

WildPose: A Unified Framework for Robust Pose Estimation in the Wild

Jianhao Zheng¹ Liyuan Zhu¹ Zihan Zhu² Iro Armeni¹
¹Stanford University ²ETH Zürich
wildpose.github.io

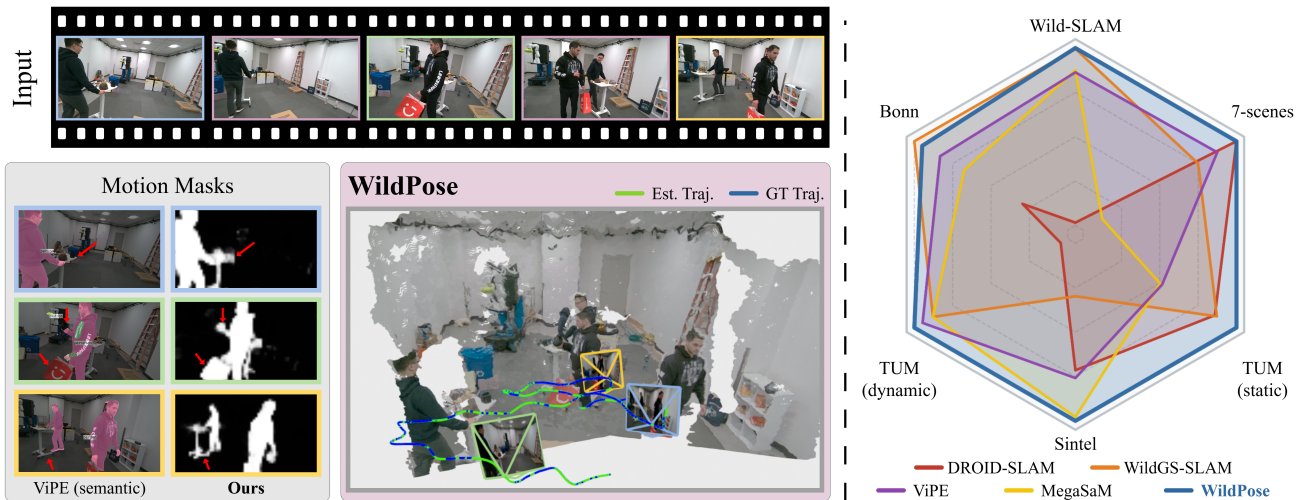


Figure 1. **WildPose**. *Left*: Given a calibrated video sequence captured in dynamic environments, our method effectively detects the moving distractors and accurately estimates the camera trajectory, whereas methods relying on semantic segmentation fail to identify all dynamic elements. *Right*: Pose estimation performance across multiple SLAM benchmarks, where our method achieves superior and stable performance on both dynamic and static datasets.

Abstract

Estimating camera pose in dynamic environments is a critical challenge, as most visual SLAM and SfM methods assume inputs from static environments. While recent dynamic-aware methods exist, they are often not unified: semantic-based approaches are brittle, per-sequence optimization methods fail on short sequences, and other learned models sometimes perform badly on static-only scenes. We present *WildPose*, a unified monocular pose estimation framework that is robust in dynamic environments while maintaining state-of-the-art performance on static and low-ego-motion datasets. Our key insight is to connect the two powerful paradigms in modern 3D vision: the rich perceptual frontend of feed-forward models and the end-to-end optimization of differentiable bundle adjustment (BA). We achieve this by enhancing the differentiable BA pipeline in two ways. First, we introduce a new 3D-aware update operator by integrating a frozen, pre-trained MAST3R feature backbone and training the operator’s subsequent layers on a diverse curriculum of static and dynamic data. Second, we propose a high-capacity motion mask detector that leverages rich, multi-level 3D-

aware features from the same frozen backbone. Extensive experiments show *WildPose* consistently outperforms prior methods across a wide variety of benchmarks, including dynamic (Wild-SLAM, Bonn), static (TUM, 7-Scenes), and low-ego-motion (Sintel) datasets.

1. Introduction

Estimating camera pose and 3D scene structure is a cornerstone of 3D vision, vital for applications like AR/VR and robotics. These tasks are commonly formulated as Structure from Motion (SfM) or, in the case of sequential input, Simultaneous Localization and Mapping (SLAM). While traditional geometric methods [12, 28–30, 60, 61] offered mature solutions, recent progress has been dominated by deep learning. This has led to two distinct and powerful research directions. On one hand, feed-forward SfM models [24, 44, 47] are trained to directly regress 3D structure and, in some cases, camera poses. On the other hand, differentiable optimization frameworks [41] learn an update operator to feed a differentiable bundle adjustment (BA) layer. Despite architectural differences, all these approaches largely share a critical vulnerability: they assume a static

world, which causes their performance to degrade significantly in dynamic scenes.

To handle dynamic environments, a common approach is to perform motion segmentation from 2D images. Several recent methods [7, 17, 52] utilize semantic information to extract motion masks. Nevertheless, such approaches are highly sensitive to the quality of semantic segmentation and often struggle to generalize across scenes with varied and unconstrained motion patterns. Two recent works that eliminate the reliance on semantic priors are WildGS-SLAM [58] and MegaSaM [26]. WildGS-SLAM [58] leverages image-rendering inconsistency from 3D Gaussian Splatting (3DGS) [22] to train on-the-fly a per-sequence specialized motion detector, but its effectiveness diminishes when input sequences lack sufficient temporal and spatial coverage. MegaSaM [26], on the other hand, trains a more general motion prediction network, offering a promising direction but suffers from a key limitation: it attempts to decode motion from the recurrent hidden state of its flow operator, which is designed to propagate flow information and not to explicitly segment motion. Hence, it lacks the capacity and feature richness required for a complex segmentation task. Furthermore, its reliance on a synthetic training corpus of limited diversity creates a domain gap, causing its performance to degrade on long, real-world trajectories. Critically, existing dynamic-scene methods such as the above often fail to maintain accuracy with solely static inputs. We argue that a robust SLAM system must excel in both static and dynamic scenarios, as real-world applications are inherently unpredictable and will inevitably encounter a mix of both.

We present *WildPose*, a unified and generalizable pose estimation framework capable of handling both dynamic and static environments, while remaining robust across short-term and long-term sequences. Our key insight is to fuse the two powerful paradigms in deep visual pose estimation and 3D reconstruction: the end-to-end optimization of differentiable BA frameworks [41] and the rich perceptual frontend of feed-forward models [31]. To this end, we adopt the differentiable BA pipeline of DROID-SLAM [41], which relies on a learnable update operator to refine poses and geometry. We enhance this pipeline in two fundamental ways.

First, we create a new 3D-aware update operator. We are the first to replace the standard, simple CNN encoder with a powerful, pre-trained 3D-aware feature extractor [24], providing rich geometric priors. We train this operator on a highly diverse, augmented dataset of both static and dynamic scenes with varied motion patterns, improving robustness to diverse, in-the-wild inputs. Second, we propose a new motion mask detector that also leverages these 3D-aware features. Unlike MegaSaM, which decodes from a weak recurrent state, our detector is a dedicated module receiving multi-level features from the MAST3R backbone. This high-capacity design enables more accurate dynamic region

detection and allows seamless integration as a robust weighting in the BA process.

Our full pipeline integrates these components with monocular depth priors for metric scale, along with loop closure and global BA for long-term consistency. Extensive experiments show WildPose consistently outperforms prior work across a wide variety of benchmarks. We further demonstrate that our improved pose accuracy and motion masks are beneficial for downstream tasks like consistent video depth estimation. In summary, our main contributions are:

- A unified monocular pose estimator, *WildPose*, that operates robustly in highly dynamic environments while maintaining state-of-the-art performance on static and short-term datasets.
- An enhanced update operator for differentiable BA, which leverages 3D-aware features and is trained on a new, diverse curriculum of static and dynamic data.
- A novel, high-capacity motion mask detector to identify dynamic regions that offers seamless integration into the BA pipeline.

2. Related Work

Feed-forward 3D Reconstruction. Feed-forward 3D reconstruction has emerged as a significant paradigm from classical optimization-based techniques, aiming to solve SfM in a single pass. These methods, pioneered by DUST3R [47] and MAST3R [23], directly predict 3D point maps and camera parameters from a set of unposed images in a single forward pass, followed by VGGT [44], which improved scalability and generalization. More recently, the focus has expanded to address the challenge of dynamic environments. MonST3R [55] directly finetunes DUST3R [47] to support robust prediction under dynamic scenes. CUT3R [45] proposes a recurrent state model that continuously updates its pose and point map prediction with each new observation. π^3 [48] and MapAnything [21] directly train multi-view transformers on large-scale static and dynamic datasets to generalize to in-the-wild data. While fast and robust, these models struggle with long sequences, precise camera pose estimation, and are not fully compatible with camera calibration.

Visual SLAM. Monocular Visual SLAM systems [28, 35, 41, 54, 62] estimate camera motion and scene geometry from RGB sequences. Classical approaches rely on sparse feature correspondences [3, 29] or direct photometric consistency [10, 11, 13], which often produce false matches in dynamic regions, leading to tracking failures. A prominent approach within this domain is the differentiable optimization framework proposed by Droid-SLAM [41]. This paradigm trains a deep neural network to predict intermediate variables, which are consumed by a dense, differentiable BA layer. While this design has demonstrated superior performance in static environments, its network is trained purely

on static data, limiting its robustness to dynamic inputs. Furthermore, its frontend consists of a simple CNN-based feature extractor trained from scratch. In contrast, our work leverages the powerful, pre-trained feature backbone from MAST3R [24], a successful feed-forward SfM model. Another method that leverages MAST3R is Mast3r-SLAM [31], which builds its system bottom-up from the MAST3R network’s direct 3D point map outputs. While it achieves robust real-time tracking, it operates as a fully training-free system. This contrasts with our approach, which integrates MAST3R’s internal features into a learnable, optimization-based pipeline. Moreover, this approach, like most in its category [27], is primarily designed for static scenes and struggles with significant dynamic elements.

Pose Estimation in Dynamic Scenes. To address the challenge of pose estimation in dynamic environments, many approaches first detect dynamic objects by leveraging geometric cues [6, 14, 32, 37, 49, 56, 57], or by predefining semantic priors to segment out moving objects, as in MASK-SLAM [19], Crowd-SLAM [39], and ViPE [17]. Others combine both strategies [1, 18, 25, 36, 51]. However, these semantic-based methods are inherently brittle: they are constrained by predefined categories, fail to detect novel dynamic objects, and are sensitive to segmentation quality. Other recent learning-based methods avoid these priors. BA-Track [5] learns to decompose the observed point motion into static and dynamic components. Since only the static component is induced by camera motion, it excludes the dynamic motion from bundle adjustment to eliminate distractions from moving objects. WildGS-SLAM [58] integrates dynamic object handling by training an uncertainty MLP online, supervised by image-rendering inconsistencies from 3D Gaussian splatting. This design, however, makes it reliant on high-quality mapping and wide view-point coverage. Furthermore, its pose estimation is tightly coupled with the computationally demanding mapping module and online mask optimization overhead, resulting in a slow pipeline. Alternatively, MegaSaM [26] learns a motion estimator from large-scale training data. However, it suffers from key limitations: its training on a synthetic corpus of limited diversity creates a domain gap, hindering generalization to long sequences. Architecturally, it decodes motion from the ConvGRU’s hidden state—a low-capacity representation designed for flow. In contrast, our method trains a separate, high-capacity detector leveraging 3D-aware features [24].

3. WildPose

Given a monocular RGB sequence $\{I_i\}_{i=1}^N$ with known camera intrinsics (f_x, f_y, c_x, c_y) , our goal is to estimate the per-frame camera poses $\{\omega_i\}_{i=1}^N \in SE(3)$. We first review the preliminaries of DROID-SLAM [41], then describe how we integrate the feed-forward front-end MAST3R [23] into a differentiable SLAM framework for end-to-end training.

We further introduce our learned motion mask detector and present a multi-stage training strategy for all components. Finally, we describe how the trained modules are combined for unified inference.

3.1. Preliminaries

DROID-SLAM [41] is characterized by two core components: a learnable update operator and a differentiable BA pipeline. The state of each keyframe i comprises the camera pose $\omega_i \in SE(3)$ and downsampled disparity map $\hat{d}_i \in \mathcal{R}^{\frac{H}{8} \times \frac{W}{8}}$. The optimization process is acted on a frame graph G , where edges $(i, j) \in G$ connect keyframes with significant observational overlap. For each edge in the graph, the update operator iteratively refines intermediate variables which are further utilized in the differentiable BA process to optimize the camera poses and geometry.

Update Operator. For each image pair (I_i, I_j) forming an edge in the pose graph, we estimate the optical flow by projecting and back-projecting pixels as $\tilde{\mathbf{f}}_{i,j} = \Pi_c \left(\omega_j^{-1} \omega_i \Pi_c^{-1} \left(p_i, \hat{d}_i \right) \right)$, where Π_c denotes the pinhole camera projection and p_i is a pixel location in the reference frame i . The update operator then iteratively refines the pose and disparity by adjusting intermediate variables guided by the estimated optical flow

$$(\hat{\mathbf{f}}_{i,j}^{t+1}, \hat{\mathbf{w}}_{i,j}^{t+1}, \eta^{t+1}, \hat{\mathbf{u}}^{t+1}) = F(I_i, I_j, \tilde{\mathbf{f}}_{i,j}^t, \mathbf{r}_{i,j}^t), \quad (1)$$

where $\hat{\mathbf{f}}_{i,j}^t \in \mathcal{R}^{\frac{H}{8} \times \frac{W}{8} \times 2}$ is the predicted optical flow between the two frames i and j , $\hat{\mathbf{w}}_{i,j}^t \in \mathcal{R}^{\frac{H}{8} \times \frac{W}{8} \times 2}$ is the confidence of the predicted optical flow, η^t is a damping factor used to stabilize the BA process, $\hat{\mathbf{u}}^t \in \mathcal{R}^{8 \times 8}$ is a mask for disparity upsampling, and $\mathbf{r}_{i,j}^t = \tilde{\mathbf{f}}_{i,j}^t - \hat{\mathbf{f}}_{i,j}^t$. The update operator F comprises a CNN feature encoder, a feature correlation module, and a ConvGRU-based recurrent network to iteratively refine the state variables. For more architecture details, please refer to the supplementary material.

Differentiable BA. The differentiable BA layer jointly optimizes all camera poses $\hat{\omega}$ and disparity maps $\hat{d}$ by minimizing the reprojection error over the frame graph G :

$$E(\hat{\omega}, \hat{d}) = \sum_{(i,j) \in G} \left\| \hat{\mathbf{f}}_{ij} - \tilde{\mathbf{f}}_{ij} \right\|_{\Sigma_{ij}}^2, \quad (2)$$

where $\Sigma_{ij}^{-1} = \text{diag}(\hat{\mathbf{w}}_{i,j})$. This objective penalizes the weighted residual between the predicted flow $\hat{\mathbf{f}}$ and the geometry-induced flow $\tilde{\mathbf{f}}$, weighted by the confidence $\hat{\mathbf{w}}$ from the update operator. This non-linear least-squares problem is solved by the Gauss-Newton algorithm. The cost function is linearized using local parameterization, and the resulting linear system for updates $(\Delta\hat{\omega}, \Delta\hat{d})$ is solved efficiently by applying the Schur complement. The entire process is fully differentiable, enabling end-to-end training by backpropagating gradients through the solver.

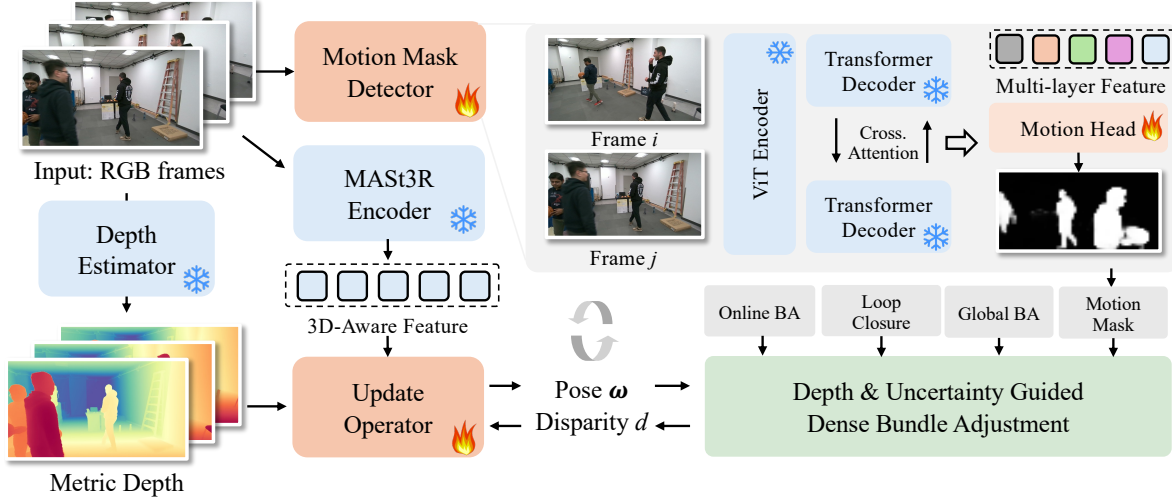


Figure 2. **System Overview.** WildPose robustly estimates the camera trajectory from a monocular RGB sequence. We leverage 3D-aware features from the frozen MAST3R encoder [24], which are fed into our update operator. Concurrently, a motion mask detector generates motion masks from the backbone’s multi-layer features. These outputs, combined with the metric depth prior [46], enable our Dense Bundle Adjustment layer to mitigate dynamic outliers and perform robust trajectory optimization.

3.2. Feed-forward Transformer Encoder

Recent advances in feed-forward models, such as MAST3R [24], have demonstrated that large-scale pre-training can produce powerful, 3D-aware image representations. In contrast, the update operator in DROID-SLAM [41] employs a relatively simple CNN encoder, trained from scratch, merely to extract auxiliary features for its recurrent ConvGRU. Therefore, we propose to replace this simple CNN with the pre-trained MAST3R backbone, integrating a feed-forward frontend with the robust optimization of a differentiable BA backend.

A direct integration is not possible, as the outputs of the MAST3R backbone are not compatible with the inputs expected by the DROID update operator. The MAST3R architecture consists of a ViT encoder [9] and two intertwined decoders (DecBlk¹, DecBlk²) that produce complex, multi-level correspondence tokens. An obvious design choice is to adapt the final multi-level tokens or the intermediate ViT features. However, we find that fusing the decoder tokens with the ConvGRU layer is computationally prohibitive for both training and inference.

Therefore, we opt to use the features from the ViT encoder. To make the encoder compatible with the ConvGRU, we introduce a lightweight adapter module. This module consists of two convolutional residual layers that take the patch-based ViT features as input. It processes these features to produce the local feature maps and global context information that the ConvGRU subsequently uses to update the flow field. We empirically selected MAST3R over other feed-forward models [44, 48] as we found its features yielded the best performance. We hypothesize this is because MAST3R’s

pre-training objectives on point map precision and 2D pixel correspondences are highly aligned with the update operator’s role in estimating the optical flow, providing “cleaner” and more relevant features than those from more versatile models.

3.3. Motion Mask Detector

The BA objective in Eq. (2) only holds on static scenes, where only ego-motion $\mathbf{f}_{ij}$ determines the full optical flow $\hat{\mathbf{f}}_{ij}$. This assumption is violated in dynamic environments. For a pixel p_i corresponding to a moving 3D point, its true optical flow $\tilde{\mathbf{f}}_{i,j}^*$ is a composite of camera motion and the object’s self motion:

$$\tilde{\mathbf{f}}_{i,j}^* = \Pi_c \left(\hat{\omega}_j^{-1} \hat{\omega}_i \Pi_c^{-1} \left(p_i, \hat{d}_i \right) + X_{i,j}(p_i) \right)$$

where $X_{i,j}(p_i)$ represents the object’s 3D displacement between frames I_i and I_j . Consequently, directly minimizing Eq. (2) forces the optimizer to incorrectly adjust $\hat{\omega}$ and $\hat{d}$ to compensate for the large residuals from dynamic pixels, leading to inaccurate state estimation. To mitigate this, we propose a deep motion detector that takes an image pair as input and predicts a motion map to identify and filter these dynamic regions.

Our motion detector, illustrated in Fig. 2, is designed to predict a downsampled motion map $\hat{M}_{i,j} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8}}$. This map is used to robustify the BA objective (Eq. (2)) by identifying and down-weighting residuals from dynamic pixels. To achieve this, we adapt the MAST3R backbone [24] by aggregating the multi-level correspondence tokens produced by its transformer decoder via a DPT-style fusion layer [34].

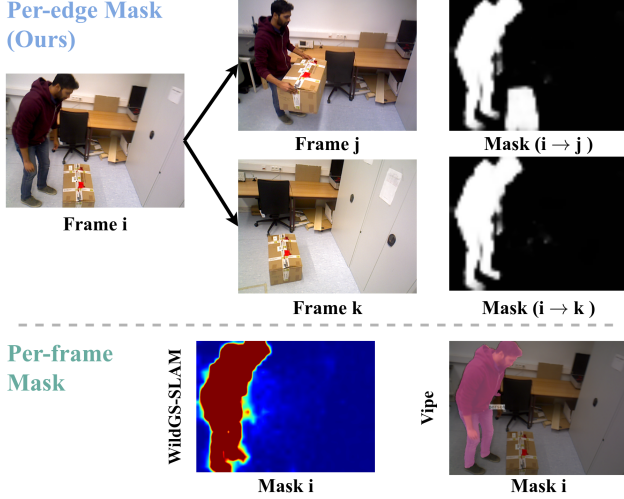


Figure 3. **Visualization of Motion Masks.** *Top:* Our per-edge masks (Frame $i \rightarrow j$ and $i \rightarrow k$) resolve temporal ambiguity by capturing motion relative to a second frame, enabling fine-grained detection of inconsistencies along each frame-graph edge. *Bottom:* Per-frame masks from prior methods (WildGS-SLAM [58] and Vipe [17]) are shown for comparison; these approaches produce frame-level predictions that are unable to identify transient motion.

The fused features are then passed to a final CNN-based motion head, which regresses the motion map $\hat{M}_{i,j}$ that identifies the dynamic regions on reference frame i .

Our edge-dependent mask design avoids the ambiguity of per-frame [58] or global dynamic object segmentation [17]. An object may move transiently, remaining static in other parts of the sequence, making a single-frame "dynamic" label unreliable. For instance, as shown in Fig. 3, a global-consistency-based method [58] fails to identify the moved box, as it is static in most other views. Our pairwise detector circumvents this. It only masks objects if they move between the input pair (I_i, I_j) , thereby correctly identifying true outliers for a specific edge in the BA graph. If the object is static within that pair, it is preserved as a valid constraint, improving robustness. This design resolves temporal ambiguity and integrates directly into the BA optimization.

3.4. Training

Our pipeline contains two learnable modules, the update operator and the motion detector, which we train using a multi-stage curriculum. Unlike prior work [35, 54, 58] that uses the pre-trained DROID-SLAM operator, our goal is to train a new operator with enhanced robustness to both static and dynamic scenes. This is followed by a separate stage to train the motion detector.

Training Datasets. We train our methods on a diverse set of synthetic datasets, including the static TartanAir V2 [42] and TartanGround [33], and the dynamic Dynamic Replica [20] and OmniWorld-Game [59]. While these provide various

trajectories, they underrepresent certain challenging camera motions. To address this, we augment our training pool with data from the Kubric simulator [15]. We generate both static and dynamic sequences focusing on three distinct motion patterns: linear translation, pure rotation, and target-locked motion, i.e., fixing the look-at point while the camera moves. The latter two are less prevalent in the other public datasets and enhance the dataset diversity.

Training Strategy. We first train the update operator end-to-end using only the static datasets. This stage trains the model to effectively learn pairwise flows induced purely by camera ego-motion. The loss function, similar to [41], is a weighted sum:

$$\mathcal{L}_1 = w_{cam}\mathcal{L}_{cam} + w_{flow}\mathcal{L}_{flow} + w_{res}\mathcal{L}_{res}, \quad (3)$$

where $\mathcal{L}_{cam}$ is the loss on the camera poses, $\mathcal{L}_{flow}$ compares the geometry-induced flow $\tilde{f}$ with the ground truth flow, and $\mathcal{L}_{res}$ is the residual loss between $\tilde{f}$ and the operator-predicted flow $\hat{f}$.

In the second stage, we finetune the operator on a mix of static and dynamic datasets. To train the model to handle dynamic content, we provide the annotated motion masks $\mathbf{M}$ and inject them directly into the BA process by modifying the weighting matrix from Eq. (2) to $\bar{\Sigma}_{ij}^{-1} = \text{diag}(\tilde{\mathbf{w}}_{i,j}\mathbf{M}_i)$. This stage trains the update operator to adapt to masked inputs and generalize to dynamic outliers.

Finally, we train the motion mask detector by coupling it with the frozen update operator and the differentiable BA layer. In this stage, we replace the ground truth motion masks with the predicted masks from our motion detector. We remove the flow loss and residual loss from the supervision to focus solely on the pose and mask quality:

$$\mathcal{L}_2 = w_{cam}\mathcal{L}_{cam} + w_{mask}\mathcal{L}_{BCE}, \quad (4)$$

where $\mathcal{L}_{BCE}$ is the binary cross-entropy loss between the predicted motion map and the annotated mask.

3.5. Inference

During inference, our system operates as an online visual odometry. We designate a new incoming frame as a keyframe if it has sufficient motion wrt. the latest keyframe. For each new keyframe I_i , we initialize its disparity $\hat{d}_i$ using the metric depth D_i from Moge2 [46] and form new edges with its co-visible keyframes in the graph G . For each new edge (i, j) created in the graph, we run our motion detector to predict the motion mask $M_{i,j}$. The poses and disparities are then optimized via BA by minimizing a composite objective similar to the training stage:

$$E(\hat{\omega}, \hat{d}) = \sum_{(i,j) \in G} \left\| \hat{\mathbf{f}}_{ij} - \tilde{\mathbf{f}}_{ij} \right\|_{\bar{\Sigma}_{ij}}^2 + \lambda \sum_{i \in G} \left\| \left(\hat{d}_i - 1/D_i \right) \right\|^2 \quad (5)$$

Method	ANYmal1	ANYmal2	Ball	Crowd	Person	Racket	Stones	Table1	Table2	Umbrella	Avg.
<i>Keyframe Poses</i>											
MASt3R-SLAM [31]	0.5	1.2	0.5	1.3	4.5	1.9	0.9	4.2	29.5	1.5	4.59
VGGT [44]	1.2	87.2	1.0	1.3	2.9	1.7	2.0	1.9	58.8	1.2	15.91
π^3 [48]	0.8	3.7	0.9	1.1	1.8	1.5	1.3	2.4	5.7	1.1	2.02
<i>Full Trajectory</i>											
DROID-SLAM [41]	0.6	4.7	1.2	2.3	0.6	1.5	3.4	48.0	95.6	3.8	16.17
WildGS-SLAM [58]	0.2	0.3	0.2	0.3	0.8	0.4	0.3	0.6	1.3	0.2	0.46
MegaSaM [26]	0.6	2.7	0.6	1.0	3.2	1.6	3.2	1.0	9.4	0.6	2.40
ViPE [17]	0.4	0.7	0.4	0.6	4.8	1.2	0.7	1.7	14.8	0.5	2.59
WildPose (Ours)	0.2	0.3	0.2	0.4	0.4	0.5	0.3	0.5	1.0	0.2	0.39

Table 1. **Tracking Performance on Wild-SLAM MoCap Dataset [58]** (ATE RMSE $\downarrow$ [cm]). Best results are highlighted as **first**, **second**, and **third**. All baseline methods were run using their publicly available code.

where the reprojection error is further weighted by the predicted motion mask, $\bar{\Sigma}_{ij}^{-1} = \text{diag}(\hat{\mathbf{w}}_i, j\hat{M}_i, j)$, with an additional disparity regularization term which encourages the estimated disparity to cohere with the metric depth prior. This optimization is performed in a sliding window manner, where old keyframes and edges are removed when new keyframes come in.

To mitigate scale and pose drift in longer sequences, we employ loop closure and global BA in addition to the local window optimization, similar to [35, 54, 58]. We detect loops by computing optical flow between current active keyframes and past keyframes; if the temporal interval between two frames exceeds a threshold τ and they have sufficient co-visibility, we create a new graph with these loop closure edges and perform BA on them. Global BA is performed on all keyframes to ensure global consistency, running periodically during the sequence with fewer iterations and as a final post-processing step. In the final global BA, we remove the second term in Eq. (5), which is the depth regularizer term. We empirically find that while the metric depth provides good initialization, its inherent noise can be detrimental when estimates are highly refined.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our approach on a wide range of datasets to assess its performance under various conditions.

- **Dynamic Scenes:** To demonstrate robustness to dynamic objects, we follow the benchmark from [58] and evaluate on Wild-SLAM MoCap Dataset [58], Bonn RGB-D Dynamic Dataset [32], and the dynamic sequences in TUM RGB-D Dataset [40].
- **Low-Motion and Short Trajectories:** To assess performance on challenging sequences with minimal egomotion, we evaluate on the MPI Sintel dataset [2]. This dataset contains 18 short sequences, each of which has 20-50 frames, with small camera motion.
- **Static Scenes:** Finally, to validate performance on standard static benchmarks, we evaluate on 7-Scenes

Dataset [38] and the nine static sequences in TUM RGB-D Dataset [40], following [31]. We also report the evaluation on the ScanNet Dataset [8] in the supplementary material.

Metrics. We follow the standard monocular SLAM evaluation pipeline [31, 54, 58]. The estimated trajectory is aligned to the ground truth using Sim(3) Umeyama alignment [43], after which we compute the RMSE of the absolute trajectory error (ATE).

Implementation details. The update operator is trained in a two-stage process using the Adam optimizer. The first pre-training stage uses only static datasets and runs for 200K iterations with a learning rate (LR) of 1.5×10^{-3} . We use a batch size of 3, where each sample consists of a 7-frame video sequence. The second stage is conducted jointly on both static and dynamic datasets for 100K iterations with a reduced LR of 1×10^{-4} . Subsequently, the motion mask detector is trained for 15K iterations with a batch size of 8 and a learning rate of 5×10^{-3} . The entire training curriculum takes approximately two weeks on 8 A100 GPUs. We employ gradient norm clipping with a threshold of 2.5 to ensure training stability. All training images are resized to 384×512 . During inference, we resize input images to a similar resolution while preserving the original aspect ratio.

Baselines. We compare WildPose against SOTA methods, categorized as dynamic, static, and feed-forward: WildGS-SLAM [58], ViPE [17], MegaSaM [26], DROID-SLAM [41], MASt3R-SLAM [31], VGGT [44], and π^3 [48]. Several baselines require specific evaluation protocols. For ViPE, we pre-identify and provide moving semantics as input, in accordance with its requirements. MASt3R-SLAM is evaluated only on keyframes, as it does not predict a full trajectory. Similarly, due to high GPU memory consumption on long sequences, VGGT and π^3 are also evaluated only on the keyframe poses identified by our method, except on Sintel, where the full trajectory is used due to the small number of frames. Noting that the keyframe-only evaluation is inherently less challenging than the full-trajectory evaluation employed for our method and other baselines.

Method	Balloon	Balloon2	Crowd	Crowd2	Person	Person2	Moving	Moving2	Avg.
<i>Keyframe Poses</i>									
MAS3R-SLAM [31]	2.8	3.1	4.4	6.6	2.6	1.9	7.5	2.8	3.96
VGGT [44]	3.3	3.5	2.0	2.1	6.6	27.1	1.4	4.0	6.26
π^3 [48]	1.5	1.7	2.7	2.9	3.6	7.3	1.4	2.3	2.92
<i>Full Trajectory</i>									
DROID-SLAM [41]	7.5	4.1	5.2	6.5	4.3	5.4	2.3	4.0	4.91
WildGS-SLAM [58]	2.8	2.4	1.5	2.3	3.1	2.7	1.6	2.2	2.31
MegaSaM [26]	3.7	2.6	1.6	7.2	4.1	4.0	1.4	3.4	3.51
ViPE [17]	3.3	2.7	2.6	3.9	3.4	3.8	1.4	2.3	2.93
WildPose (Ours)	2.6	2.4	1.4	1.9	4.0	2.6	1.3	2.7	2.36

Table 2. **Tracking Performance on Bonn RGB-D Dynamic Dataset [32]** (ATE RMSE ↓ [cm]).

Method	f3/ws	f3/wx	f3/wr	f3/whs	Avg.
<i>Keyframe Poses</i>					
MAS3R-SLAM [31]	0.7	1.7	3.9	3.7	2.51
VGGT [44]	0.7	1.5	3.6	2.0	1.96
π^3 [48]	0.9	2.4	3.8	2.6	2.44
<i>Full Trajectory</i>					
DROID-SLAM [41]	1.2	1.6	4.0	2.2	2.25
WildGS-SLAM [58]	0.4	1.3	3.3	1.6	1.63
MegaSaM [26]	0.6	1.5	2.6	1.8	1.63
ViPE [17]	0.5	1.3	3.0	1.6	1.58
WildPose (Ours)	0.6	1.3	2.7	1.7	1.57

Table 3. **Tracking Performance on TUM RGB-D (dynamic) Dataset [40]** (ATE RMSE ↓ [cm]).

4.2. Pose Estimation

Wild-SLAM Mocap. As shown in Table 1, our method achieves the best performance across all the sequences of Wild-SLAM Mocap Dataset [58]. While ViPE [17] performs comparably on most sequences, its error increases significantly on the Table1 and Table2 sequences. These scenes contain a human pushing a table while navigating, a challenging scenario for ViPE as tables are typically treated as static and thus not identified as dynamic by its semantic segmentation, leading to tracking errors. MegaSaM [26] and WildGS-SLAM [58] also detect motions without semantic priors. Our method still surpasses them, which we attribute to our use of 3D-aware features [24] and carefully designed training curriculum. Although π^3 is evaluated only on keyframes, its performance is nonetheless lower than ours. This suggests that while feed-forward models excel at dense pointmap predictions, their end-to-end pose estimation accuracy still lag behind optimization-based methods. Lacking mechanisms to handle dynamic content, other static methods performs dramatically worse on the dynamic dataset.

Bonn and TUM Dynamic. Tracking results on the Bonn RGB-D [32] and TUM Dynamic dataset [40], presented in Table 2 and Table 3, demonstrate superior or competitive performance of our method (second-best on Bonn, best on TUM). The Bonn dataset features non-trivial illumination changes across frames. WildGS-SLAM outperforms our approach on this dataset, likely because its explicit handling of illumination changes. WildPose outperforms all other

Method	Sintel [2] low-motion	TUM RGB-D [40] static	7-scenes [38] static
<i>Keyframe Poses</i>			
MAS3R-SLAM [31]	-	0.030	0.047
VGGT [44]	0.072 [†]	0.042	0.058
π^3 [48]	0.047 [†]	0.038	0.052
<i>Full Trajectory</i>			
DROID-SLAM [41]	0.030	0.038	0.049
WildGS-SLAM [58]	0.049	0.037	0.051
MegaSaM [26]	0.018	0.066	0.056
ViPE [17]	0.028	0.065	0.050
WildPose (Ours)	0.017*	0.027	0.049

Table 4. **Tracking Performance on Low-motion and Static benchmarks.** Reported as ATE RMSE ↓ in meters, except for Sintel where trajectories are scale-normalized following MegaSaM [26]. † indicates evaluation on all the frames.

baselines by a clear margin.

Sintel. We report tracking performance on Sintel [2] in Table 4. As these sequences are short (<50 frames), there are no memory constraints on VGGT [44] and π^3 [48], allowing evaluation on all frames. Notably, WildGS-SLAM, which performs competitively on the long dynamic sequences, struggles here, exhibiting higher error than even some static methods. This is because the short sequences are insufficient to build a high-quality 3DGS map for training its uncertainty MLP effectively.

Static Benchmarks. We report the performance on static datasets in Table 4. Our method achieves the best performance among all full-trajectory pose estimators on both TUM [40] and 7-Scenes [38]. This highlights the key advantage of WildPose: while MegaSaM [26] and ViPE [17] are robust on some dynamic benchmarks, their performance degrades significantly on static datasets, unlike our method which excels in both.

Qualitative Results. We provide qualitative trajectory comparisons in Fig. 4, visualizing 2D projections from three representative sequences, one each for dynamic, low-ego-motion, and static. In all examples, our estimated trajectory demonstrates the closest alignment to the ground truth.

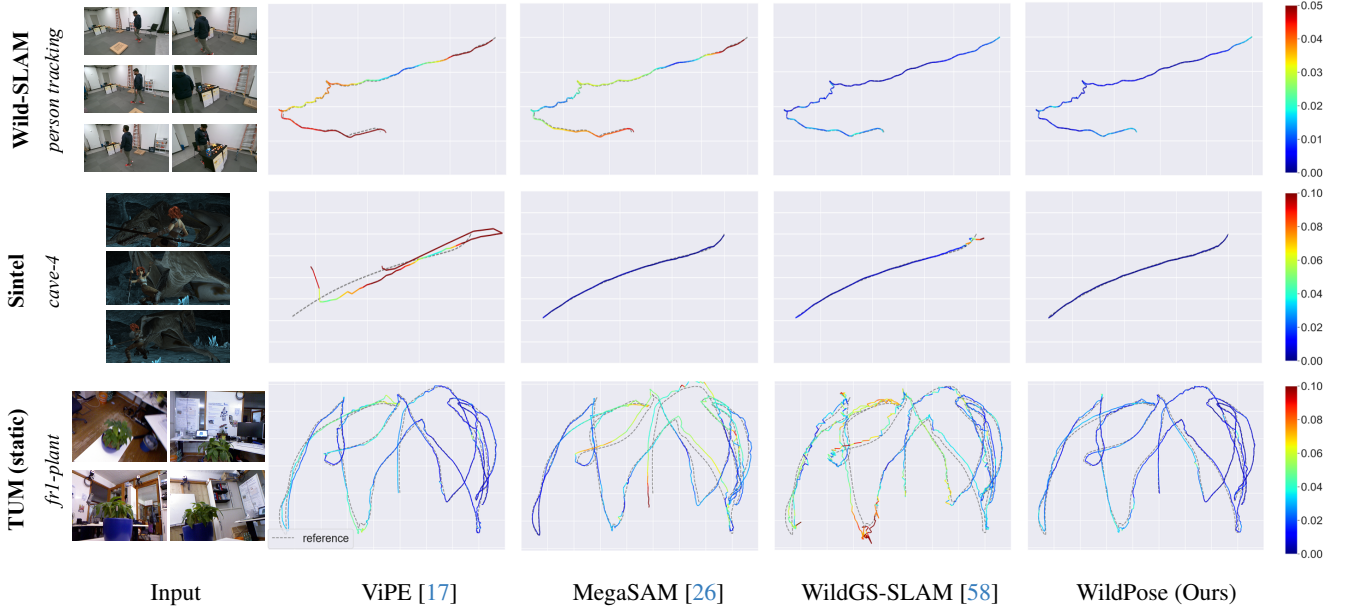


Figure 4. **Visualization of Camera Trajectories.** The estimated trajectory is colored by the translation error (ATE).

Method	Abs.Rel. ↓	Log-RMSE ↓	$\delta_{1.25}$ ↑
DA-v2 [53]	0.16	0.24	91.1
PPD [50]	0.15	0.24	94.6
DepthCrafter [16]	0.19	0.26	86.8
Video-Depth-Anything [4]	0.14	0.23	95.7
MegaSaM [26]	0.13	0.23	94.5
WildPose (Ours)	0.12	0.22	96.3

Table 5. **Comparison on Long-video Depth Estimation on Bonn RGB-D Dynamic [32] Dataset.**

4.3. Downstream Application: Depth Estimation

We demonstrate the benefit of our high-quality poses on video depth estimation. We adopt the depth optimization pipeline from MegaSaM [26], but substitute their poses with our own and incorporate our predicted motion masks as additional optimization weights. We evaluate on all eight sequences of the Bonn Dynamic Dataset [32], each of which is temporally subsampled by a stride of 10. The results in Table 5 further showcase our state-of-the-art performance on depth estimation in long sequences, outperforming all baselines on all metrics. This confirms that depth estimation benefits from more accurate pose estimation by WildPose.

4.4. Ablation Study

We validate our primary design choices in Table 6, where we ablate three key components: (1) the second-stage finetuning of the update operator with mixed static and dynamic datasets (*Mix. Ft.*), and (2) the application of our motion mask detector to down-weight dynamic regions during BA

<i>Mix. Ft.</i>	<i>Mot. Mask</i>	<i>GBA Dep. Off</i>	Wild-SLAM [58]	Bonn [32]	TUM [40]
✗	✗	✓	2.16	5.58	2.10
✓	✗	✓	0.76	2.61	1.58
✗	✓	✓	0.61	2.55	1.66
✓	✓	✗	2.34	2.60	2.13
✓	✓	✓	0.39	2.50	1.54

Table 6. **WildPose Ablation Study** (ATE RMSE ↓ [cm]). We report the average tracking error for each dataset. *Mix. Ft.* denotes finetuning the update operator with the mix of static and dynamic datasets, *Mot. Mask* denotes the motion mask detector, and *GBA Dep. Off* denotes removing the depth regularization term during the final global BA.

(*Mot. Mask*), and (3) the removal of the depth regularizer in the final global BA (*GBA Dep. off*). Our full model, integrating all components, achieves the best performance, validating our complete system design.

5. Conclusion

We presented WildPose, a novel pose estimation framework that achieves robust performance on both static and dynamic scenes. Our method enhances the differentiable BA pipeline by integrating a 3D-aware feature backbone from MAST3R, which powers both a new update operator and a high-capacity motion mask detector. This hybrid design proves superior to prior work, setting a new state-of-the-art on a wide array of static and dynamic datasets and benefiting downstream tasks like depth estimation. We discuss our limitations and future work in the supplementary material.

Acknowledgment. This work is supported by the Center for Integrated Facility Engineering (CIFE) and the Stanford

Robotics Center (SRC). We also thank Stanford’s Marlowe computing clusters and Stanford Sherlock clusters for providing GPU computing for model training and evaluation.

References

- [1] Berta Bescos, José M. Fàcil, Javier Civera, and José Neira. DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes. *RAL*, 2018. 3
- [2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 6, 7
- [3] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021. 2
- [4] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22831–22840, 2025. 8
- [5] Weirong Chen, Ganlin Zhang, Felix Wimbauer, Rui Wang, Nikita Araslanov, Andrea Vedaldi, and Daniel Cremers. Back on track: Bundle adjustment for dynamic scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4951–4960, 2025. 3
- [6] Jiyu Cheng, Yuxiang Sun, and Max Q-H Meng. Improving monocular visual slam in dynamic environments: an optical-flow-based approach. *Advanced Robotics*, 2019. 3
- [7] Shuhong Cheng, Changhe Sun, Shijun Zhang, and Dianfan Zhang. Sg-slam: A real-time rgb-d visual slam toward dynamic scenes with semantic and geometric information. *IEEE Transactions on Instrumentation and Measurement*, 2022. 2
- [8] 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. 6
- [9] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4
- [10] 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
- [11] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE TPAMI*, 2017. 2
- [12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2017. 1
- [13] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014. 2
- [14] Lily Goli, Sara Sabour, Mark Matthews, Marcus A Brubaker, Dmitry Lagun, Alec Jacobson, David J Fleet, Saurabh Saxena, and Andrea Tagliasacchi. Romo: Robust motion segmentation improves structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6155–6164, 2025. 3
- [15] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanaprasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 5
- [16] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. 8
- [17] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. 2, 3, 5, 6, 7, 8
- [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*, 2024. 3
- [19] Masaya Kaneko, Kazuya Iwami, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation. In *CVPR Workshops*, 2018. 3
- [20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. 5
- [21] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kotschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *arXiv:2509.13414*, 2025. 2
- [22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 2
- [23] 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, 3
- [24] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 1, 2, 3, 4, 7
- [25] Mingrui Li, Zhetao Guo, Tianchen Deng, Yiming Zhou, Yuxiang Ren, and Hongyu Wang. Ddn-slam: Real time dense dynamic neural implicit slam. *IEEE Robotics and Automation Letters*, 2025. 3
- [26] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megagam: Accurate, fast, and robust structure and motion from casual dynamic videos. *CVPR*, 2025. 2, 3, 6, 7, 8

- [27] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *NeurIPS*, 2025. 3
- [28] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *CVPR*, 2024. 1, 2
- [29] 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. 2
- [30] 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. 1
- [31] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *CVPR*, 2025. 2, 3, 6, 7
- [32] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *iros*, 2019. 3, 6, 7, 8
- [33] Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartan-ground: A large-scale dataset for ground robot perception and navigation. *arXiv preprint arXiv:2505.10696*, 2025. 5
- [34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 4
- [35] 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
- [36] Nicolas Schischka, Hannah Schieber, Mert Asim Karaoglu, Melih Gorgulu, Florian Grötzner, Alexander Ladikos, Nasir Navab, Daniel Roth, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic neural radiance fields. *IEEE Robotics and Automation Letters*, 2024. 3
- [37] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. 2018. 3
- [38] 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. 6, 7
- [39] João Carlos Virgolino Soares, Marcelo Gattass, and Marco Antonio Meggiolaro. Crowd-slam: visual slam towards crowded environments using object detection. *Journal of Intelligent & Robotic Systems*, 2021. 3
- [40] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 6, 7, 8
- [41] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6, 7
- [42] The AirLab. TartanAir-V2 Dataset. <https://tartanair.org>, 2022. Accessed: 2025-10-28. 5
- [43] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI*, 1991. 6
- [44] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025. 1, 2, 4, 6, 7
- [45] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 2
- [46] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 4, 5
- [47] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2
- [48] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3 : Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 2, 4, 6, 7
- [49] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6112–6122, 2021. 3
- [50] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *arXiv preprint arXiv:2510.07316*, 2025. 8
- [51] 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. 3
- [52] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. DG-SLAM: Robust Dynamic Gaussian Splatting SLAM with Hybrid Pose Optimization. In *NeurIPS*, 2024. 2
- [53] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, pages 10371–10381, 2024. 8
- [54] Ganlin Zhang, Erik Sandström, Youmin Zhang, Manthan Patel, Luc Van Gool, and Martin R Oswald. Glorie-slam: Globally optimized rgb-only implicit encoding point cloud slam. *arXiv preprint arXiv:2403.19549*, 2024. 2, 5, 6
- [55] 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
- [56] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and

- motion from casual videos. In *ECCV*, pages 20–37. Springer, 2022. [3](#)
- [57] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*, pages 523–542. Springer, 2022. [3](#)
- [58] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *CVPR*, pages 11461–11471, 2025. [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [59] 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. *arXiv preprint arXiv:2509.12201*, 2025. [5](#)
- [60] Liyuan Zhu, Yue Li, Erik Sandström, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. In *3DV*, 2025. [1](#)
- [61] 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 *CVPR*, 2022. [1](#)
- [62] 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. 2024. [2](#)