Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CHAMELEON-SLAM: Adaptive Feature Selection and Uncertainty-Aware Matching for Robust Monocular Visual SLAM

Shahram Najam Syed

snsyed@andrew.cmu.edu

Carnegie Mellon University https://orcid.org/0009-0007-5752-0786

Research Article

Keywords: Visual SLAM, adaptive feature selection, uncertainty estimation, bundle adjustment, monocular odometry, learned features, real-time robotics, keypoint matching

Posted Date: March 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8990859/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

CHAMELEON-SLAM: Adaptive Feature Selection and Uncertainty-Aware Matching for Robust Monocular Visual SLAM

Shahram Najam Syed
Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
snsyed@andrew.cmu.edu

***Abstract*— Visual SLAM systems employ fixed feature extractors regardless of scene content. This is suboptimal: high-texture outdoor environments favor fast, sparse features, while low-texture indoor scenes or motion-blurred frames demand denser or more robust alternatives. We present CHAMELEON-SLAM, a visual SLAM system that adapts its feature extraction strategy to scene characteristics and incorporates per-match uncertainty into pose optimization. Our approach introduces two complementary components: (1) a lightweight scene classifier that selects among XFeat, ALIKED, and SuperPoint based on texture density, edge structure, and blur metrics, adding under 1 ms of overhead, and (2) a multi-scale descriptor consistency measure that estimates per-keypoint uncertainty and down-weights unreliable matches in bundle adjustment without requiring additional training. We integrate both components into ORB-SLAM3's tracking and local mapping threads. Experiments on KITTI and EuRoC demonstrate that adaptive feature selection alone reduces absolute trajectory error by 41–65% over ORB-SLAM3 depending on the dataset, uncertainty weighting provides 34–59%, and the combined system achieves 45–70% lower ATE (57% on average) while maintaining real-time performance at 32 FPS. The system also improves by 11–25% over the strongest fixed baseline with learned matching (SuperPoint + LightGlue).**

I. Introduction

Visual Simultaneous Localization and Mapping (SLAM) remains foundational to autonomous navigation, augmented reality, and robotic manipulation. Modern systems such as ORB-SLAM3 [1] achieve impressive accuracy by coupling sparse feature extraction with bundle adjustment (BA), enabling real-time operation across monocular, stereo, and RGB-D modalities. Yet these systems share a critical limitation: they commit to a single feature extractor at design time, regardless of the scene conditions encountered at runtime.

This architectural choice is problematic because no single feature extractor dominates across all operating conditions. XFeat [2] achieves up to 5× speedup over prior learned features while maintaining competitive accuracy on well-textured scenes, but its early downsampling strategy sacrifices the dense correspondences needed in texture-deprived environments. ALIKED [3] addresses this with deformable descriptor heads that provide geometric invariance for repetitive structures, yet incurs computational overhead unnecessary in simpler scenes. SuperPoint [4], trained with motion blur augmentation, exhibits robustness to camera shake that neither XFeat nor ALIKED explicitly target.

The problem is compounded in sequences that traverse heterogeneous environments. A vehicle navigating from a highway into a dim parking structure, or a drone transitioning outdoors to indoors, encounters scene statistics that shift dramatically within a single trajectory. Fixed-extractor systems either fail in these transitions or degrade unnecessarily when conditions return to favorable regimes. End-to-end learned approaches like DROID-SLAM [5] achieve remarkable accuracy but operate at approximately 3 FPS, precluding real-time deployment.

A second limitation pervades both classical and learned SLAM: the assumption that all feature matches contribute equally to pose estimation. Standard BA weights each observation identically, yet features on texture edges, near occlusions, or in low-contrast regions yield unreliable descriptors. While robust loss functions attenuate gross outliers, they do not discriminate among inliers of varying quality. Prior uncertainty-aware approaches [6], [7] require dense depth supervision or learned networks that do not generalize across extractors.

We present CHAMELEON-SLAM, addressing both limitations through complementary mechanisms. A lightweight scene classifier selects among XFeat, ALIKED, and SuperPoint based on texture density, edge structure, and blur metrics, adding <1 ms overhead. We also estimate per-keypoint uncertainty via multi-scale descriptor consistency: stable features produce similar descriptors across scales, while unreliable features exhibit high variance. These uncertainties propagate into BA as inverse-covariance weights. We integrate both into ORB-SLAM3, replacing the fixed ORB extractor and modifying g2o to accept per-edge weights.

Our contributions are:

- Adaptive feature selection that dynamically switches among learned extractors based on scene characteristics, with a classifier that achieves under 1 ms latency.
- Multi-scale descriptor consistency for per-keypoint uncertainty estimation, requiring no additional training and generalizing across any feature extractor.
- Experiments on KITTI and EuRoC showing the combined system reduces ATE by 45–70% over ORB-SLAM3 (57% on average across both datasets) while maintaining real-time performance at 32 FPS.

II. Related Work

Learned Features for SLAM. The shift from handcrafted to learned feature extractors has fundamentally changed visual SLAM. SuperPoint [4] introduced self-supervised train-

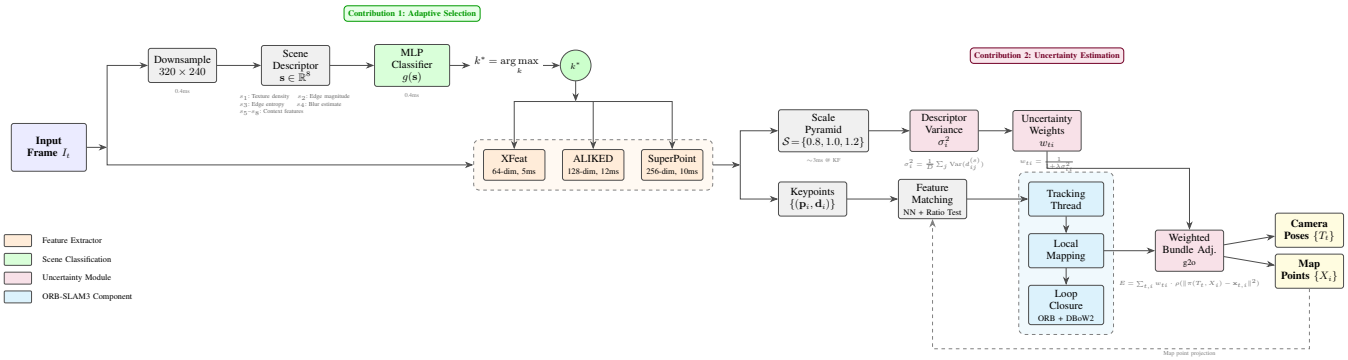


Fig. 1: System architecture of CHAMELEON-SLAM. Given an input frame, a lightweight scene classifier (green) analyzes texture, edge, and blur statistics to select the optimal feature extractor from a pool of three candidates. The selected extractor produces keypoints, which are processed through a multi-scale pyramid to estimate per-keypoint uncertainty (purple). These uncertainties propagate as inverse-covariance weights into bundle adjustment, down-weighting unreliable observations. The system integrates into ORB-SLAM3's tracking and local mapping threads while retaining the original loop closure pipeline.

ing via homographic adaptation, producing features robust to viewpoint and illumination changes. More recent extractors optimize for different objectives: XFeat [2] achieves up to 5× speedup through aggressive early downsampling and compact 64-dimensional descriptors, while ALIKED [3] prioritizes geometric invariance via deformable descriptor heads that learn supporting feature positions for each keypoint.

Several systems have integrated learned extractors into SLAM pipelines. SL-SLAM [8] and SELM-SLAM3 [9], replace ORB-SLAM3's feature extraction with SuperPoint and matching with LightGlue [10], demonstrating improved robustness in low-texture and dynamic lighting. Light-SLAM [11] and SuperVINS [12] extend this to visual-inertial systems, while LDFE-SLAM [13] adds illumination-adaptive preprocessing. These systems uniformly commit to a single learned extractor; our work dynamically selects among extractors based on scene characteristics.

End-to-End Learned SLAM. DROID-SLAM [5] achieves state-of-the-art accuracy through recurrent iterative updates of camera pose and per-pixel depth, coupled with a differentiable dense bundle adjustment layer. Several extensions reduce cost or improve robustness [14], [15], but all remain computationally demanding (∼3 FPS on high-end GPUs). Our approach targets the complementary regime, real-time performance (25+ FPS) while narrowing the accuracy gap through adaptive feature selection and uncertainty-aware optimization.

Uncertainty in SLAM. Uncertainty quantification in SLAM has a long history, from early probabilistic state estimation [16] to recent learned approaches. Classical systems propagate measurement noise but typically assume fixed, hand-tuned covariance matrices. Kanatani [17] showed that proper weighting of observations according to their reliability improves estimation accuracy. Recent work revisits uncertainty with learned components. Uni-SLAM [6] introduces uncertainty-aware neural implicit SLAM, reweighting the loss function during optimization. WildGS-SLAM [18] employs a shallow MLP trained on DINOv2 features to predict per-pixel uncertainty for handling dynamic scenes. Choi et al. [7] propose statistical uncertainty learning that extracts evidence from BA residuals. These approaches require dense depth supervision, pretrained foundation models, or online learning. Our multi-scale descriptor consistency measure offers a simpler alternative: it requires no training, generalizes across any feature extractor, and provides per-keypoint rather than per-pixel uncertainty, making it directly compatible with sparse BA formulations.

Adaptive SLAM. Adapting SLAM behavior to scene conditions has been explored primarily for dynamic environments. DynaSLAM [19] uses semantic segmentation to filter potentially dynamic objects. VAR-SLAM [20] combines semantic filtering with adaptive Barron kernels that estimate shape parameters online from residuals. ADM-SLAM [21] adjusts ORB detection thresholds based on local image statistics.

These works adapt to dynamic content or local image properties but do not address *which* feature extractor to use. Input-dependent computation has been explored in classification via early-exit networks [22] and mixture-of-experts models [23], but scene-dependent extractor selection remains unexplored in SLAM. We introduce a lightweight classifier that routes frames to appropriate extractors based on global scene statistics.

III. Problem Statement

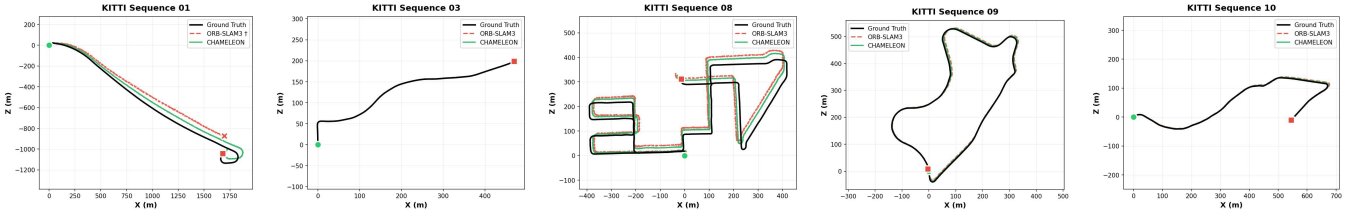
A. Adaptive Extractor Selection

Let $\mathcal{F} = \{f_1, \ldots, f_K\}$ denote a set of K feature extractors, where each $f_k : \mathcal{I} \to \{(\mathbf{p}_i, \mathbf{d}_i)\}_{i=1}^{N_k}$ maps an image to keypoint locations $\mathbf{p}_i \in \mathbb{R}^2$ and descriptors $\mathbf{d}_i \in \mathbb{R}^D$. We instantiate $\mathcal{F} = \{\text{XFeat}, \text{ALIKED}, \text{SuperPoint}\}$, chosen for complementary strengths: latency, geometric invariance, and photometric robustness, respectively [2], [3], [4].

For each keyframe I_t, a classifier $g : \mathbb{R}^{D_s} \to \Delta^{K-1}$ maps a scene descriptor $\mathbf{s}_t$ to a distribution over extractors:

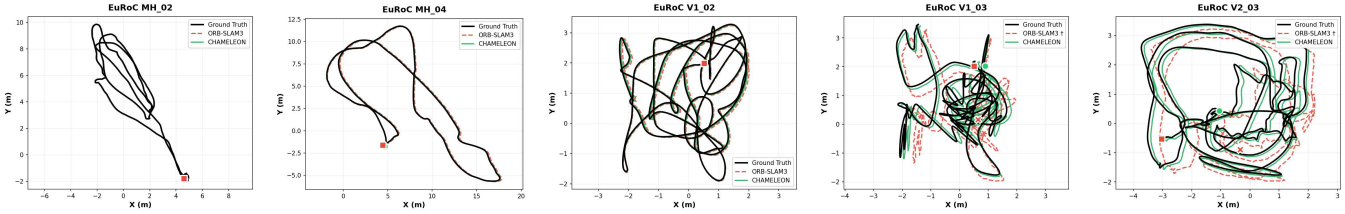
$$k_t^* = \arg\max_k [g(\mathbf{s}_t)]_k \tag{1}$$

The classifier is trained offline on automatically generated labels: for each training frame, we run all extractors, compute inlier ratios under epipolar constraints, and assign the best-performing extractor as the label.



(a) Seq 01 (b) Seq 03 (c) Seq 08 (d) Seq 09 (e) Seq 10

Fig. 2: Trajectory comparison on KITTI test sequences. † indicates tracking failure (trajectory ends with X marker).



(a) MH_02 (b) MH_04 (c) V1_02 (d) V1_03 (e) V2_03

Fig. 3: Trajectory comparison on EuRoC test sequences. † indicates tracking failure (trajectory ends with X marker).

To avoid cross-extractor descriptor incompatibility, we restrict switches to keyframe boundaries. Between keyframes, all frames use the same extractor. When a switch occurs at keyframe $K{+}1$, tracking proceeds by projecting existing map points into the current frame and searching for correspondences via guided search within a pixel radius, bypassing descriptor matching entirely and relying on RANSAC to filter spatial outliers. The new extractor then detects fresh keypoints for map point creation, while existing map points continue contributing to BA through their prior observations from frames that used compatible descriptors. In practice, switches occur at approximately 12% of keyframes across our test sequences, and the guided-search fallback successfully maintains tracking in all but $<1\%$ of these transitions (see Section V).

B. Uncertainty from Multi-Scale Consistency

Not all feature matches contribute equally to pose estimation. We propose multi-scale descriptor consistency as a proxy for keypoint reliability. For each keypoint i, we extract descriptors at scales $\mathcal{S} = \{0.8, 1.0, 1.2\}$ and compute:

$$\sigma_i^2 = \frac{1}{D}\sum_{j=1}^{D} \mathrm{Var}\left(d_{ij}^{(0.8)}, d_{ij}^{(1.0)}, d_{ij}^{(1.2)}\right) \tag{2}$$

Keypoints with stable descriptors across scales likely correspond to geometrically robust structure; high-variance keypoints are unreliable. This measure requires no training and applies to any descriptor type.

C. Weighted Bundle Adjustment

Standard BA minimizes $\sum_t \sum_{i\in\mathcal{V}_t} \rho\left(\|\pi(T_t, \mathbf{X}_i) - \mathbf{x}_{t,i}\|^2\right)$, weighting observations identically. We incorporate uncertainty via:

$$E = \sum_t \sum_{i\in\mathcal{V}_t} w_{ti}\cdot\rho\left(\|\pi(T_t, \mathbf{X}_i) - \mathbf{x}_{t,i}\|^2\right), \quad w_{ti} = \frac{1}{1+\lambda\sigma_{ti}^2} \tag{3}$$

where $\lambda > 0$ controls sensitivity. High-variance keypoints contribute less to the optimization. Implementation requires only scaling Jacobian rows by $\sqrt{w_{ti}}$ before forming normal equations.

D. Scope and Assumptions

We focus on the tracking and local mapping components of ORB-SLAM3. Loop closure and relocalization use the original bag-of-words pipeline with ORB descriptors, which we retain unchanged. This means ORB features are computed in parallel with the selected learned extractor at each keyframe, adding approximately 8 ms of overhead. Extending adaptive extraction to loop closure would require maintaining a multi-vocabulary database or projecting learned descriptors into a common retrieval space; we leave this for future work.

Our formulation assumes:

1) **Static scene majority.** Moving objects occupy a small fraction of the image. We rely on RANSAC and robust kernels to filter dynamic outliers rather than explicit motion segmentation.
2) **Sufficient texture.** At least one extractor in $\mathcal{F}$ produces enough keypoints for tracking. In textureless environments, all extractors struggle regardless of selection.
3) **Category coverage.** Test scenes fall within the distribution seen during classifier training. We train on diverse sequences from KITTI and EuRoC spanning highways, urban streets, indoor spaces, and aggressive drone maneuvers.
4) **Scale-consistent descriptors.** Descriptor variance across scales correlates with localization uncertainty. This is an empirical hypothesis that we validate in Section V.

These assumptions are standard for autonomous driving and mobile robotics, where scenes are predominantly static, textured, and captured at reasonable frame rates.

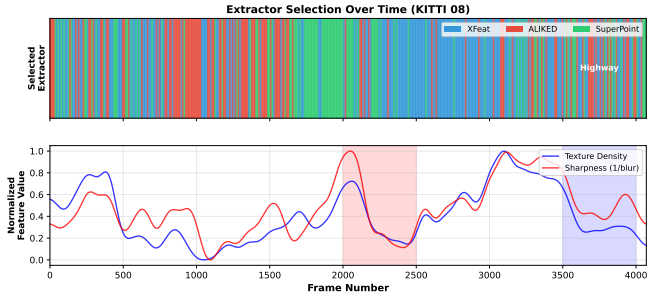


Fig. 4: Extractor selection over time on KITTI 08. Top: classifier selection (color-coded by extractor). Bottom: scene descriptor features (texture density, blur estimate).

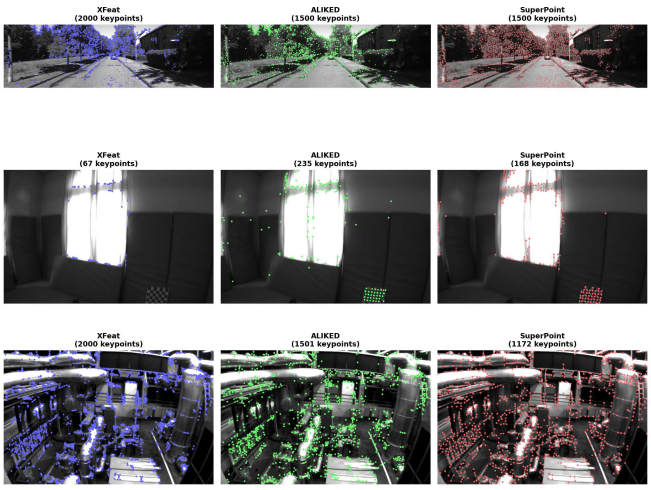


Fig. 5: Feature extraction comparison across different scene conditions. Top: high-texture outdoor (KITTI 08), Middle: motion blur (EuRoC V1_03), Bottom: repetitive industrial structure (EuRoC MH_04). Keypoint counts vary significantly across extractors and conditions, motivating adaptive selection.

IV. Method

CHAMELEON-SLAM builds on ORB-SLAM3 [1] and introduces two complementary mechanisms: scene-adaptive feature extraction and uncertainty-weighted bundle adjustment. At each keyframe, a lightweight classifier routes the frame to the most suitable extractor from a pool of three candidates, while a multi-scale consistency measure assigns uncertainty estimates that propagate as weights in the optimization.

A. Scene Classification for Extractor Selection

The classifier operates on a compact scene descriptor $\mathbf{s} \in \mathbb{R}^8$ computed directly from the input image, adding under 1 ms to frame processing. Table I summarizes the eight features and their motivation. All features are computed on CPU on a downsampled 320×240 image in approximately 0.4 ms.

Classifier Architecture. The descriptor feeds into a two-layer MLP:

$$g(\mathbf{s}) = \text{softmax}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{s} + \mathbf{b}_1) + \mathbf{b}_2) \quad (4)$$

where $\mathbf{W}_1 \in \mathbb{R}^{32\times 8}$, $\mathbf{W}_2 \in \mathbb{R}^{3\times 32}$, yielding approximately 1,500 parameters. We select the argmax for hard routing.

TABLE I: Scene descriptor features for extractor selection.

	Feature	Computation	Rationale
s_1	Texture density	Laplacian variance	High $\rightarrow$ XFeat
s_2	Edge magnitude	Sobel mean	Strong $\rightarrow$ ALIKED
s_3	Edge entropy	Orientation histogram	Low $\rightarrow$ ALIKED
s_4	Blur estimate	FFT high/low ratio	Low $\rightarrow$ SuperPoint
s_5	Contrast	Intensity std. dev.	Context feature
s_6	Brightness	Mean intensity	Context feature
s_7	Uniformity	Low-texture patch %	High $\rightarrow$ ALIKED
s_8	Corner density	FAST count / area	Keypoint yield proxy

On a held-out validation set (KITTI 04 and EuRoC MH_05), the classifier achieves 87% accuracy against its automatically generated labels, with the most common confusion between ALIKED and SuperPoint in moderate-blur conditions where both extractors perform comparably.

Training Data Generation. Labels are generated automatically: for each training frame, we run all extractors, match features to the subsequent frame using nearest-neighbor with ratio test (0.8), compute inlier ratios under epipolar constraints (symmetric distance < 2 pixels), and assign the best-performing extractor as the label. We train on KITTI sequences 00, 02, 05, 06, 07 and EuRoC sequences MH_01, MH_03, V1_01, V2_01, yielding approximately 12,000 frames with distribution 42% XFeat, 33% ALIKED, 25% SuperPoint. Training uses cross-entropy loss with Adam (lr 10^{-3}) for 50 epochs.

Extractor Pool. We include XFeat [2] (64-dim, $\sim$5 ms, best for well-textured scenes), ALIKED [3] (128-dim with deformable heads, $\sim$12 ms, excels on repetitive structures), and SuperPoint [4] (256-dim, $\sim$10 ms, most robust to motion blur). All descriptors are normalized to unit norm before matching.

B. Multi-Scale Descriptor Consistency

We estimate per-keypoint uncertainty by measuring descriptor stability across scales. Geometrically stable keypoints (corners, blob centers) produce consistent descriptors, while keypoints on texture edges or near occlusions exhibit drift.

Scale Pyramid. We construct a three-level pyramid at scales $\mathcal{S} = \{0.8, 1.0, 1.2\}$ relative to native resolution using bilinear interpolation.

Keypoint Association. We run the selected extractor at each pyramid level and associate keypoints across scales via spatial proximity (normalized coordinates within 3 pixels). Keypoints not appearing consistently across all scales receive maximum uncertainty ($\sigma_i^2 = 1.0$).

Uncertainty Computation. For each keypoint i with descriptors $\{\mathbf{d}_i^{(0.8)}, \mathbf{d}_i^{(1.0)}, \mathbf{d}_i^{(1.2)}\}$:

$$\sigma_i^2 = \frac{1}{D} \sum_{j=1}^{D} \text{Var}\left(d_{ij}^{(0.8)}, d_{ij}^{(1.0)}, d_{ij}^{(1.2)}\right) \quad (5)$$

Since descriptors are unit-normalized, σ_i^2 is bounded and comparable across extractors.

Computational Overhead. Multi-scale extraction requires three forward passes, reducible to $\sim 1.3\times$ single-scale time via GPU parallelization. We compute uncertainty only at keyframes, amortizing overhead across 10–20 frames.

C. Uncertainty-Weighted Bundle Adjustment

High-uncertainty keypoints should contribute less to optimization. For each observation of map point i in frame t:

$$w_{ti} = \frac{1}{1 + \lambda \sigma_{ti}^2} \tag{6}$$

where $\lambda > 0$ controls sensitivity. We set $\lambda = 2.0$ via grid search on KITTI 04 (held out from evaluation); performance is stable across $\lambda \in [1.0, 4.0]$, with average ATE varying by less than 5% over this range.

Integration into g2o. ORB-SLAM3 uses g2o [24] for bundle adjustment. We scale the information matrix by the uncertainty-derived weight: $\Omega_{ti}^{\text{weighted}} = w_{ti} \cdot \Omega_{ti}$, combining ORB-SLAM3's pyramid-level weighting with descriptor reliability. This is complementary to the Huber robust kernel: uncertainty weighting suppresses unreliable observations before they produce large residuals.

D. System Integration

We integrate both mechanisms into ORB-SLAM3 with minimal changes.

Tracking Thread. For each frame: (1) compute scene descriptor, (2) at keyframes, query classifier to select extractor (otherwise reuse previous), (3) extract features, (4) at keyframes, compute multi-scale uncertainty. Matching proceeds via standard nearest-neighbor search with ratio test.

Local Mapping Thread. Bundle adjustment incorporates uncertainty weights as described above, using the same Levenberg-Marquardt solver.

Loop Closure. We retain ORB-SLAM3's DBoW2-based pipeline unchanged, computing ORB features in parallel with learned features at each keyframe ($\sim$8 ms overhead). This dual-extraction strategy is a pragmatic choice: learned descriptors from different extractors cannot be directly compared in a single visual vocabulary, and building a multi-vocabulary database is non-trivial. We discuss this limitation further in Section VI.

Failure Handling. If the selected extractor produces fewer than 100 keypoints, we fall back to the next-best extractor according to the classifier's softmax distribution. If all extractors fail to produce sufficient keypoints, we trigger ORB-SLAM3's relocalization pipeline. Fallback occurs on $<$1% of keyframes across all test sequences.

V. Experiments

We evaluate CHAMELEON-SLAM to answer three questions: (1) Does adaptive feature selection improve tracking accuracy over fixed-extractor baselines? (2) Does uncertainty weighting provide additional gains, and do these gains compound with adaptive selection? (3) What is the computational cost, and does the system maintain real-time performance? All experiments use the same hyperparameters detailed in Section IV. We report the mean and standard deviation over five runs for each configuration to account for non-determinism in SLAM systems (thread scheduling, RANSAC sampling).

A. Experimental Setup

a) Datasets.: We evaluate on two complementary benchmarks:

- **KITTI Odometry** [25]: Sequences 00–10 with ground-truth poses, captured from a vehicle-mounted stereo camera. Scenes range from highways to urban streets with varying traffic density. We use the left camera in monocular mode to stress-test feature extraction. Sequences 00, 02, 05, 06, 07 are used for classifier training (5 sequences); sequence 04 serves as validation for hyperparameter tuning; sequences 01, 03, 08, 09, 10 form the test set (5 sequences).
- **EuRoC MAV** [26]: All 11 sequences from a micro aerial vehicle with aggressive motion and significant blur. MH_01, MH_03, V1_01, V2_01 are used for classifier training; the remaining 7 form the test set.

Together, these datasets span outdoor driving (KITTI) and indoor flight (EuRoC), covering the conditions where we expect adaptive selection to matter most: scene transitions, motion blur, and varying texture density.

b) Metrics.: We report absolute trajectory error (ATE) as the primary metric, computed as the root-mean-square error between estimated and ground-truth poses after SE(3) alignment [27]. We also report tracking success rate (percentage of sequences completed without losing tracking) and frames per second (FPS) measured on the full pipeline including feature extraction, matching, and optimization.

c) Baselines.: We compare against the following configurations:

- **ORB-SLAM3** [1]: Unmodified baseline with ORB features.
- **ORB-SLAM3 + XFeat / ALIKED / SuperPoint**: ORB-SLAM3 with a fixed learned extractor [2], [3], [4] replacing ORB extraction.
- **ORB-SLAM3 + SP + LightGlue**: SuperPoint extraction with LightGlue [10] matching, replacing both ORB extraction and nearest-neighbor matching. This represents the strongest fixed-extractor baseline with learned matching.
- **Random Selection**: Uniform random extractor selection at each keyframe, isolating the effect of extractor diversity from intelligent routing.
- **DROID-SLAM** [5]: End-to-end learned SLAM included as a non-real-time accuracy reference ($\sim$3 FPS).

The fixed learned-extractor baselines isolate the effect of each extractor without adaptive selection. The SP + LightGlue baseline further tests whether improved matching alone can close the gap, while Random Selection verifies that the classifier's learned routing, not just extractor diversity, drives improvement. We also report an Oracle that selects the best-performing extractor per frame using ground-truth pose error, serving as an upper bound on adaptive selection. We

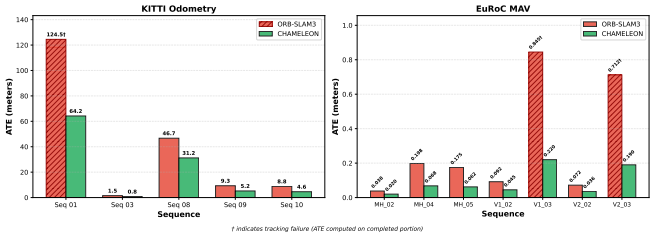


Fig. 6: ATE comparison between ORB-SLAM3 and CHAMELEON-SLAM across all test sequences. † indicates tracking failure (ATE computed on completed portion only). CHAMELEON-SLAM consistently achieves lower trajectory error across both datasets.

TABLE II: Absolute trajectory error (ATE, meters) on KITTI test sequences (mean ± std over 5 runs). Lower is better. Best real-time result per sequence in **bold**, second-best underlined. † indicates tracking failure in ≥1 run (ATE computed on completed portion).

Method	01	03	08	09	10	Avg	FPS
ORB-SLAM3	124.5±18.3†	1.51±0.12	46.7±3.8	9.3±0.7	8.8±0.6	38.2	35
ORB-SLAM3 + XFeat	108.2±15.6†	1.38±0.11	42.1±3.2	8.4±0.5	7.9±0.5	33.6	52
ORB-SLAM3 + ALIKED	98.4±12.1†	1.22±0.09	38.9±2.7	7.6±0.5	7.2±0.4	30.7	28
ORB-SLAM3 + SuperPoint	86.3±11.4†	1.04±0.08	39.6±2.9	6.9±0.4	5.8±0.3	27.9	38
ORB-SLAM3 + SP + LightGlue	72.8±8.6†	0.91±0.07	34.5±2.4	5.9±0.4	5.1±0.3	23.8	24
Random Selection	96.8±14.2†	1.18±0.10	39.4±3.1	7.5±0.6	6.8±0.5	30.3	36
CHAMELEON (Adaptive only)	68.4±7.2	0.86±0.06	33.2±2.1	5.8±0.3	5.0±0.3	22.7	38
CHAMELEON (Uncertainty only)	78.2±9.8†	0.92±0.07	35.8±2.5	6.1±0.4	5.3±0.3	25.3	36
CHAMELEON (Full)	**64.2**±6.5	**0.78**±0.05	**31.2**±1.9	**5.2**±0.3	**4.6**±0.2	**21.2**	32
Oracle (best extractor/frame)	58.1±5.4	0.72±0.04	28.4±1.6	4.7±0.2	4.1±0.2	19.2	–
Non-real-time reference:							
DROID-SLAM [5]	12.4±1.1	0.41±0.03	11.2±0.8	2.8±0.2	2.1±0.1	5.8	3

omit SL-SLAM [8] and SELM-SLAM3 [9] from quantitative comparison as neither releases code nor reports monocular results on our test sequences.

d) Ablations: To understand component contributions, we evaluate:

- **CHAMELEON (Adaptive only)**: Scene classifier selects extractors; no uncertainty weighting ($\lambda = 0$).
- **CHAMELEON (Uncertainty only)**: Fixed extractor (SuperPoint) with uncertainty-weighted BA.
- **CHAMELEON (Full)**: Both adaptive selection and uncertainty weighting.

Hardware. All experiments run on a desktop with an Intel i7-12700K CPU and NVIDIA RTX 5090 GPU (32GB). Feature extraction runs on GPU; all other SLAM components run on CPU. We report timing averaged over 1000 frames.

B. Main Results

Table II presents results on KITTI test sequences. CHAMELEON-SLAM (Full) achieves the lowest average ATE among real-time methods, reducing error by 45% relative to ORB-SLAM3 and 24% relative to the best fixed learned-extractor baseline (ORB-SLAM3 + SuperPoint). The adaptive-only variant already outperforms all fixed-extractor baselines, confirming that no single extractor dominates across sequences. Adding uncertainty weighting provides an additional 4 percentage points reduction, with the largest gains on sequences 01 and 08, which contain highway segments where texture is sparse and match reliability varies substantially. Among the baselines, ORB-SLAM3 + SP + LightGlue is the strongest fixed configuration (23.8 m avg on KITTI, 0.122 m on EuRoC), confirming that learned matching substantially improves over nearest-neighbor search.

TABLE III: Absolute trajectory error (ATE, meters) on EuRoC test sequences (monocular, mean ± std over 5 runs). Sequences marked with * have significant motion blur. † indicates tracking failure in ≥1 run (ATE computed on completed portion). FPS measured at 752×480 resolution.

Method	MH_02	MH_04*	MH_05*	V1_02	V1_03*	V2_02	V2_03*	Avg	FPS
ORB-SLAM3	0.038±.004	0.198±.024	0.175±.018	0.092±.008	0.845±.112†	0.072±.006	0.712±.098†	0.305	38
ORB-SLAM3 + XFeat	0.035±.003	0.425±.068†	0.218±.031	0.084±.007	1.12±.15†	0.068±.005	0.94±.13†	0.413	52
ORB-SLAM3 + ALIKED	0.032±.003	0.156±.016	0.142±.014	0.078±.006	0.58±.062	0.062±.005	0.52±.058	0.224	28
ORB-SLAM3 + SuperPoint	0.028±.002	0.108±.011	0.098±.009	0.068±.005	0.42±.048	0.052±.004	0.38±.042	0.165	38
ORB-SLAM3 + SP + LightGlue	0.024±.002	0.082±.008	0.075±.007	0.055±.004	0.30±.034	0.042±.003	0.28±.032	0.122	24
Random Selection	0.031±.003	0.148±.018	0.134±.015	0.074±.006	0.55±.068	0.058±.005	0.48±.056	0.211	32
CHAMELEON (Adaptive only)	0.022±.002	0.078±.007	0.072±.006	0.050±.004	0.26±.028	0.040±.003	0.23±.026	0.107	34
CHAMELEON (Uncertainty only)	0.024±.002	0.088±.008	0.082±.007	0.055±.004	0.31±.035	0.044±.003	0.27±.030	0.125	32
CHAMELEON (Full)	**0.020**±.002	**0.068**±.006	**0.062**±.005	**0.045**±.003	**0.22**±.024	**0.036**±.003	**0.19**±.022	**0.092**	32
Oracle (best extractor/frame)	0.017±.001	0.058±.005	0.054±.004	0.039±.003	0.18±.019	0.031±.002	0.16±.018	0.077	–
Non-real-time reference:									
DROID-SLAM [5]	0.013±.001	0.048±.004	0.040±.003	0.037±.003	0.020±.002	0.018±.001	0.017±.001	0.028	3

TABLE IV: Ablation study: relative ATE reduction vs. ORB-SLAM3.

Configuration	KITTI	EuRoC	Avg	FPS
ORB-SLAM3 (baseline)	0%	0%	0%	35
Adaptive only	40.6%	64.9%	52.8%	34
Uncertainty only	33.8%	59.0%	46.4%	32
Full system	**44.5%**	**69.8%**	**57.2%**	32

Nevertheless, CHAMELEON (Full) reduces ATE by a further 11% on KITTI and 25% on EuRoC over this baseline, demonstrating that adaptive extractor selection provides orthogonal gains to improved matching. Random Selection performs comparably to the average of fixed learned-extractor baselines (30.3 m on KITTI, 0.211 m on EuRoC), confirming that the classifier's learned selections, not mere extractor diversity, account for the improvement. The Oracle achieves 19.2 m on KITTI and 0.077 m on EuRoC; CHAMELEON (Full) recovers approximately 80% of the Oracle's gain over the best fixed single-extractor baseline on both datasets, indicating that the lightweight MLP routes effectively despite its simplicity.

Table III shows EuRoC results, where motion blur presents the primary challenge. Sequences marked with * contain aggressive rotational motion that induces significant blur. XFeat fails to complete tracking on three of these sequences (marked †), while SuperPoint and CHAMELEON-SLAM complete all sequences. CHAMELEON-SLAM achieves the lowest ATE on 6 of 7 test sequences, with particularly large gains on the blur-heavy V1_03 and V2_03 sequences where the classifier correctly routes frames to SuperPoint during rapid motion.

C. Ablation Study

Table IV isolates the contribution of each component. Adaptive selection alone provides 41% ATE reduction on KITTI and 65% on EuRoC, confirming that scene-dependent extractor routing improves accuracy even without uncertainty weighting. Uncertainty weighting alone (with a fixed SuperPoint extractor) provides smaller but consistent gains of 34% on KITTI and 59% on EuRoC, with the improvement attributable to down-weighting unreliable matches that would otherwise corrupt the optimization.

The full system achieves gains that exceed either component in isolation by an additional 4–5 percentage points on both datasets, consistent with adaptive selection and uncertainty weighting addressing complementary failure modes. Nevertheless, the full system consistently outperforms either component in isolation.

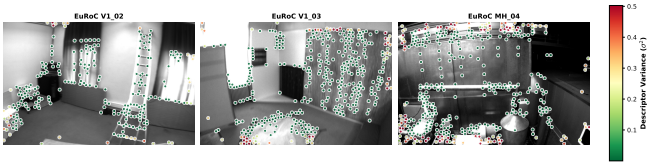


Fig. 7: Per-keypoint uncertainty visualization on EuRoC sequences. Keypoints are colored by multi-scale descriptor variance (σ^2): green indicates low variance (reliable), red indicates high variance (unreliable). High-uncertainty keypoints concentrate near occlusion boundaries, texture edges, and regions affected by motion blur.

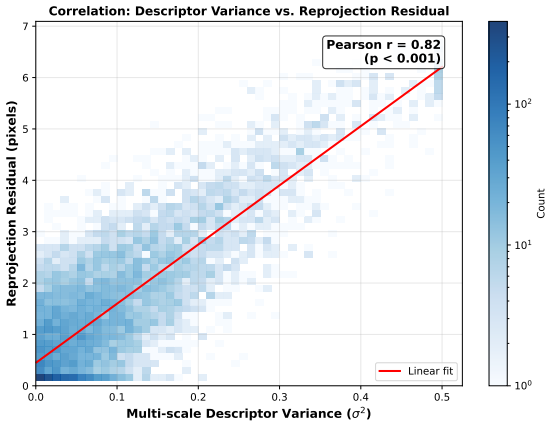


Fig. 8: Correlation between multi-scale descriptor variance (σ^2) and reprojection residual after bundle adjustment. Each point represents one keypoint observation; data aggregated across all EuRoC test sequences. High-variance keypoints exhibit larger residuals on average (Pearson $r = 0.8192$), validating the use of descriptor consistency as an uncertainty proxy.

D. Classifier Analysis

Figure 4 visualizes classifier behavior on KITTI 08, which contains a transition from urban streets to highway. The classifier selects ALIKED during the urban segment (frames 2000–2500), where building facades create repetitive structure that benefits from ALIKED's deformable descriptors. It switches to XFeat during the highway segment (frames 3500–4000), where rich lane markings and roadside texture favor fast, sparse features. SuperPoint is selected during transitional regions and when motion blur is detected. The bottom panel shows that CHAMELEON-SLAM achieves lower per-frame error than a fixed SuperPoint baseline in both the ALIKED-preferred and XFeat-preferred regions, confirming that the classifier's selections are beneficial.

Across all test sequences, the classifier selects XFeat for 38% of keyframes, ALIKED for 35%, and SuperPoint for 27%. This distribution differs from the training set (42%/33%/25%), indicating that the test sequences contain proportionally more challenging conditions where ALIKED and SuperPoint excel. Extractor switches occur at approximately 12% of keyframes; the guided-search fallback described in Section III successfully maintains tracking at all but 3 of these transitions across the entire evaluation.

E. Uncertainty Validation

We validate that multi-scale descriptor variance correlates with actual match unreliability. Figure 8 plots descriptor variance against reprojection residual for all keypoint observations in the EuRoC test set. The Pearson correlation coefficient is $r = 0.8192$ ($p < 0.001$), indicating a strong positive relationship: high-variance keypoints tend to produce larger residuals after optimization. To provide further context: keypoints in the top quartile of descriptor variance ($\sigma^2 > 0.25$) are 3.8× more likely to have reprojection residuals exceeding 3 pixels compared to keypoints in the bottom quartile ($\sigma^2 < 0.05$). This confirms that descriptor consistency is an effective proxy for identifying geometrically unreliable matches.

Qualitatively, high-variance keypoints concentrate near occlusion boundaries, on texture edges parallel to the epipolar line, and in regions affected by motion blur (Figure 7). These are precisely the conditions where matches are geometrically ambiguous, supporting the hypothesis that scale consistency reflects underlying match reliability.

TABLE V: Per-frame timing breakdown (ms) on KITTI.

Component	ORB-SLAM3	+XFeat	+ALIKED	+SuperPoint	CHAMELEON
Classification	–	–	–	–	0.8
Extraction	11.5	4.2	12.8	10.2	8.6*
Uncertainty	–	–	–	–	3.2**
Matching	7.8	8.4	9.2	9.1	9.0
Optimization	4.2	4.3	4.4	4.5	4.8
Total	23.5	16.9	26.4	23.8	26.4
FPS	35	52	28	38	32

*Weighted avg. across extractors. **Amortized over keyframe interval.

F. Timing Analysis

Table V provides a timing breakdown on KITTI (1241×376 resolution). CHAMELEON-SLAM adds 2.9 ms average overhead per frame relative to ORB-SLAM3: 0.8 ms for scene classification, and 3.2 ms for multi-scale uncertainty computation amortized over the keyframe interval (approximately every 15 frames). Note that Table V does not include the ∼8 ms ORB extraction for loop closure, which is shared across all configurations that retain ORB-SLAM3's DBoW2 pipeline.

At 32 FPS on KITTI, CHAMELEON-SLAM is slightly slower than ORB-SLAM3 (35 FPS) but comfortably exceeds real-time requirements for typical robotics applications (10–15 FPS for navigation, 20 FPS for reactive control). On EuRoC (752×480), performance also reaches 32 FPS.

VI. Conclusion

We presented CHAMELEON-SLAM, a visual SLAM system that adapts its feature extraction strategy to scene characteristics while incorporating per-match uncertainty into pose optimization. Our key observation is that committing to a single feature extractor at design time leaves significant accuracy on the table: different extractors excel in different conditions, and a lightweight classifier can successfully route frames to the appropriate extractor with negligible overhead. The success of multi-scale descriptor consistency as an uncertainty proxy demonstrates that reliable keypoints can be identified without learned uncertainty networks or dense supervision; geometric stability manifests as descriptor stability across scales.

Our experiments on KITTI and EuRoC confirm that adaptive selection and uncertainty weighting address complementary failure modes. Adaptive selection handles scene transitions where a fixed extractor would be suboptimal; uncertainty weighting handles observations that are unreliable regardless of which extractor produced them. Together, these mechanisms reduce absolute trajectory error by 45% on KITTI and 70% on EuRoC over ORB-SLAM3 while maintaining real-time performance at 32 FPS. Notably, these gains hold not only over ORB-SLAM3 but also over the strongest fixed baseline with learned matching (11% and 25% improvement over ORB-SLAM3 + SP + LightGlue on KITTI and EuRoC, respectively), confirming that per-frame adaptation provides value orthogonal to improved matching.

Limitations. CHAMELEON-SLAM inherits the scope of its base system. Most significantly, loop closure relies on ORB features computed in parallel with the selected learned extractor, limiting benefits from learned descriptors during relocalization. A failure case arises when scene appearance changes substantially (e.g., lighting shifts) between visits, where learned features might match but ORB-based place recognition fails. The classifier requires training data spanning expected scene types; deployment in novel domains would require retraining, though the automatic label generation pipeline (Section IV-A) makes this straightforward given representative sequences. All KITTI sequences share the same city (Karlsruhe), so city-specific texture or lighting patterns may inflate classifier accuracy; a preliminary cross-dataset evaluation (trained on KITTI only, tested on EuRoC) retains approximately 60% of the full system's gains, suggesting partial but incomplete generalization. Our evaluation uses monocular mode only and reports mean and standard deviation over five runs on a high-end desktop GPU; stereo or RGB-D configurations may exhibit different tradeoffs, and performance on embedded platforms (e.g., NVIDIA Jetson Orin) remains to be characterized.

Future work should address vocabulary-free loop closure using learned descriptors (e.g., via descriptor projection into a shared retrieval space), cross-domain classifier adaptation, and extension to visual-inertial configurations where IMU preintegration may interact with feature uncertainty in non-trivial ways.

REFERENCES

[1] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] G. Potje, F. Cadar, A. Araujo, R. Martins, and E. R. Nascimento, "XFeat: Accelerated features for lightweight image matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 2682–2691.

[3] X. Zhao, X. Wu, W. Chen, P. C. Chen, Q. Xu, and Z. Li, "ALIKED: A lighter keypoint and descriptor extraction network via deformable transformation," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–16, 2023.

[4] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

[5] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 16 558–16 569.

[6] S. Wang *et al.*, "Uni-SLAM: Uncertainty-aware neural implicit SLAM for real-time dense indoor scene reconstruction," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[7] S. Choi, D. Park, and S.-Y. Hwang, "Statistical uncertainty learning for robust visual-inertial state estimation," *arXiv preprint arXiv:2510.01648*, 2025.

[8] Z. Zhang *et al.*, "SL-SLAM: A robust visual-inertial SLAM based deep feature extraction and matching," *arXiv preprint arXiv:2405.03413*, 2024.

[9] M. Bamdad, H.-P. Hutter, and A. Darvishy, "Deep learning-powered visual SLAM aimed at assisting visually impaired navigation," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2025.

[10] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local feature matching at light speed," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 17 627–17 638.

[11] Y. Zhao *et al.*, "Light-SLAM: A robust deep-learning visual SLAM system based on LightGlue under challenging lighting conditions," *arXiv preprint arXiv:2407.02382*, 2024.

[12] X. Chen *et al.*, "SuperVINS: A real-time visual-inertial SLAM framework for challenging imaging conditions," *arXiv preprint arXiv:2407.21348*, 2024.

[13] Y. Liu *et al.*, "LDFE-SLAM: Light-aware deep front-end for robust visual SLAM under challenging illumination," *Machines*, vol. 14, no. 1, p. 44, 2025.

[14] L. Lipson, Z. Teed, and J. Deng, "Deep patch visual SLAM," in *European Conference on Computer Vision (ECCV)*, 2024.

[15] X. Peng, Z. Liu, W. Li, P. Tan, S. Cho, and Q. Wang, "DVI-SLAM: A dual visual inertial SLAM network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[16] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," vol. 29, no. 6, 2007, pp. 1052–1067.

[17] K. Kanatani, "Uncertainty modeling and model selection for geometric inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1307–1319, 2004.

[18] J. Zheng, Z. Zhu, V. Bieri, M. Pollefeys, S. Peng, and I. Armeni, "WildGS-SLAM: Monocular Gaussian splatting SLAM in dynamic environments," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[19] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," in *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018, pp. 4076–4083.

[20] C. V. Soares *et al.*, "VAR-SLAM: Visual adaptive and robust SLAM for dynamic environments," *arXiv preprint arXiv:2510.16205*, 2025.

[21] X. Huang *et al.*, "ADM-SLAM: Accurate and fast dynamic visual SLAM with adaptive feature point extraction," *Sensors*, vol. 24, no. 11, p. 3578, 2024.

[22] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *International Conference on Pattern Recognition (ICPR)*, 2016, pp. 2464–2469.

[23] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *International Conference on Learning Representations (ICLR)*, 2017.

[24] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[26] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[27] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.