] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1991. 5

[51] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023. 2

[52] 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. 2

[53] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 3

[54] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[55] Wenhua Wu, Chenpeng Su, Siting Zhu, Tianchen Deng, Zhe Liu, and Hesheng Wang. Add-slam: Adaptive dynamic dense slam with gaussian splatting. *arXiv preprint arXiv:2505.19420*, 2025. 1, 2, 5, 6

[56] Wenhua Wu, Guangming Wang, Ting Deng, Sebastian Ægidiu, Stuart Shanks, Valerio Modugno, Dimitrios Kanoulas, and Hesheng Wang. Dvn-slam: Dynamic visual neural slam based on local-global encoding. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14564–14571. IEEE, 2025. 2

[57] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. Dg-slam: Robust dynamic gaussian splatting slam with hybrid pose optimization. *Advances in Neural Information Processing Systems*, 37:51577–51596, 2024. 1, 2, 5, 6

[58] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024. 2

[59] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2

[60] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1168–1174. IEEE, 2018. 2

[61] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2d feature representations by 3d-aware fine-tuning. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 4

[62] 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, 5, 6, 7, 1

[63] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2020. 2

[64] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023. 2

[65] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, et al. Fast-livo2: Fast, direct lidar-inertial-visual odometry. *IEEE Transactions on Robotics*, 2024. 5, 1, 2

[66] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11461–11471, 2025. 1, 2, 4, 5, 6, 7, 8

[67] Wancai Zheng, Linlin Ou, Jiajie He, Libo Zhou, Xinyi Yu, and Yan Wei. Up-slam: Adaptively structured gaussian slam with uncertainty prediction in dynamic environments. *arXiv preprint arXiv:2505.22335*, 2025. 1, 2, 5, 6

[68] Fangwei Zhong, Sheng Wang, Ziqi Zhang, and Yizhou Wang. Detect-slam: Making object detection and slam mutually beneficial. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1001–1010. IEEE, 2018. 2

[69] Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, et al. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66):eabm5954, 2022. 1

[70] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 6, 7

[71] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2024. 2

DROID-SLAM in the Wild

Supplementary Material

Sequence	Number of Frames	Length of Trajectory [m]
`Downtown 1`	1427	90.83
`Downtown 2`	2200	122.25
`Downtown 3`	1438	62.33
`Downtown 4`	1794	85.19
`Downtown 5`	2157	129.93
`Downtown 6`	1900	104.99
`Downtown 7`	1900	109.35

Table 7. **Overview of Our DROID-W Dataset.**

In the supplementary material, we provide additional details about the following:

- More information about the DROID-W dataset and downloaded YouTube videos (Sec. 6).
- Details about the Jacobian derivation of our uncertainty optimization (Sec. 7).
- Additional qualitative comparisons for uncertainty, point clouds, and ablation study (Sec. 8).

6. Dataset

Prior benchmarks on dynamic SLAM are limited to indoor environments, exhibiting simple object motions and lacking truly challenging real-world conditions. To enable evaluation in more complex and unconstrained settings, we introduce an outdoor dataset, DROID-W, and additionally download 6 challenging videos from YouTube.

DROID-W Dataset. The DROID-W dataset is captured using a Livox Mid-360 LiDAR rigidly mounted with an RGB camera. It comprises 7 outdoor sequences (`Downtown 1-7`) with RGB frames at a resolution of 1200×1600, ground-truth camera poses, and synchronized IMU and LiDAR measurements. The RGB stream is recorded at 20 FPS, while RTK provides ground-truth poses at 10 Hz. We use the estimated trajectories from FAST-LIVO2 [65] as ground truth for `Downtown 1-2` due to the absence of RTK measurements. As shown in Table 8, FAST-LIVO2 provides sufficiently accurate estimates to serve as reliable ground truth for these sequences. Detailed information, including trajectory lengths and frame numbers, is reported in Table 7. The DROID-W dataset features long camera trajectories, high scene dynamics, and partial over-exposure – characteristics commonly encountered in real-world scenarios. We believe this dataset will provide significant value to the community and support future research on robust in-the-wild SLAM.

YouTube Videos. The framerates of the YouTube videos vary substantially. Therefore, we report the FPS and sequence duration for each sequence in Table 9, which provides a more meaningful characterization of camera motion and scene dynamics. Camera intrinsics are estimated using MonST3R [62] from the first 20 frames of each video. The sequences contain a large number of dynamic objects of diverse categories, with many of them moving simultaneously, leading to highly dynamic scenes. They also exhibit challenging and cluttered conditions, including motion blur, strong view-dependent effects, and low dynamic range.

7. Uncertainty Optimization and Jacobians

Given the definition in Sec. 3.3 of the main paper, we obtain the following uncertainty energy function:

$$\begin{aligned}
\mathbf{u}'_i &= \log(\exp(\boldsymbol{\theta} \cdot \mathbf{F}_i) + 1), \\
\mathbf{E}_{\text{uncer}}(\mathbf{u}') &= \sum_{(i,j)\in\mathcal{E}} \mathbf{e}_{ij} + \gamma_{\text{prior}} \sum_i \log(\mathbf{u}'_i + 1.0), \\
&= \sum_{(i,j)\in\mathcal{E}} \frac{1 - \frac{\mathbf{F}_i \cdot \mathbf{F}_{ij}}{\|\mathbf{F}_i\|_2 \|\mathbf{F}_{ij}\|_2}}{\mathbf{u}'_i \cdot \mathbf{u}'_{ij}} + \gamma_{\text{prior}} \sum_i \log(\mathbf{u}'_i + 1.0).
\end{aligned} \tag{11}$$

Thus, we can derive the following Jacobians:

$$\begin{aligned}
\frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{u}'_i} &= -\frac{1 - \frac{\mathbf{F}_i \cdot \mathbf{F}_{ij}}{\|\mathbf{F}_i\|_2 \|\mathbf{F}_{ij}\|_2}}{(\mathbf{u}'_i)^2 \cdot \mathbf{u}'_{ij}} = -\frac{\mathbf{e}_{ij}}{\mathbf{u}'_i}, \\
\frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{u}'_j} &= -\frac{1 - \frac{\mathbf{F}_i \cdot \mathbf{F}_{ij}}{\|\mathbf{F}_i\|_2 \|\mathbf{F}_{ij}\|_2}}{\mathbf{u}'_i \cdot (\mathbf{u}'_{ij})^2} \cdot \frac{\partial \mathbf{u}'_{ij}}{\mathbf{u}'_j} = -\frac{\mathbf{e}_{ij}}{\mathbf{u}'_j} \cdot \boldsymbol{\alpha}_{ij}.
\end{aligned} \tag{12}$$

where $\boldsymbol{\alpha}_{ij} \in \mathcal{R}^{(\frac{H}{8} \times \frac{W}{8}) \times (\frac{H}{8} \times \frac{W}{8})}$ is the bilinear interpolation weight matrix whose non-zero elements are of size $\frac{H}{8} \times \frac{W}{8} \times 4$. The final Jacobians are defined as follows:

$$\begin{aligned}
\frac{\partial \mathbf{E}_{uncer}}{\mathbf{u}'_l} &= -\sum_{(l,m)\in\mathcal{E}} \frac{\mathbf{e}_{lm}}{\mathbf{u}'_l} - \sum_{(k,l)\in\mathcal{E}} \frac{\mathbf{e}_{kl}}{\mathbf{u}'_k} \cdot \boldsymbol{\alpha}_{kl} + \gamma_{\text{prior}} \cdot \frac{1}{\mathbf{u}'_l + 1.0}, \\
\frac{\partial \mathbf{u}'_l}{\partial \boldsymbol{\theta}} &= \frac{1}{1 + \exp(-\boldsymbol{\theta} \cdot \mathbf{F}_{\boldsymbol{l}})} \cdot \mathbf{F}_l, \\
\frac{\partial \mathbf{E}_{\text{uncer}}}{\partial \boldsymbol{\theta}} &= \sum_{l=0}^{N} \frac{\partial \mathbf{E}_{\text{uncer}}}{\partial \mathbf{u}'_l} \cdot \frac{\partial \mathbf{u}'_l}{\partial \boldsymbol{\theta}}.
\end{aligned} \tag{13}$$

Here, frame l serves as the reference frame in all edges (l, m) and as the target frame in all edges (k, l).

8. Additional Experiments

8.1. Uncertainty Estimation

Uncertainty Comparisons on YouTube Videos. We provide additional qualitative results on uncertainty estimation

Method	Inputs	Downtown 3	Downtown 4	Downtown 5	Downtown 6	Downtown 7	**Avg.**
FAST-LIVO2 [65]	RGB + IMU + LiDAR	0.06	0.06	0.09	0.09	0.06	0.071
DROID-W (Ours)	RGB	0.37	0.32	0.51	0.07	0.05	0.264

Table 8. **Tracking performance of FAST-LIVO2 [65] on the DROID-W dataset** (ATE RMSE ↓ [m]). FAST-LIVO2 is an efficient and accurate LiDAR-inertial-visual fusion system capable of delivering centimeter-level localization accuracy. Its performance on Downtown 3-7 demonstrates sufficient accuracy to serve as ground truth for Downtown 1-2.

Sequence	Time	FPS	Resolution
Elephant Herd	00:08	24	1280 × 720
Giraffe	00:09	24	1280 × 720
Taylor	01:13	30	1280 × 720
Tomyum 1	01:40	30	1280 × 720
Tomyum 2	01:40	30	1280 × 720
St. Moritz	30:00	50	1280 × 720
Tokyo Walking 1	00:50	60	1920 × 1080
Tokyo Walking 2	00:22	60	1920 × 1080
Tokyo Walking 3	00:40	60	1920 × 1080

Table 9. **Overview of Downloaded YouTube Videos.**

in Fig. 5. As shown in Fig. 5, WildGS-SLAM [66] always fails to construct the reliable Gaussian map, leading to inaccurate and noisy uncertainty predictions. In contrast, our method leverages frame-to-frame feature alignment, demonstrating significantly greater robustness in visually complex and truly in-the-wild environments.

Moreover, our approach effectively handles strong view-dependent effects such as reflections and shadows (e.g. reflections in Taylor 22 and Tokyo Walking 2). It is also highly sensitive to small dynamic objects, enabling precise uncertainty estimation even under challenging real-world conditions. Our method further exhibits strong robustness to severe motion blur and low dynamic range (e.g. Tomyum 1 and Tomyum 2), which are extremely difficult for conventional segmentation or detection approaches.

Overall, Fig. 5 highlights the accuracy and robustness of our uncertainty optimization in unconstrained in-the-wild settings, effectively delineating uncertain regions while maintaining high confidence in static areas.

Uncertainty Visualization for Consecutive Keyframes. We visualize the optimized uncertainties across consecutive keyframes in Fig. 6 and Fig. 7. Our method optimizes frame-wise uncertainty by exploiting multi-view feature similarity, allowing it to fully leverage static-scene information whenever available. As shown in Fig. 6, the system effectively utilizes the door region for camera tracking prior to keyframe 280. In addition, our uncertainty estimation integrates multi-view cues from frames connected through the frame graph, i.e. it captures multi-view inconsistency within a local window. Thus, our approach assigns the reasonable higher uncertainty to the door region of keyframes 280/281 before the door begins to move.

Method	ATE RMSE [cm]
w/o prior term	5.18
Full	**2.31**

Table 10. **Ablation Studies on Bonn RGB-D Dataset [38].**

We observe strong mirror-reflection effects in Fig. 7, with keyframe 283 exhibiting the largest appearance change. Accordingly, our method assigns the highest uncertainty to keyframe 283, while maintaining relatively low uncertainty for the remaining keyframes that show only minor appearance differences. This behavior demonstrates the effectiveness of our approach and the precision of the resulting uncertainty estimates.

8.2. Point Cloud Reconstruction

Static Reconstruction Comparisons. We present 3D reconstruction comparisons between DROID-SLAM [48] and our method in Fig. 8. In the top row, DROID-SLAM fails to recover consistent geometry – erroneously reconstructing a single corridor as two separate structures – and exhibits noticeable tracking drift in sequences with the presence of strong dynamic distractors. In contrast, our approach produces coherent geometric reconstructions and accurate pose estimates. The second row further highlights the robustness of our method: it reconstructs cleanly visible and highly accurate white lane markings on the asphalt road, even under challenging real-world conditions.

Static / Dynamic Reconstruction. We visualize the static and dynamic point clouds in Fig. 9 and Fig. 10, providing complementary perspectives from both top-down and interior viewpoints. The high geometric consistency between our static reconstruction and the static regions within the dynamic point clouds illustrates the precision of our uncertainty estimation. These results demonstrate that our method effectively suppresses dynamic or uncertain regions while preserving the underlying static scene structure.

8.3. Additional Ablation Study

Table 10 shows that the prior regularization term effectively avoid the trivial solution $\mathbf{u} \to \infty$. Without this prior, the system assigns uniformly large uncertainties to all pixels, resulting in results similar to the *w/o Uncertainty-aware BA* configuration reported in Table 6 of the main paper.

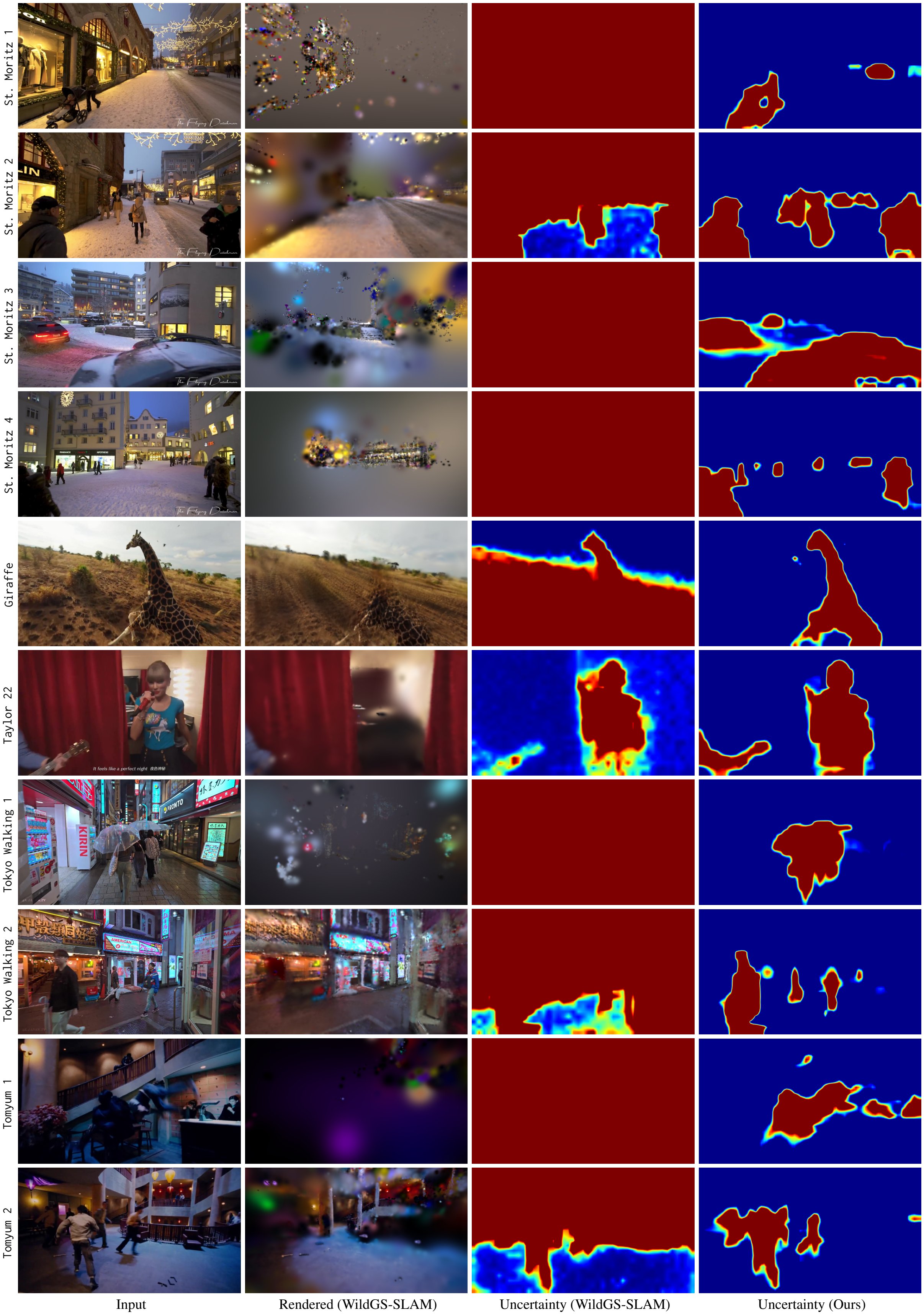

Figure 5. **Uncertainty Estimation.**

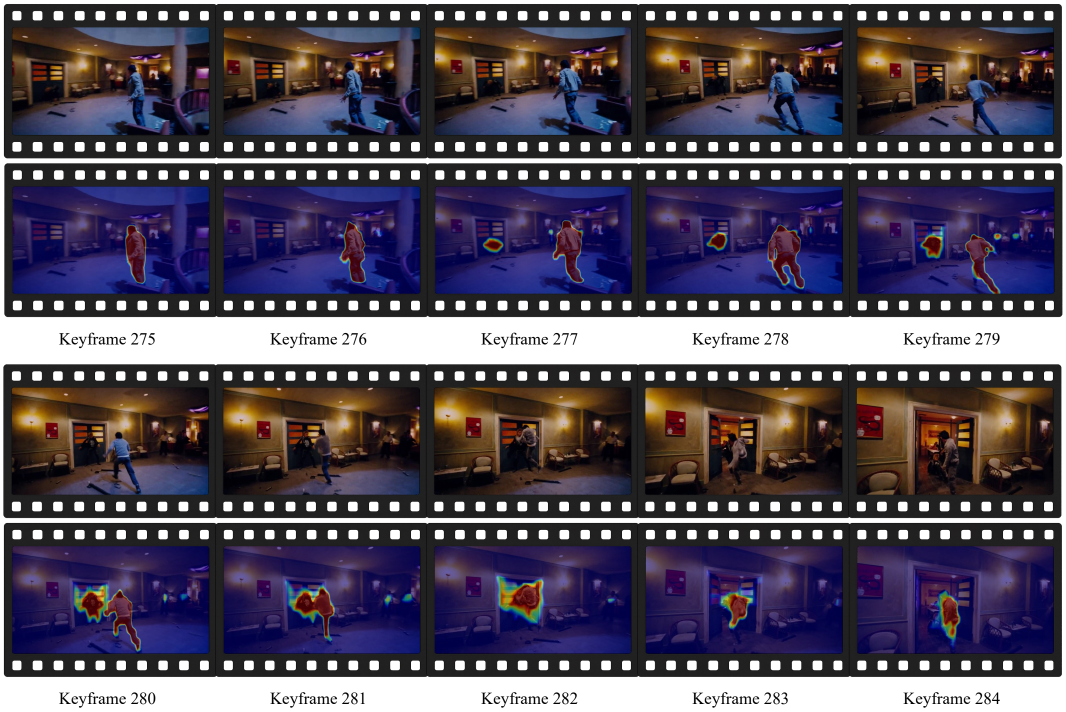

Figure 6. **Uncertainty Visualization for Consecutive Frames of YouTube `Tomyum 1`.** We observe that our method robustly handles scenarios in which an object transitions from a static state to dynamic motion, such as a door being pushed open by a person. Prior to the onset of motion, our approach leverages stable visual correspondences on the door to help tracking, since our uncertainty optimization is based on frame-to-frame feature alignment.

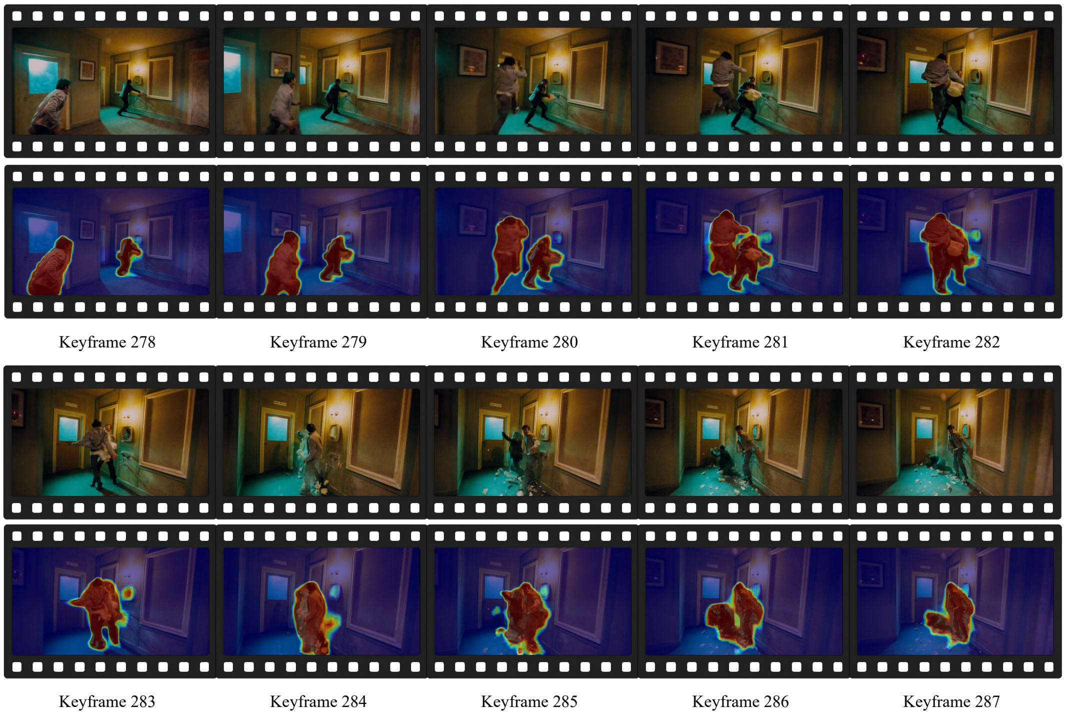

Figure 7. **Uncertainty Visualization for Consecutive Frames of YouTube `Tomyum 2`.** Our approach assigns high uncertainty to regions exhibiting strong view-dependent effects (e.g., the mirror).

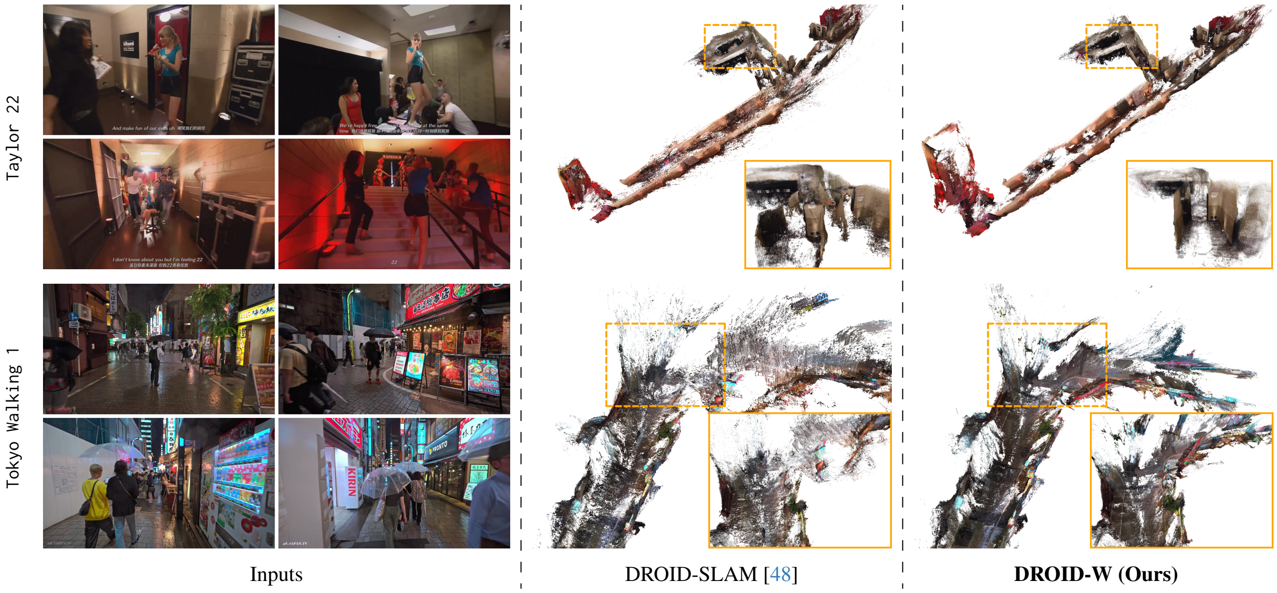

Figure 8. **3D Reconstruction Comparisons on YouTube Sequences.** We compare reconstructed static point clouds between DROID-SLAM [48] and our method. Our approach produces more accurate and consistent reconstructions across highly dynamic and visually challenging real-world sequences. For the `Taylor 22` scene, DROID-SLAM reconstructs two separate corridor structures due to inaccurate pose tracking, as highlighted in the boxed region, whereas our method produces a consistent geometric reconstruction. Moreover, the white lane marking on the asphalt road of `Tokyo Walking 1` is faint and fragmented in the point cloud reconstructed by DROID-SLAM, while in our reconstruction it remains cleanly visible, continuous, and highly accurate.

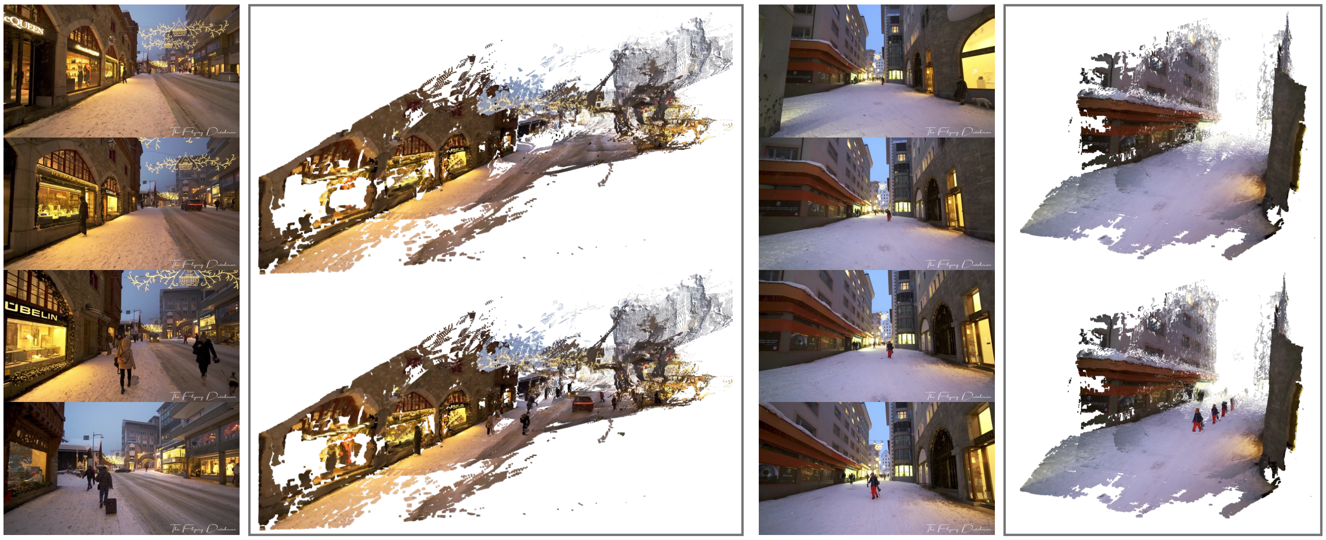

Figure 9. **Point Clouds Visualization from 4 Views.** *Top view*: reconstructed static scene from 4 input views shown in the figure. *Bottom view*: reconstructed dynamic point clouds. The comparisons between the dynamic and static point clouds further demonstrate the effectiveness of our uncertainty estimation. In both visualizations, the static point clouds remain highly consistent, indicating that our method reliably preserves static geometry while filtering dynamic regions.

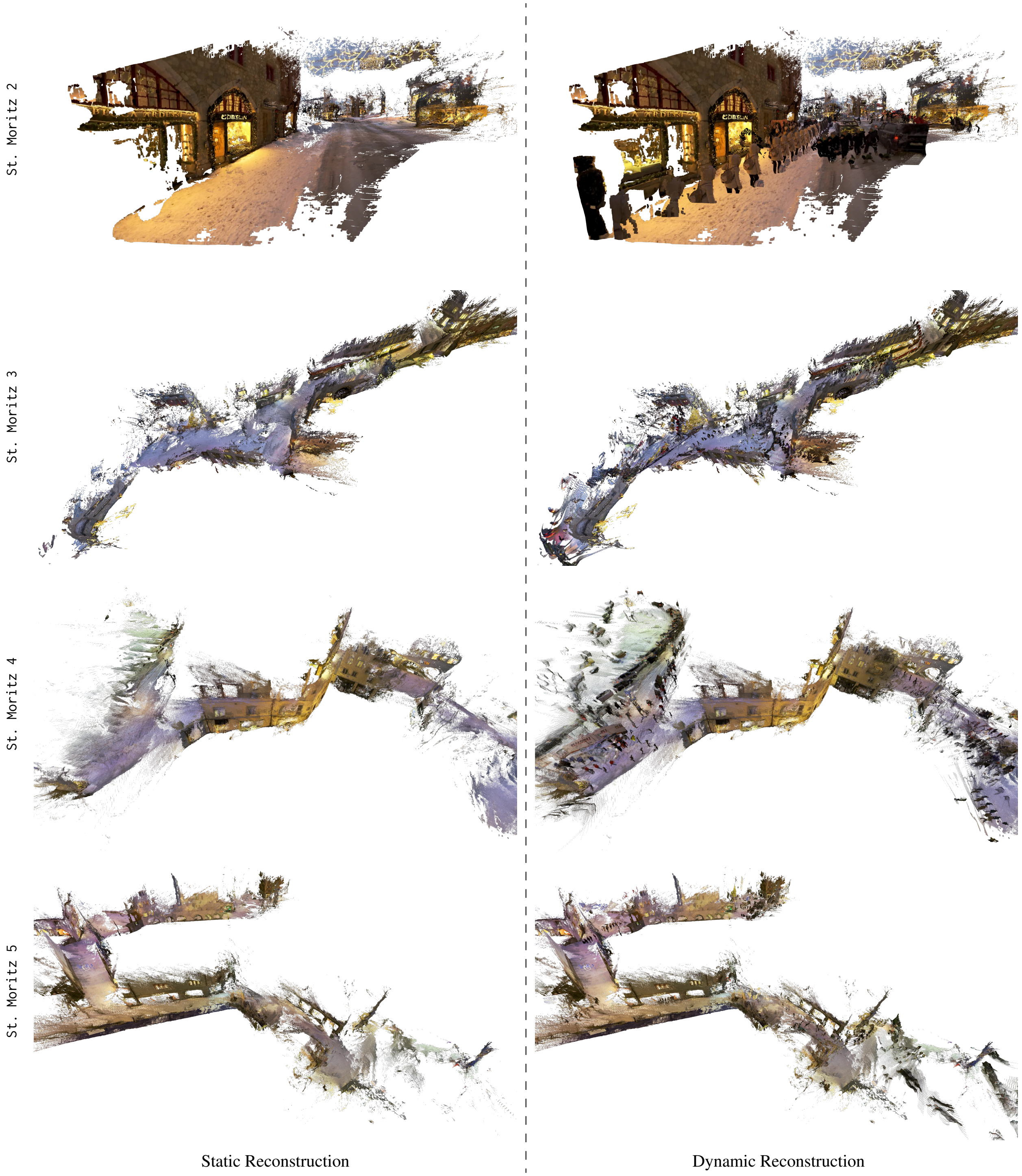

Figure 10. **Qualitative Results of Our Static and Dynamic Reconstruction.** We visualize globally aligned static reconstructions alongside dynamic point clouds across all keyframes. Notably, we apply the estimated per-frame dynamic uncertainty to filter out dynamic points. The left and right columns show the static and dynamic reconstructions, respectively. These comparisons highlight the accuracy of our uncertainty estimation, as our method effectively suppresses dynamic regions while preserving geometric fidelity in static areas.