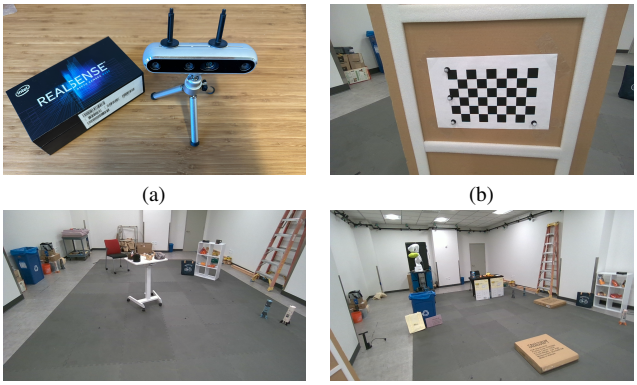
WildGS-SLAM: Monocular Gaussian Splatting SLAM in Dynamic Environments

Supplementary Material



(a) (b) (c) (d)

Figure 7. (a) Intel RealSense D455 camera [15]. (b) Calibration board used to align camera reference frame with OptiTrack's rigid body frame. (c) Static scene 1. (d) Static scene 2.

Abstract

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

1. *More information about the Wild-SLAM dataset (Sec. 6).*
2. *Implementation details of WildGS-SLAM and baseline methods (Sec. 7).*
3. *Additional results and ablations (Sec. 8).*

6. Wild-SLAM Dataset

Wild-SLAM MoCap Dataset. This dataset comprises a total of 10 sequences of RGB-D frames featuring various moving objects as distractors, specifically designed for dynamic SLAM benchmarking. Although WildGS-SLAM works with monocular inputs, aligned depth images are included to support the evaluation of other RGB-D baselines or future research. The RGB-D frames were captured using an Intel RealSense D455 camera (Fig. 7a) at a resolution of 720×1280 and a frame rate of 30 fps. All image sequences in this dataset were recorded with a fixed exposure time. The dataset includes two distinct static environment layouts: 2 sequences were captured in one static scene (Fig. 7c), and the remaining 8 sequences were recorded in the other (Fig. 7d). A summary of each sequence is provided in Table 6, while all sequences are presented in the video. The room used for dataset collection was equipped with an OptiTrack motion capture (MoCap) system [34], consisting of 32 OptiTrack PrimeX-13 cameras, to provide the ground truth camera poses. The OptiTrack system operates at 120 fps.

Inspired by [65], to synchronize the OptiTrack system with the Intel RealSense D455 camera, we positioned the RealSense camera and one of the OptiTrack cameras to observe an iPhone. By switching the iPhone flashlight on and off, we identified the corresponding timestamps in the images captured by both devices that reflected the flashlight's state change. This allowed us to get the timestamp offset between the two devices. To improve synchronization accuracy, the frame rate of the D455 was increased to 60 fps. We switched the flashlight on and off before and after each sequence recording, obtaining four timestamp offset values. The timestamp offset per recording was calculated as their average. Across all sequences, the average standard deviation among the four timestamp offset values was 5.25 ms, while the time interval between consecutive frames in the captured sequence is 33.33 ms, highlighting the precision of the synchronization.

To track the pose of the D455 camera using the MoCap system, we attached four reflective markers to the camera, defining a rigid body. We then performed a calibration procedure using a calibration board (Fig. 7b) to determine the relative transformation between the MoCap coordinate system and the camera coordinate system for this rigid body. Four reflective markers were carefully placed on the calibration board, enabling the MoCap system to track their 3D positions. These positions were then utilized to compute the locations of the grid corners on the board. Meanwhile, the corresponding 2D pixel coordinates of these grid corners in the camera frame were identified using the method described in [6]. Using this information, the camera poses in the MoCap coordinate system were determined, allowing us to compute the transformation between the rigid body and the camera frame. The calibration process was repeated 19 times, with the camera and the calibration board positioned in different poses for each trial, resulting in 19 transformation matrices. The final transformation between the rigid body and the camera frame was computed by averaging the results across all trials. Specifically, the rotation component was averaged using the chordal L_2 method [10]. The average deviation between an individual estimated transformation matrix and the final averaged transformation is 0.44° for rotation and 0.24 cm for translation.

Wild-SLAM iPhone Dataset. To further assess performance in more unconstrained, real-world scenarios, we captured 7 sequences using an iPhone 14 Pro. These sequences comprise 4 outdoor and 3 indoor scenes, showcasing a variety of daily-life activities such as strolling along streets, shopping, navigating a parking garage, and exploring an art museum. Each sequence provides RGB images at a resolution of 1920 $\times$ 1280, accompanied by LiDAR depth images at 256 $\times$ 192

Sequence Name	Distractors	Static Environment	Number of Frames	Length of Trajectory [m]
ANYmal1	ANYmal Robot	Scene 1	651	7.274
ANYmal2	ANYmal Robot	Scene 1	1210	11.567
Ball	Human, Basketball	Scene 2	931	11.759
Crowd	Human, Basketball, Bag	Scene 2	1268	14.189
Person	Human	Scene 2	986	10.354
Racket	Human, Racket	Scene 2	962	12.421
Stones	Human, Table, Bag, Gripper, Stone	Scene 2	962	12.421
Table1	Human, Table, Gripper, Stone	Scene 2	561	6.592
Table2	Human, Table, Gripper, Stone	Scene 2	1029	11.184
Umbrella	Human, Umbrella	Scene 2	458	4.499

Table 6. **Overview of our WildGS-SLAM MoCap Dataset.**

resolution. While WildGS-SLAM only requires monocular inputs, the inclusion of LiDAR data facilitates the evaluation of RGB-D baselines and future research. All sequences are showcased in the supplementary video.

Discussion. Both datasets capture humans performing activities. The *Wild-SLAM MoCap Dataset* was recorded in controlled environments, with explicit consent obtained from all participants for publishing, presenting, and sharing the data with the research community. In contrast, the *Wild-SLAM iPhone Dataset* was captured in more unconstrained settings, where we had less control over the presence of bystanders in the scene. While consent was obtained from the primary individuals featured, additional people may occasionally appear in the background. In most cases, these individuals are positioned too far from the camera to be identifiable (occupying very few pixels). Additionally, in the Parking sequence, certain car license plates are visible. To ensure privacy, all sensitive regions, including faces and license plates, have been masked in the data. It is important to note that recordings were conducted in locations where capturing people in public spaces is legally permitted, provided the footage does not target individuals in a way that could be considered intrusive or harassing.

7. Implementation Details

7.1. WildGS-SLAM

Two-Stage Initialization. We use the first 12 keyframes to run the DBA layer for tracking initialization. However, the uncertainty MLP $\mathcal{P}$ has not yet been trained to identify uncertain regions. Hence, we deactivate the uncertainty weight β in Eq. (5) during the first stage of initialization to obtain coarse camera poses. These initial poses are used for map initialization and training of $\mathcal{P}$. Subsequently, we perform a reduced number of iterations in the DBA layer, with uncertainty weighting activated, to refine the coarse keyframe camera poses from the first stage.

Frame Graph Management. We manage the frame graph as in [47] but enforce the insertion of a new keyframe every 8 frames, independent of the criterion in [47] (average optical flow to the last keyframe larger than a threshold).

Disparity Regularization Mask M. For each newly inserted keyframe i, we project each of its connected keyframes j in the frame graph, i.e., $(i,j) \in E$, onto i using the metric depth $\tilde{D}_j$ and calculate the multi-view depth consistency count as:

$$n_i(u,v) = \sum_{j|(i,j)\in E} \mathbb{1}\Bigg(\frac{|\tilde{D}_i(u,v) - \tilde{D}_{j\to i}(u',v')|}{\tilde{D}_{j\to i}(u',v')} < \epsilon \\ \wedge \cos\left(F_i\left(u,v\right), F_j\left(u',v'\right)\right) > \gamma \Bigg) \tag{8}$$

where $\mathbb{1}(\cdot)$ is the indicator function, (u',v') is the pixel coordinate in jth frame that falls to (u,v) when re-projected to frame i using $\tilde{D}_j$, $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$, $\tilde{D}_{j\to i}(u',v')$ is the projected depth from point (u',v') to frame i, and ϵ is the relative depth threshold. The second condition is to filter out incorrect correspondences that have lower than a threshold γ DINO feature cosine similarity. The depth mask $M_i(u,v)$ is set to 0 if (u,v) has more than one valid correspondence in neighboring frames and $n_i(u,v)$ is less than a threshold.

Final Global BA. After processing all the input frames, we incorporate a final global Bundle Adjustment (BA) module, similar to DROID-SLAM [47], to refine the keyframe poses. The frame graph construction follows the same approach as DROID-SLAM [47]. For the DBA objective during tracking, we retain only the first term of Eq. (5), omitting the disparity regularization term, as sufficient multiview information is already available, and the uncertainty map has converged to a stable state. We include an ablation study in Sec. 8.

Final Map Refinement. After the final global BA, we perform a final refinement of the map using all keyframes, following the same strategy as Splat-SLAM [39] and MonoGS [30]. In the final refinement, we fix the keyframe poses and optimize both the uncertainty MLP and 3D Gaussian map using Eq. (6).

Obtaining Non-keyframe Pose. After completing the final global BA and map refinement, we conduct a motion-only bundle adjustment to estimate non-keyframe poses, similar to the approach in DROID-SLAM [47]. During this opti-

Method	Input Type	Dynamic	Open Source	Prior Free	Scene Representation
Classic SLAM methods					
DSO [7]	RGB	✗	✓	✓	Sparse Point Cloud
ORB-SLAM2 [32]	RGB	✗	✓	✓	Sparse Point Cloud
DROID-SLAM [47]	RGB	✗	✓	✓	-
Classic SLAM methods with dynamic environment handling					
Refusion [36]	RGB-D	✓	✓	✓	TSDF
DynaSLAM (RGB) [2]	RGB	✓	✓	✗(S)	Sparse Point Cloud
DynaSLAM (N+G) [2]	RGB-D	✓	✓	✗(S)	Sparse Point Cloud
Static neural implicit and 3DGS SLAM methods					
NICE-SLAM [68]	RGB-D	✗	✓	✓	Neural Implicit
MonoGS [30]	RGB/RGB-D	✗	✓	✓	3D Gaussian Splatting
SplatSLAM [39]	RGB	✗	✓	✓	3D Gaussian Splatting
Concurrent neural implicit and 3DGS SLAM methods for dynamic scenes					
DG-SLAM [54]	RGB-D	✓	✗	✗(S)	3D Gaussian Splatting
RoDyn-SLAM [16]	RGB-D	✓	✗	✗(S)	Neural Implicit
DDN-SLAM [27]	RGB/RGB-D	✓	✗	✗(O)	Neural Implicit
DynaMoN (MS) [40]	RGB	✓	✓	✓	Neural Implicit
DynaMoN (MS & SS) [40]	RGB	✓	✓	✗(S)	Neural Implicit
Recent feed-forward methods					
MonST3R [61]	RGB	✓	✓	✓	Dense Point cloud
WildGS-SLAM (Ours)	RGB	✓	✓*	✓	3D Gaussian Splatting

Table 7. **Overview of Baseline Methods.** 'Dynamic' indicates whether the method explicitly addresses dynamic scenes. 'Open Source' specifies if a public implementation is available. 'Prior Free' refers to not using class priors, where 'O' represents object detection and 'S' denotes semantic segmentation. In all our experiments, we employ the RGB mode of MonoGS [30].

mization, we also deactivate the disparity regularization term in Eq. (5). These poses are further refined using an L1 RGB re-rendering loss, as employed in MonoGS [30], weighted by the uncertainty map.

7.2. Baseline Details

The characteristics of all baseline methods are presented in Table 7. Here we detail the source of each baseline method's results tabulated in the main paper and include the implementation details of MonST3R-SW (our sliding window extension of MonST3R [61]).

Details for Table 3. For tracking performance on the Bonn RGB-D Dynamic Dataset [36], results for ORB-SLAM2 [32], NICE-SLAM [68], and DDN-SLAM [27] are taken from the DDN-SLAM [27] paper. Results for DROID-SLAM [47] are taken from the DynaMoN [40] paper. Results for DynaSLAM (N+G) [2] and ReFusion [36] are taken from the ReFusion [36] paper. DG-SLAM [54], RoDyn-SLAM [16], and DynaMoN [40] are not open-sourced by the time of submission, therefore we take results from their own paper. The results for DSO [7], MonoGS [30], Splat-SLAM [39], and MonST3R-SW [61] are obtained by running their open-source implementation.

Details for Table 4. For tracking performance on the TUM RGB-D Dataset [44], results for Refusion [36], DG-SLAM [54], DynaSLAM (N+G) [2], RoDyn-SLAM [16], and DDN-SLAM [27] are based on data reported in their respective papers. Results for ORB-SLAM2 [32], DROID-SLAM [47], and DynaMoN [40] are sourced from the DynaMoN [40] paper. For DSO [7], NICE-SLAM [68], MonoGS [30], Splat-SLAM [39], and MonST3R-SW [61] results were obtained by running their open-source code.

MonST3R-SW. High VRAM usage is required for MonST3R [61], making it impractical to process an entire sequence as input. Instead, we apply a sliding window approach, merging overlapping frames from consecutive windows to form a complete sequence. Specifically, we use a window of 30 frames with a stride of 3, as in the original paper, and maintain an overlap of 25 frames to ensure consistent alignment. We employ Sim(3) Umeyama alignment [48] to integrate each new window's trajectory with the global trajectory.

8. Additional Experiments

Time Analysis. Table 9 presents the average fps of our method and the baselines. We also provide a fast version to support more efficient processing with minimal loss of accuracy by disabling low-impact processes and reducing

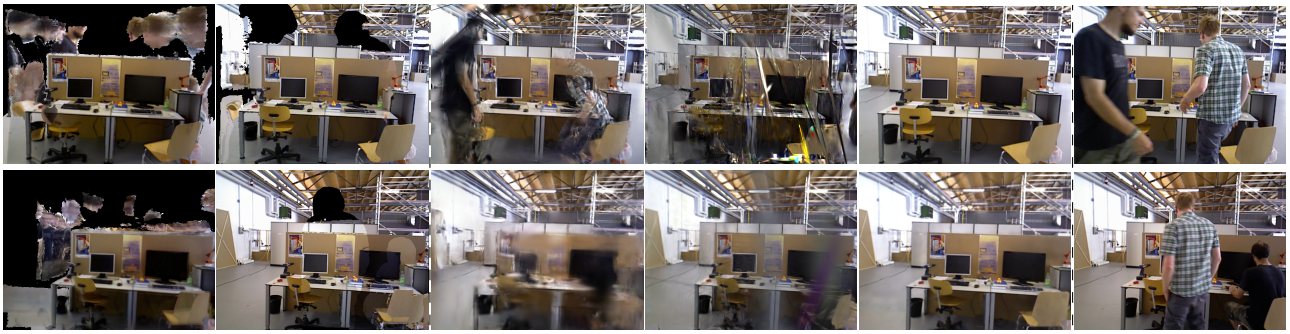


Figure 8. ***Input* View Synthesis Results on TUM RGB-D Dataset [44].** We show results on the `freiburg3_walking_static` (first row) and `freiburg3_walking_xyz` (second row) sequences. Our method produces substantially better rendering results.

Method	f2/dp	f3/ss	fr3/sx	f3/sr	f3/shs	f3/ws	f3/wx	f3/wr	f3/whs	Avg.
RGB-D										
Refusion [36]	4.9*	0.9	4.0	13.2*	11.0	1.7	9.9	40.6*	10.4	10.73*
ORB-SLAM2 [32]	**0.6**	0.8	1.0	2.5	2.5	40.8	72.2	80.5	72.3	30.4
DynaSLAM (N+G) [2]	0.7*	0.5*	1.5	2.7*	1.7	0.6	1.5	3.5	2.5	1.7*
NICE-SLAM [68]	88.8	1.6	32.0	59.1	8.6	79.8	86.5	244.0	152.0	83.6
DG-SLAM [54]	3.2	-	1.0	-	-	0.6	1.6	4.3	-	-
RoDyn-SLAM [16]	-	-	-	-	4.4	1.7	8.3	-	5.6	-
DDN-SLAM (RGB-D) [27]	-	-	1.0	-	1.7	1.0	1.4	3.9	2.3	-
Monocular										
DSO [7]	2.2	1.7	11.5	3.7	12.4	1.5	12.9	13.8	40.7	11.1
DROID-SLAM [47]	**0.6**	**0.5**	0.9	2.2	**1.4**	1.2	1.6	4.0	2.2	1.62
MonoGS [30]	112.8	1.2	6.1	5.1	28.3	1.1	21.5	17.4	44.2	26.4
Splat-SLAM [39]	0.7	**0.5**	0.9	2.3	1.5	2.3	**1.3**	3.9	2.2	1.71
DynaMoN (MS) [40]	**0.6**	**0.5**	0.9	**2.1**	1.9	1.4	1.4	3.9	2.0	1.63
DynaMoN (MS&SS) [40]	0.7	**0.5**	0.9	2.4	2.3	0.7	1.4	3.9	1.9	1.63
DDN-SLAM (RGB) [27]	-	-	1.3	-	3.1	2.5	2.8	8.9	4.1	-
MonST3R-SW [61]	51.6	2.4	28.2	5.4	36.5	2.2	27.3	13.6	19.8	20.8
WildGS-SLAM (Ours)	1.4	**0.5**	**0.8**	2.4	2.0	**0.4**	**1.3**	**3.3**	**1.6**	**1.51**

Table 8. **Tracking Performance on TUM RGB-D Dataset [44]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first** , second , and third . For methods without complete scene coverage in the original reports, results obtained by running their open-source code are marked with '*'. If open-source code is unavailable, scenes without results are marked with '-'. DynaSLAM (RGB) [2] consistently fails to initialize or experiences extended tracking loss across all sequences and therefore cannot be included in this table.

iterations. To be more specific, the modifications involve (i) removing the calculation of disparity regularization mask; (ii) optimizing the map $\mathcal{G}$ and the uncertainty MLP $\mathcal{P}$ every 5 keyframes; (iii) skipping the refinement of non-keyframe pose via re-rendering loss; (iv) decrease the number of iterations of final map refinement to 3000. As shown in Table 9, the fast version still outperforms baselines by a clear margin with comparable runtime.

Rendering Results on TUM RGB-D Dataset [44]. Our method effectively removes distractors, as illustrated in Fig. 8. ReFusion [36] struggles to fully eliminate distractors, leading to the presence of multiple ghosting artifacts. DynaSLAM (N+G) [2] exhibits "black holes" due to insufficient multiview information for effective inpainting, while the regions it does manage to inpaint often suffer from noticeable whitish artifacts. MonoGS [30] and Splat-SLAM [39] exhibit blurry and floating artifacts as they do not explicitly address dynamic environments.

Dataset	MonoGS [30]		Splat-SLAM [39]		**Ours-full**		**Ours-fast**	
	FPS ↑	ATE ↓	FPS ↑	ATE ↓	FPS ↑	ATE ↓	FPS ↑	ATE ↓
Wild-SLAM	2.41	47.99	2.44	8.71	0.49	0.46	1.96	0.48
Bonn	2.98	22.80	1.99	-	0.50	2.31	2.13	2.47

Table 9. **Running time evaluation.** For each dataset, we report the average FPS and RMSE of ATE [cm]. We logged the total running time to process a sequence and compute FPS by dividing the total number of processed frames by the total running time. *Ours-full* is the full pipeline presented, while *Ours-fast* is a fast version of WildGS-SLAM.

Full Tracking Results on the TUM RGB-D Dataset [44]. We report our performance on the full TUM RGB-D

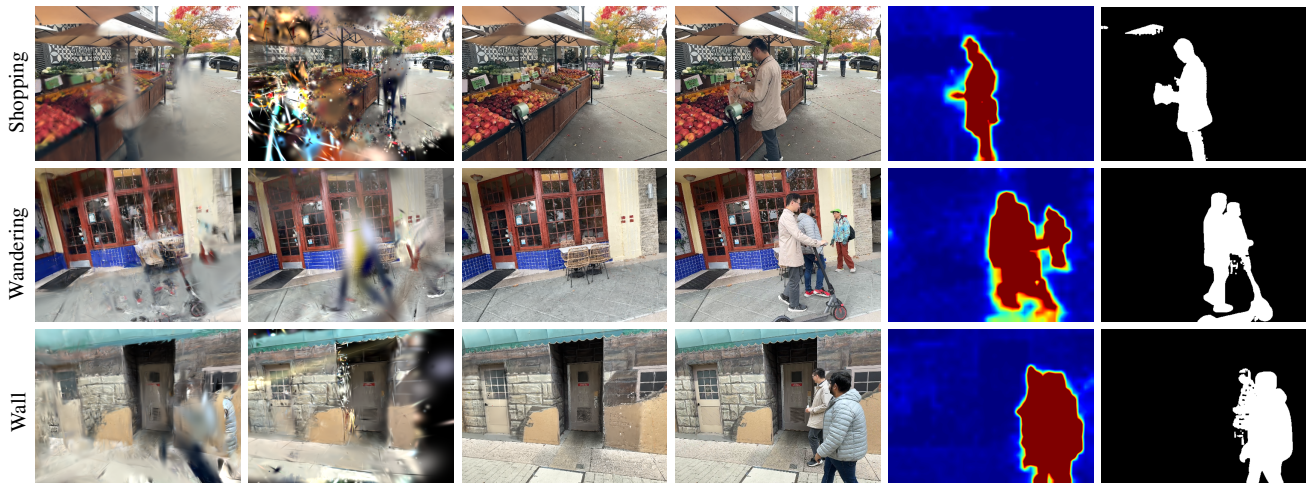


Figure 9. **Additional *Input* View Synthesis Results on our Wild-SLAM iPhone Dataset.** Faces are blurred to ensure anonymity.

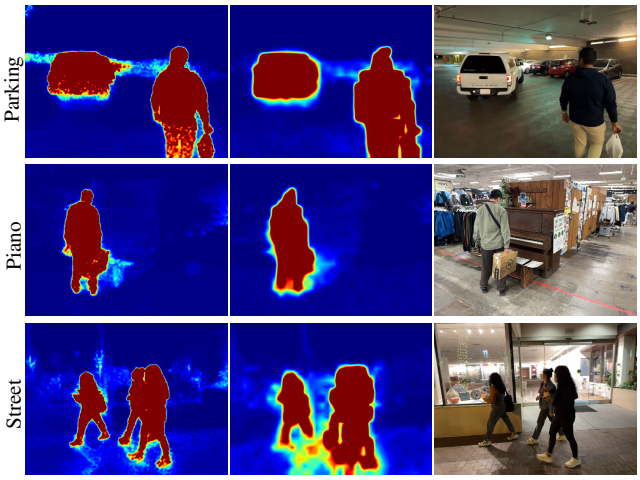


Figure 10. **High Resolution Uncertainty Map.**

Dataset [44] dynamic sequences in Table 8. Our method performs the best on average.

High Resolution Uncertainty Map. In Fig. 10, we present the visualization of high-resolution maps, as referenced in Fig. 5. Achieving higher resolution and sharper uncertainty maps is possible, though it comes at the cost of computational efficiency.

More Results on our Wild-SLAM iPhone Dataset. In addition to the results shown in Fig. 5 of the main paper, we provide additional results in Fig. 9.

Online Uncertainty Prediction. We visualize the online uncertainty prediction for frame 215—with MLP trained before, during, and after the umbrella enters the scene—in Fig. 11. Before (trained until frame 80), the MLP mainly classifies the moving human as a moving distractor since it has never seen the umbrella in the first 80 frames. As the umbrella enters the scene (frame 215), our uncertainty prediction module rapidly identifies it as a moving distractor due to the inconsistency between the Gaussian map and the frame 215. Moreover, the uncertainty estimate stabilizes shortly afterward (frame 451).

	fr1/desk	fr2/xyz	fr3/off	Avg.
RGB-D				
ORB-SLAM2 [32]	**1.6**	0.4	**1.0**	**1.0**
NICE-SLAM [68]	2.7	1.8	3.0	2.5
Monocular				
DROID-SLAM [47]	1.8	0.5	2.8	1.7
MonoGS [30]	3.8	5.2	2.9	4.0
Splat-SLAM [39]	**1.6**	**0.2**	1.4	1.1
WildGS-SLAM (Ours)	1.7	0.3	1.4	1.1

Table 10. **Tracking Performance on TUM RGB-D Dataset (Static) [44]** (ATE RMSE ↓ [cm]). Best results are highlighted as **first** , second , and third . Results for ORB-SLAM2 [32] and NICE-SLAM [68] are taken from NICE-SLAM [68]. Results for MonoGS [30] and Splat-SLAM [39] are taken from Splat-SLAM [39]. The results for DROID-SLAM [47] are obtained by running their open-source code.

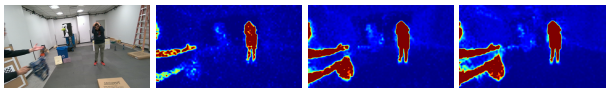


Figure 11. **Online Uncertainty Prediction.**

Pure Static Sequences. To demonstrate the robustness of our method, we also evaluate it on static sequences from the TUM RGB-D Dataset [44], shown in Table 10. Our approach performs on par with state-of-the-art monocular Gaussian Splatting SLAM methods, such as MonoGS [30] and Splat-SLAM [39].

Ablation Study on Disparity Regularization. Table 11 presents an ablation study evaluating the effects of (a) the

	Disp. Reg. Mask M	No Disp. Reg. in Final Global BA	Wild-SLAM		Bonn		TUM	
			Before BA	After BA	Before BA	After BA	Before BA	After BA
(i)	✗	✗	3.12	1.95	4.34	2.56	2.17	1.86
(ii)	✓	✗	**2.90**	1.57	3.97	2.56	**1.92**	1.69
(iii)	✗	✓	3.17	**0.46**	4.40	2.47	2.16	**1.55**
(iv)	✓	✓	2.92	**0.46**	**3.89**	**2.31**	1.94	1.63

Table 11. **Ablation Study on Disparity Regularization** (ATE RMSE ↓ [cm]). For each dataset, we report the average tracking error before and after the final global BA. 'Before BA' denotes before final global BA. 'After BA' denotes after final global BA.

	Wild-SLAM	Bonn	TUM
MonST3R Mask	2.60	2.58	1.80
YOLOv8 + SAM Mask	3.06	2.37	1.65
WildGS-SLAM (Ours)	**0.46**	**2.31**	**1.63**

Table 12. **Ablation Study on Distractor Estimation.** (ATE RMSE ↓ [cm]). For each dataset, we report the average tracking error.

disparity regularization mask M used in DBA (Eq. (5)) during on-the-fly capture and (b) the exclusion of the disparity regularization term in the final global BA. Removing M *(rows iii and iv)* has minimal impact on the final global BA, as shown in the 'After BA' results. However, the 'Before BA' results are significantly degraded, highlighting the multi-view inconsistencies in monocular predictions. Excluding the disparity regularization term in the final global BA *(rows ii and iv)* has no effect on the 'Before BA' results (minor deviations are expected due to randomness and initialization) but leads to improved 'After BA' performance. This improvement is attributed to the availability of multiple views in the final global BA, which refines depth accuracy compared to monocular predictions. The best results are achieved when M is applied and the disparity regularization term is excluded during the final global BA *(row iv)*, validating our design choices.

Ablation Study on Distractor Estimation. We compare various distractor estimation methods and utilize their resulting distractor masks for tracking in WildGS-SLAM. For the MonST3R mask, we aggregate masks from multiple runs because MonST3R supports only a limited number of images per run. The YOLOv8 + SAM mask corresponds to (c) in Table 5; we include it here for a clearer comparison. As shown in Table 12, our method consistently outperforms others, as other approaches struggle to produce accurate enough masks, particularly on our Wild-SLAM dataset, which features diverse and complex distractors.

Ablation Study on Pretrained Models. We conduct an ablation study on the pretrained models, namely the depth estimator and the feature exactor DINOv2 model, as presented in Table 13. Both novel view synthesis and tracking evaluations confirm that Metric3D V2 [13], combined with the finetuned DINOv2 model [57], achieves the best overall performance, validating our design choices.

Depth Estimator	DINOv2 model	Wild-SLAM		Bonn	TUM
		PSNR ↑	ATE ↓	ATE ↓	ATE ↓
DPTv2 [56]	Original [35]	20.56	0.47	2.36	1.76
DPTv2 [56]	Finetuned[57]	20.57	0.47	2.41	1.66
Metric3D V2 [13]	Original [35]	**20.58**	0.52	**2.31**	**1.61**
Metric3D V2 [13]	Finetuned[57]	**20.58**	**0.46**	**2.31**	1.63

Table 13. **Ablation Study on Different Pretrained Models**. For the Wild-SLAM dataset, we report the novel view synthesis results (PSNR ↑) and the tracking error (ATE RMSE ↓ [cm]). For the Bonn and TUM datasets, we report the average tracking error (ATE RMSE ↓ [cm]).

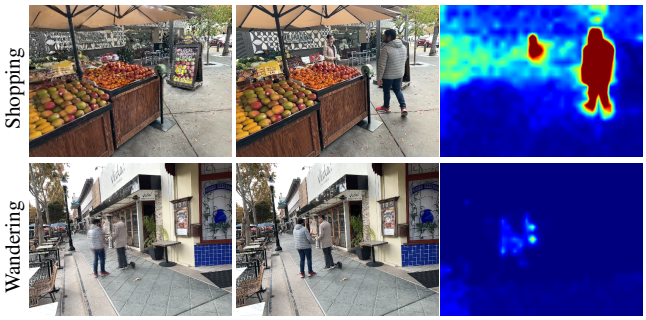


Figure 12. **Failure Cases.** In shopping dataset, patterns on the wall is incorrectly regarded as medium uncertainty because of the difficulty of reconstructing the complicated textures. In wandering, humans are not removed due to the lack of observation of the static scene. Faces are blurred to ensure anonymity.

Failure Cases. In Fig. 12, we present two failure cases of our method. In the first case, while our method successfully removes dynamic objects, it struggles to reconstruct the complex background, leading to a high SSIM loss in Eq. (4). Therefore, the high SSIM loss drives the uncertainty prediction to incorrectly assign higher uncertainty to static regions.

In the second case, the dynamic objects remain stationary in some of the frames and, since all frames are captured from roughly the same camera direction, no earlier frames are available to observe the static scene without the dynamic objects. As a result, the system assigns lower uncertainty to these regions and mistakenly reconstructs the dynamic objects.

References

[1] Thomas Belos, Pascal Monasse, and Eva Dokladalova. Mod slam: Mixed method for a more robust slam without loop closing. In *VISAPP*, 2022. 2

[2] 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. 2, 5, 6, 7, 8, 3, 4

[3] Jiyu Cheng, Yuxiang Sun, and Max Q-H Meng. Improving monocular visual slam in dynamic environments: an optical-flow-based approach. *Advanced Robotics*, 2019. 2

[4] 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. 1, 2

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

[6] Alexander Duda and Udo Frese. Accurate detection and localization of checkerboard corners for calibration. In *BMVC*, 2018. 1

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 1, 2, 5, 6, 8, 3, 4

[8] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 5

[9] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgbd gs-icp slam. *arXiv preprint arXiv:2403.12550*, 2024. 2

[10] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103:267–305, 2013. 1

[11] 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

[12] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2

[13] 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, 6

[14] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[15] Intel RealSense. Intel® RealSense™ Depth Camera D455, 2024. 5, 1

[16] 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. 2, 5, 8, 3, 4

[17] 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

[18] 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. 2

[19] 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

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

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 8

[22] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*, 2007. 2

[23] Evgenii Kruzhkov, Alena Savinykh, Pavel Karpyshev, Mikhail Kurenkov, Evgeny Yudin, Andrei Potapov, and Dzmitry Tsetserukou. Meslam: Memory efficient slam based on neural fields. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022. 2

[24] 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, 3, 4

[25] Kunyi Li, Michael Niemeyer, Nassir Navab, and Federico Tombari. Dns slam: Dense neural semantic-informed slam. *arXiv preprint arXiv:2312.00204*, 2023. 2

[26] Linfei Li, Lin Zhang, Zhong Wang, and Ying Shen. Gs{3}lam: Gaussian semantic splatting slam. In *ACM MM*, 2024. 2

[27] 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, 8, 3, 4

[28] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2

[29] 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, 6, 7, 8

[30] 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, 3, 4, 5, 6, 7, 8

[31] Goeffrey J McLachlan. Mahalanobis distance. *Resonance*, 1999. 4

[32] 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, 5, 8, 3, 4

[33] 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, 2

[34] OptiTrack. Optitrack - motion capture systems. 5, 1

[35] 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. 3, 4, 6

[36] 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. 1, 2, 5, 6, 7, 8, 3, 4

[37] Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *SIGGRAPH 2024 Conference Papers*, 2024. 2

[38] 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, 3, 4, 5

[39] 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, 4, 5, 6, 7, 8, 3

[40] 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. 2, 5, 8, 3, 4

[41] 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

[42] Rulin Shen, Kang Wu, and Zhi Lin. Robust visual slam in dynamic environment based on motion detection and segmentation. *Journal of Autonomous Vehicles and Systems*, 2024. 2

[43] 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. 2

[44] 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, 8, 3, 4

[45] 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

[46] Yuxiang Sun, Ming Liu, and Max Q-H Meng. Motion removal for reliable rgb-d slam in dynamic environments. *Robotics and Autonomous Systems*, 2018. 2

[47] 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, 4, 5, 6, 8, 2, 3

[48] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1991. 5, 3

[49] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[50] 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. 3

[51] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 3

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing (TIP)*, 2004. 5

[53] Wenxin Wu, Liang Guo, Hongli Gao, Zhichao You, Yuekai Liu, and Zhiqiang Chen. Yolo-slam: A semantic slam system towards dynamic environment with geometric constraint. *Neural Computing and Applications*, 2022. 2

[54] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. DG-SLAM: Robust Dynamic Gaussian Splatting SLAM with Hybrid Pose Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2, 5, 8, 3, 4

[55] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[56] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 6

[57] 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. 2, 3, 4, 6

[58] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 3

[59] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 2

[60] 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. 4

[61] 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. 3, 5, 6, 7, 8, 4

[62] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[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] Wei Zhang, Tiecheng Sun, Sen Wang, Qing Cheng, and Norbert Haala. Hi-slam: Monocular real-time dense mapping with hybrid implicit fields. *IEEE Robotics and Automation Letters*, 2023. 2

[65] Yiming Zhao, Taein Kwon, Paul Streli, Marc Pollefeys, and Christian Holz. Egopressure: A dataset for hand pressure and pose estimation in egocentric vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1

[66] Liyuan Zhu, Yue Li, Erik Sandström, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2025. 2

[67] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[68] 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, 8, 3, 4

[69] 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