Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Efficient Feature-Free Initialization for Monocular Visual-Inertial Systems Using a Feed-Forward 3D Model

Yuantai Zhang[1] Jiaqi Yang[1,2] Huajian Zeng[1] Changhao Chen[3] Haoang Li[3]
Liang Li[4] Dezhen Song[1] Xingxing Zuo[1†]

[1]MBZUAI [2]ShanghaiTech [3]HKUST (GZ) [4]Zhejiang University
[†]Corresponding author

***Abstract*—Fast and reliable initialization is critical for monocular visual–inertial navigation systems (VINS), as it establishes the starting conditions for subsequent state estimation. Despite steady progress, most existing methods heavily rely on visual feature correspondences and require 3-4 seconds of sensory data for successful initialization, which limits their applicability and efficiency. With the advent of feed-forward 3D models that can directly predict point clouds from images, we revisit the visual–inertial initialization problem from a concise perspective. In this work, we propose a feature-free initialization framework that leverages up-to-scale point clouds predicted by a feed-forward 3D model, thereby obviating the need for visual feature tracking and estimation. This design substantially reduces system complexity and improves the reliability of initialization. Experiments on public datasets demonstrate that the proposed feature-free initialization method achieves the highest success rate, exceeding 90%, and significantly reduces the data duration required for successful initialization, typically to under 1.2 s. We further validate our method on a self-collected dataset covering various indoor and outdoor scenarios, demonstrating robust performance, particularly in visually degraded environments where existing methods often fail. The code and dataset are available at `github.com/Yuantai-Z/FF-VIO-Init`.**

I. Introduction

Monocular visual-inertial navigation systems (VINS), which combine a single camera with an inertial measurement unit (IMU), enable robust and accurate pose estimation by leveraging the complementary characteristics of visual and inertial sensing. With continued advances in estimation theory, optimization techniques, and sensor fusion algorithms [1, 2, 3, 4, 5], VINS has become a fundamental component in a wide range of applications, including augmented and virtual reality (AR/VR) [6, 7], robotics perception and navigation [8, 9, 10, 11]. A critical prerequisite for successful monocular VINS is reliable system initialization. Specifically, initialization aims to recover the initial linear velocity and gravity, as well as the metric scale that is inherently unobservable in visual-only pose estimation. Inaccurate initialization may introduce large estimation errors and may even lead to rapid divergence of the state estimator. At the same time, fast initialization is equally important, as it significantly expands the practical applicability of VINS in real-world scenarios such as on-device AR experiences, agile robotic deployment, and emergency navigation, where prolonged initialization periods are undesirable or infeasible.

Motivated by these requirements, numerous methods have been proposed to achieve fast and accurate initialization for monocular VINS [12, 13, 14]. However, most existing approaches implicitly or explicitly rely on visual feature tracking across image views and accurate initialization of visual features (i.e., landmarks) from tracked 2D feature observations to establish geometric constraints between different camera frames. In practice, challenging conditions such as textureless environments, rapid camera motion, low-parallax trajectories, motion blur, poor illumination, and sensor noise frequently lead to poor feature tracking and unreliable landmark initialization. Under these conditions, existing monocular VINS initialization methods often become ill-conditioned, resulting in degraded accuracy, delayed convergence, or complete failure.

Recent advances in deep learning have catalyzed a paradigm shift in 3D geometric estimation. Feed-forward 3D models have demonstrated remarkable spatial understanding capabilities, enabling end-to-end prediction of scene-level 3D point clouds directly from input images [15, 16, 17]. However, such models typically suffer from inherent scale ambiguity. In this work, we explore the potential of leveraging feed-forward 3D models for fast and reliable monocular VINS initialization, while recovering the metric scale of the underlying 3D scene.

We propose a novel feature-free formulation for monocular VINS initialization that exploits geometric constraints from point clouds predicted by a feed-forward 3D model. Our approach bypasses nuisance visual feature extraction and initialization. The initialization pipeline consists of two stages: a closed-form linear system for an initial guess and a nonlinear visual-inertial bundle adjustment (VI-BA) for refinement. Our method not only simplifies the monocular VINS initialization problem into an extremely concise formulation, but also substantially improves its efficiency and robustness. The main contributions are:

- We propose an efficient monocular VINS initialization paradigm that leverages point cloud predictions from a feed-forward 3D model. To the best of our knowledge, this is the first work to integrate feed-forward 3D foundation models into monocular VINS initialization.

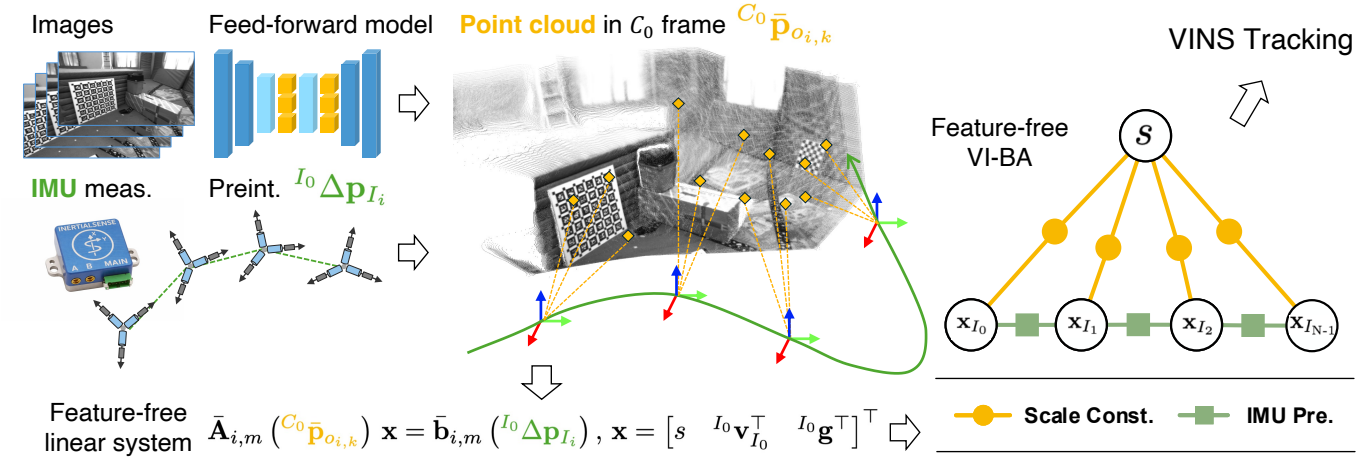

Fig. 1: System overview. A feed-forward 3D model predicts up-to-scale point clouds (yellow) from input images, while IMU measurements are preintegrated (green). These are combined into a feature-free closed-form linear system $\bar{\mathbf{A}}(\cdot)\mathbf{x} = \bar{\mathbf{b}}(\cdot)$, where $\bar{\mathbf{A}}(\cdot)$ incorporates up-to-scale point cloud geometry and $\bar{\mathbf{b}}(\cdot)$ encodes metric-scale IMU information. The closed-form solution is refined by a compact feature-free bundle adjustment involving only scale and IMU states, finally bootstrapping VINS tracking.

- We derive a feature-free, closed-form linear system that naturally eliminates visual feature association while reliably estimating metric scale, initial velocity, and gravity.
- We formulate two variants of VI-BA leveraging constraints from the predicted point clouds: a feature-free formulation that achieves maximum efficiency and success rate, and a scale-constrained formulation that preserves feature correspondences for higher accuracy.
- We conduct extensive experiments on both public benchmarks and a self-collected dataset, achieving over 90% success rate while reducing the required data duration across various challenging scenarios.

II. Related Work

A. Monocular Visual-Inertial Initialization

Early initialization approaches often relied on static-motion assumptions, but such conditions are frequently violated in practice. Consequently, most existing work focuses on dynamic initialization, which can be broadly categorized into loosely coupled and tightly coupled (closed-form) approaches. Loosely-coupled approaches rely on a successful visual odometry (VO) and align the up-to-scale camera trajectory with inertial constraints to recover metric scale. The initial states are recovered through different steps of linear least-squares [18, 19]. In contrast, closed-form solutions, pioneered by [12, 20], tightly couple visual constraints and inertial measurements by jointly recovering feature positions together with the initial states. This approach is later extended by jointly estimating the gyroscope bias [21], along with additional observability tests to improve robustness [22]. More recently, DRT [23] and sqrtVINS [24] exploited the coplanarity constraint of epipolar geometry to bypass the estimation for 3D features. However, all aforementioned methods fundamentally depend on reliable feature extraction and tracking, making them inherently sensitive to degeneracy and low-parallax scenarios.

B. Feed-Forward 3D Models

Recent advances in 3D geometric perception have driven a paradigm shift from traditional iterative structure-from-motion (SfM) pipelines [25] to feed-forward models that directly infer scene geometry from images. DUSt3R [15] and MASt3R [26] introduce end-to-end formulations that predict point clouds from image pairs, while VGGT [16] further generalizes this paradigm to multi-view inputs with strong cross-view consistency. Follow-up efforts extend feed-forward 3D models toward longer sequence settings using recurrent network architecture [17, 27], spatial memory [28], and SLAM pipeline [29, 30]. Despite these advances, feed-forward 3D models typically recover geometry only up to an unknown similarity transformation [15, 16]. Several efforts address this limitation by predicting metric depth [31] or incorporating explicit scale regression with optional geometric priors [32, 33]. However, in large-scale or long-horizon scenarios, visual metric-scale estimation remains prone to scale drift and instability [29].

C. Learning-aided Visual-Inertial Initialization

Deep learning has been widely integrated in visual SLAM and VINS to enhance robustness and accuracy. These approaches either replace core SLAM components with learning-based counterparts [34, 35, 36], or augment traditional pipelines with learned priors [37, 38]. Our work falls into the latter category. In particular, the emergence of 3D foundation models has enabled SLAM systems to leverage their strong spatial prediction. MASt3R-SLAM [29] and VGGT-SLAM [30] leverage dense predictions for tracking and mapping, later extending to multi-sensor settings with IMU integration [39]. However, these systems assume a valid initial state is already available. To the best of our knowledge, exploiting feed-forward 3D geometry for visual–inertial initialization has not yet been explored. The works most closely

related to ours are [40, 41], which leverage learned single-view depth for monocular VINS initialization. Zhou et al. [40] incorporate depth predictions as additional measurements in visual-inertial bundle adjustment, while Merrill et al. [41] derive a closed-form solution that eliminates explicit 3D feature states. These works demonstrate that learned geometric priors can improve initialization robustness, yet they still rely on feature correspondences.

III. Visual-Inertial Initialization

This section presents the measurement models for IMU and camera, and revisits the standard monocular visual-inertial initialization pipeline consisting of a closed-form linear system and nonlinear refinement.

A. IMU and Camera Models

An IMU measures angular velocity $\boldsymbol{\omega}$ and acceleration $\boldsymbol{a}$, modeled as:

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{n}_\omega \tag{1}$$

$$\tilde{\boldsymbol{a}} = \boldsymbol{a} + {}^{I}_{G}\mathbf{R}\,{}^{G}\mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \tag{2}$$

where $\tilde{\boldsymbol{\omega}}$ and $\tilde{\boldsymbol{a}}$ are the measured angular velocity and acceleration, respectively. ${}^{I}_{G}\mathbf{R}$ is the rotation to transform a point in gravity-aligned global frame $\{\boldsymbol{G}\}$ to IMU frame $\{\boldsymbol{I}\}$, ${}^{G}\mathbf{g} = [0, 0, g]^\top$ is the gravity vector in global frame with $g = 9.81\,\mathrm{m/s^2}$. Based on classical inertial navigation system (INS) kinematic model [42], the discrete-time propagation from frame i to $i+1$ is:

$${}^{G}_{I_{i+1}}\mathbf{R} = {}^{G}_{I_i}\mathbf{R}\,{}^{I_i}_{I_{i+1}}\Delta\mathbf{R} \tag{3}$$

$${}^{G}\mathbf{v}_{I_{i+1}} = {}^{G}\mathbf{v}_{I_i} - {}^{G}\mathbf{g}\Delta t + {}^{I_i}_{G}\mathbf{R}^\top {}^{I_i}\Delta\mathbf{v}_{I_{i+1}} \tag{4}$$

$${}^{G}\mathbf{p}_{I_{i+1}} = {}^{G}\mathbf{p}_{I_i} + {}^{G}\mathbf{v}_{I_i}\Delta t - \frac{1}{2}{}^{G}\mathbf{g}\Delta t^2 + {}^{I_i}_{G}\mathbf{R}^\top {}^{I_i}\Delta\mathbf{p}_{I_{i+1}} \tag{5}$$

where $\{{}^{G}_{I_i}\mathbf{R},\, {}^{G}\mathbf{p}_{I_i}\}$ denotes the IMU pose in $\{\boldsymbol{G}\}$ at timestamp i, together forming the rigid-body transformation ${}^{G}_{I_i}\mathbf{T} \in SE(3)$. $\mathbf{z}^{\mathcal{I}}_{i,i+1} = \{\Delta\mathbf{R}(\tilde{\boldsymbol{\omega}}), \Delta\mathbf{v}(\tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{a}}), \Delta\mathbf{p}(\tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{a}})\}$ represents the IMU preintegration measurements between consecutive frames [43, 44], where $\Delta\mathbf{R}$, $\Delta\mathbf{v}$, and $\Delta\mathbf{p}$ denote the preintegrated relative rotation, velocity, and position, respectively. In the closed-form system, we use the first IMU frame $\{\boldsymbol{I_0}\}$ as the global frame.

For the camera model, we consider a monocular pinhole camera rigidly attached and temporally synchronized to the IMU, with known camera intrinsics and extrinsics $\{{}^{C}_{I}\mathbf{R}, {}^{C}\mathbf{p}_I\}$. The observation of the m-th visual feature is denoted by $\mathbf{z}^{\mathcal{C}}_{i,m} \in \mathbb{R}^2$, and the measurement model can be written as:

$$\mathbf{z}^{\mathcal{C}}_{i,m} = \begin{bmatrix} u_{i,m} \\ v_{i,m} \end{bmatrix} = \pi\left({}^{C_i}\mathbf{p}_{f_m}\right) + \mathbf{n} \tag{6}$$

$${}^{C_i}\mathbf{p}_{f_m} = {}^{C}_{I}\mathbf{R}\,{}^{I_i}_{I_0}\mathbf{R}\left({}^{I_0}\mathbf{p}_{f_m} - {}^{I_0}\mathbf{p}_{I_i}\right) + {}^{C}\mathbf{p}_I \tag{7}$$

where $(u_{i,m}, v_{i,m})$ denotes the normalized coordinates of the m-th feature at time t_i, $\pi(\cdot)$ is the normalized projection function, $\{\boldsymbol{C}_i\}$ is the camera frame at time t_i, and ${}^{I_0}\mathbf{p}_{f_m}$ is the 3D position of the m-th feature expressed in frame $\{\boldsymbol{I}_0\}$. Eq. (7) can be rewritten as the following linear constraint [12]:

$$\begin{bmatrix} 1 & 0 & -u_{i,m} \\ 0 & 1 & -v_{i,m} \end{bmatrix}\left({}^{C}_{I}\mathbf{R}\,{}^{I_i}_{I_0}\mathbf{R}\left({}^{I_0}\mathbf{p}_{f_m} - {}^{I_0}\mathbf{p}_{I_i}\right) + {}^{C}\mathbf{p}_I\right) = 0 \tag{8}$$

B. VI Initialization

The minimal goal of VINS initialization is to recover the initial velocity ${}^{I_0}\mathbf{v}_{I_0}$ and gravity vector ${}^{I_0}\mathbf{g}$. Typically, two phases are involved: (i) a linear system that provides a coarse initial guess of the state; (ii) a nonlinear optimization that refines the estimates and outputs all necessary states to bootstrap the pose tracking of VINS systems.

1) Constrained Linear Least-Squares Formulation: By substituting Eq. (5) into Eq. (8) and stacking all visual measurements, we obtain a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. The state vector $\mathbf{x}$ contains all M feature positions, initial velocity, and gravity:

$$\mathbf{x} = \begin{bmatrix} {}^{I_0}\mathbf{p}^\top_{f_1} & \cdots & {}^{I_0}\mathbf{p}^\top_{f_M} & {}^{I_0}\mathbf{v}^\top_{I_0} & {}^{I_0}\mathbf{g}^\top \end{bmatrix}^\top \tag{9}$$

Each measurement $\mathbf{z}^{\mathcal{C}}_{i,m}$ contributes two rows to $\mathbf{A}$ and $\mathbf{b}$:

$$\begin{aligned} \mathbf{A}_{i,m} &= \mathbf{H}_{i,m}\begin{bmatrix} \mathbf{E}_m & -\Delta t\,\mathbf{I}_3 & \frac{1}{2}\Delta t^2\,\mathbf{I}_3 \end{bmatrix} \\ \mathbf{b}_{i,m} &= \mathbf{H}_{i,m}\left({}^{I_0}\Delta\mathbf{p}_{I_i} - {}^{I_0}_{I_i}\mathbf{R}\,{}^{I}_{C}\mathbf{R}\,{}^{C}\mathbf{p}_I\right) \end{aligned} \tag{10}$$

where $\mathbf{E}_m = [\mathbf{0}_{3\times3(m-1)}\ \mathbf{I}_3\ \mathbf{0}_{3\times3(M-m)}]$ selects the m-th feature position from $\mathbf{x}$, and $\mathbf{H}_{i,m}$ is defined as:

$$\mathbf{H}_{i,m} = \begin{bmatrix} 1 & 0 & -u_{i,m} \\ 0 & 1 & -v_{i,m} \end{bmatrix}{}^{C}_{I}\mathbf{R}\,{}^{I_i}_{I_0}\mathbf{R} \tag{11}$$

The linear system can be solved as a constrained least-squares problem subject to $\|{}^{I_0}\mathbf{g}\|_2^2 = g^2$ [12]. Once the gravity vector ${}^{I_0}\mathbf{g}$ is estimated, the gravity-aligned rotation ${}^{G}_{I_0}\mathbf{R}$ can be recovered via Gram-Schmidt orthogonalization.

2) Nonlinear Refinement: The linear solution provides a coarse estimate that may be affected by noise. To properly account for measurement uncertainties, it is usually refined through nonlinear optimization. Given N keyframes in the initialization window, the full state vector is defined as:

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}^\top_{I_0} & \cdots & \mathbf{x}^\top_{I_{N-1}} & {}^{G}\mathbf{p}^\top_{f_1} & \cdots & {}^{G}\mathbf{p}^\top_{f_M} \end{bmatrix}^\top \tag{12}$$

$$\mathbf{x}_{I_i} = \begin{bmatrix} {}^{I_i}_{G}\mathbf{q}^\top & {}^{G}\mathbf{p}^\top_{I_i} & {}^{G}\mathbf{v}^\top_{I_i} & \mathbf{b}^\top_\omega & \mathbf{b}^\top_a \end{bmatrix}^\top \tag{13}$$

where $\mathbf{x}_{I_i}$ denotes the 15-dimensional IMU state that contains orientation, velocity, position, and biases of the i-th frame.

The optimization minimizes a cost function comprising three terms, as illustrated by the factor graph in Fig. 2(a):

$$\begin{aligned} \mathcal{X}^* = \arg\min_{\mathcal{X}} \Bigg\{ &\sum_{i\in\mathcal{I}} \left\|\mathbf{r}_{\mathcal{I}}(\mathbf{z}^{\mathcal{I}}_{i,i+1}, \mathbf{x}_{I_i}, \mathbf{x}_{I_{i+1}})\right\|^2_{\boldsymbol{\Sigma}_{\mathcal{I}}} \\ &+ \sum_{(i,m)\in\mathcal{C}} \left\|\mathbf{r}_{\mathcal{C}}(\mathbf{z}^{\mathcal{C}}_{i,m}, \mathbf{x}_{I_i}, \mathbf{p}_{f_m})\right\|^2_{\boldsymbol{\Sigma}_{\mathcal{C}}} + \left\|\mathbf{r}_{\mathrm{pr}}(\mathbf{z}^{\mathrm{p}}, \mathbf{x}_{I_0})\right\|^2_{\boldsymbol{\Sigma}_{\mathrm{pr}}} \Bigg\} \end{aligned} \tag{14}$$

where $\mathbf{r}_{\mathcal{C}}$ is the reprojection residual as in Eq. (6), $\mathbf{r}_{\mathcal{I}}$ is the IMU preintegration residual [43, 44], and $\mathbf{r}_{\mathrm{pr}}$ is the prior residual that constrains the state to the linearization point. $\boldsymbol{\Sigma}_{\mathcal{I}}$ is propagated from IMU noise during preintegration,

$\mathbf{\Sigma}_{\mathcal{C}}$ is set according to pixel measurement noise, and $\mathbf{\Sigma}_{\text{pr}}$ is manually configured with small variance on unobservable states to stabilize optimization [45].

IV. Feature-Free Initialization

The classical pipeline described above suffers from two fundamental limitations: (i) feature extraction and tracking are fragile under degraded scenes; (ii) jointly estimating a large number of feature positions inflates the state dimension and degrades numerical conditioning. We address both issues by replacing feature correspondences with up-to-scale point clouds from a feed-forward 3D model as illustrated in Fig. 1, yielding a feature-free initialization framework with a compact state space and improved efficiency.

A. Feed-Forward 3D Models

We incorporate feed-forward 3D foundation models within the classical VINS initialization framework for faster and more robust state recovery. Previous works leverage monocular depth prediction networks [40, 41], which provide 3D geometry expressed in the local camera frame of each image. These methods implicitly rely on feature correspondences to establish geometric constraints across multiple frames, limiting their applicability. Crucially, 3D foundation models produce point clouds expressed in a consistent reference camera frame $\{\boldsymbol{C}_0\}$. This property allows us to bypass the need for the whole feature extraction and tracking process, and to directly use geometry from the point cloud for state estimation.

We employ the pretrained VGGT [16] as our feed-forward 3D model $\mathcal{F}$ to infer point clouds. Given N input images $\{\mathcal{I}_i\}_{i=0}^{N-1}$ of size $H \times W$, the model outputs:

$$\mathcal{F}\left(\{\mathcal{I}_i\}_{i=0}^{N-1}\right) = \left\{{}^{C_0}\bar{\mathbf{p}}_{o_{i,k}},\, c_{i,k}\right\}_{i\in N,\, k\in H\times W}^{H\times W\times N} \tag{15}$$

where ${}^{C_0}\bar{\mathbf{p}}_{o_{i,k}} \in \mathbb{R}^3$ is the up-to-scale 3D position of the k-th point from image $\mathcal{I}_i$, with subscript o denoting per-pixel point cloud rather than tracked features. $c_{i,k} \in [1, \infty]$ is the confidence score indicating the prediction uncertainty. To keep the problem tractable, we do not use dense prediction. Instead, we first filter out low-confidence predictions according to $c_{i,k}$. We then partition each image into uniform patches and sample a fixed number K of high-confidence points per frame.

B. Robust Feature-Free Closed-Form Solution

In SLAM systems, 3D points are typically expressed by 3D coordinates, depth, or inverse depth [46]. In this work, given the predicted up-to-scale point cloud $\left\{{}^{C_0}\bar{\mathbf{p}}_{o_{i,k}}\right\}$, we represent 3D points by a scale parameter s as:

$${}^{I_0}\mathbf{p}_{o_{i,k}} = s\,{}^{I}_{C}\mathbf{R}\,{}^{C_0}\bar{\mathbf{p}}_{o_{i,k}} + {}^{I}\mathbf{p}_{C} \tag{16}$$

Any pixel of $\{\mathcal{I}_i\}_{i=0}^{N-1}$ can be mapped to a 3D point ${}^{I_0}\mathbf{p}_{o_{i,k}}$, and all points share a single unknown scale factor s. Unlike traditional methods that estimate M features and require a $3M$-dimensional feature state (see Eq. (9)), we reduce the feature-related state to a single scalar, significantly lowering the problem dimensionality. Substituting Eq. (16) into Eq. (7) and stacking all measurements yields a linear system $\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{b}}$, where the state vector contains only scale, initial velocity, and gravity:

$$\mathbf{x} = \begin{bmatrix} s & {}^{I_0}\mathbf{v}_{I_0}^{\top} & {}^{I_0}\mathbf{g}^{\top} \end{bmatrix}^{\top} \tag{17}$$

Each measurement $\mathbf{z}_{i,k}^{\mathcal{C}}$ contributes two rows to $\bar{\mathbf{A}}$ and $\bar{\mathbf{b}}$:

$$\begin{aligned} \bar{\mathbf{A}}_{i,k} &= \mathbf{H}_{i,k}\begin{bmatrix} {}^{I}_{C}\mathbf{R}\,{}^{C_0}\bar{\mathbf{p}}_{o_{i,k}} & -\Delta t\,\mathbf{I}_3 & \frac{1}{2}\Delta t^2\,\mathbf{I}_3 \end{bmatrix} \\ \bar{\mathbf{b}}_{i,k} &= \mathbf{H}_{i,k}\left({}^{I_0}\Delta\mathbf{p}_{I_i} - {}^{I_0}_{I_i}\mathbf{R}\,{}^{I}_{C}\mathbf{R}\,{}^{C}\mathbf{p}_{I} + {}^{I}_{C}\mathbf{R}\,{}^{C}\mathbf{p}_{I}\right) \end{aligned} \tag{18}$$

From the K sampled points from each of the N frames, we form a linear system with $\bar{\mathbf{A}} \in \mathbb{R}^{2KN\times 7}$ and $\bar{\mathbf{b}} \in \mathbb{R}^{2KN}$, and solve $\mathbf{x} \in \mathbb{R}^7$ via weighted constrained least-squares, where the predicted confidence $c_{i,k}$ is used to suppress unreliable predictions. Due to the fixed state dimension, RANSAC can be directly integrated into the solving process to reject outliers in the predicted point cloud and improve robustness [41]. The RANSAC procedure is provided in supplementary material.

C. Nonlinear Refinement

We propose two VI-BA formulations that leverage the predicted point cloud: (1) A Scale-Constrained (SC) formulation that augments traditional feature-based optimization with scale factors, preserving feature correspondences for higher accuracy; and (2) A Feature-Free (FF) formulation that eliminates feature states entirely, achieving maximum efficiency and robustness. Both formulations directly reuse the point cloud from the linear system, and thus directly benefit from the RANSAC inlier set.

1) Scale-Constrained Formulation: As shown in Fig. 2(b), additional scale parameters $\mathbf{s}$ are introduced into the feature-based BA states (see Eq. (12)). The VI-BA problem is formulated as:

$$\begin{aligned} \mathcal{X}^* = \arg\min_{\mathcal{X}} \Big\{ &\sum\|\mathbf{r}_{\mathcal{I}}\|^2_{\mathbf{\Sigma}_{\mathcal{I}}} + \sum\|\mathbf{r}_{\mathcal{C}}\|^2_{\mathbf{\Sigma}_{\mathcal{C}}} \\ &+ \sum\|\mathbf{r}_{\bar{\mathbf{p}}}\|^2_{\mathbf{\Sigma}_{\bar{\mathbf{p}}}} + \|\mathbf{r}_{\text{pr}}\|^2_{\mathbf{\Sigma}_{\text{pr}}} \Big\} \end{aligned} \tag{19}$$

where $\mathbf{r}_{\bar{\mathbf{p}}}$ is the scale residual that connects multiple features to the shared scale parameter s, providing additional geometric constraints from the predicted point cloud:

$$\mathbf{r}_{\bar{\mathbf{p}}}({}^{C_0}\bar{\mathbf{p}}_{f_m}, {}^{G}\mathbf{p}_{f_m}, s) = {}^{G}\mathbf{p}_{f_m} - (s\,{}^{G}_{C_0}\mathbf{R}\,{}^{C_0}\bar{\mathbf{p}}_{f_m} + {}^{G}\mathbf{p}_{C_0}) \tag{20}$$

Here ${}^{C_0}\bar{\mathbf{p}}_{f_m}$ is obtained by bilinearly interpolating $\{{}^{C_0}\bar{\mathbf{p}}_{o_{i,k}}\}$ in each frame at the tracked feature location. The measurement covariance $\mathbf{\Sigma}_{\bar{\mathbf{p}}}$ is determined by the predicted confidence score.

2) Feature-Free Formulation: In the second approach, we take a step further to completely eliminate features by estimating 3D points solely through the scale factor s, as shown in Fig. 2(c). The state vector of feature-free nonlinear optimization is defined as:

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}_{I_0}^{\top} & \cdots & \mathbf{x}_{I_{N-1}}^{\top} & \mathbf{s}^{\top} \end{bmatrix}^{\top} \tag{21}$$

The residuals contained in our feature-free VI-BA are the same as Eq. (14), however, the reprojection residual $\mathbf{r}_{\mathcal{C}}$ is

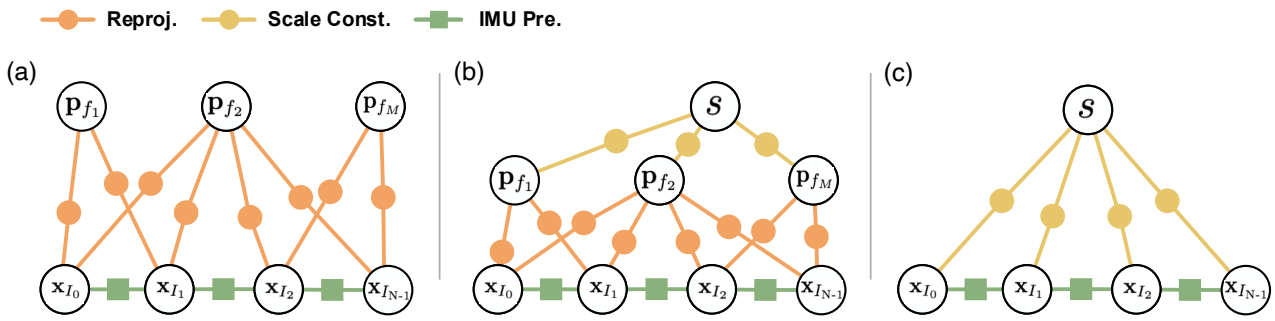

Fig. 2: Comparison of factor graphs for nonlinear refinement. (a) Traditional feature-based optimization jointly optimizes IMU states and 3D feature positions (see Eq. 14). (b) Our scale-constrained formulation augments feature-based optimization with shared scale factors (see Eq. (19)). (c) Our feature-free formulation eliminates features, only containing IMU states and scales.

parameterized by the scale factor s instead of the 3D point. For clarity, we provide the explicit form:

$$\mathbf{r}_{\mathcal{C}}({}^{C_0}\bar{\mathbf{p}}_{o_{i,k}}, \mathbf{x}_{I_i}, s) = \mathbf{z}^{\mathcal{C}}_{i,k} - \pi\big({}^{C}_{I}\mathbf{T}\,{}^{I_i}_{G}\mathbf{T}\,{}^{G}_{I_0}\mathbf{T}\,{}^{I}_{C}\mathbf{T}\,(s\,{}^{C_0}\bar{\mathbf{p}}_{o_{i,k}})\big) \tag{22}$$

The detailed expansion of $\mathbf{r}_{\mathcal{C}}$ together with its Jacobians w.r.t. the scale factor and IMU states is provided in Sec. VII-C of the supplementary material. We emphasize that $\mathbf{z}^{\mathcal{C}}_{i,k}$ can be any pixel coordinates in image $\mathcal{I}_i$, not restricted to feature correspondences. The shared index k is used solely for bookkeeping of sampled points within each frame and does not imply data association across views. Since all points share the same scale factor s, this formulation significantly reduces the state dimension from $\mathcal{O}(3M)$ of Eq. (14) to $\mathcal{O}(1)$ for feature variables, allowing us to only estimate scales and IMU states.

By coupling this feature-free VI-BA (Eq. (22)) with the feature-free closed-form linear system (Eq. (18)), we obtain a fully feature-free VI initialization pipeline, denoted **Ours (FF)**, in contrast to the scale-constrained counterpart **Ours (SC)** that retains feature-based BA. This unified FF pipeline offers several advantages:

- **Efficiency.** The compact formulation leads to lower computational complexity and fast convergence.
- **Robustness.** Being feature-free, the pipeline is inherently immune to common visual issues such as feature noise, mismatches, and insufficient parallax.
- **Numerical stability.** The largely reduced state dimension leads to better-conditioned BA.

Remark: Directly optimizing scale can cause degeneracy under measurement noise, where the scale tends to collapse toward zero. We address this by reparameterizing the scale as [47]:

$$s = \varepsilon + \log(e^{\tilde{s}} + 1) \tag{23}$$

where $\varepsilon = 10^{-5}$. This softplus-based formulation ensures $s > 0$ while enabling unconstrained optimization of $\tilde{s}$.

Moreover, although theoretically a single scale parameter suffices for the entire scene, in practice, prediction errors may vary across different regions. Thus, in nonlinear refinement, we partition the point cloud into a 2D grid along the principal plane, and assign an independent scale parameter to each region to absorb local errors. To prevent discontinuities, neighboring regions are connected by a smoothness constraint:

$$\mathbf{r}_{\Delta \mathrm{s}}(s_j, s_l) = s_j - s_l \tag{24}$$

where s_j and s_l are scale parameters of adjacent regions. In our implementation, we use a 3×3 grid, resulting in only 9 scale parameters. The effect of regional scales is evaluated in the ablation study.

V. Experiments

We conduct extensive experiments on two popular public datasets: EuRoC MAV [48] and TUM-VI [49], along with our self-collected dataset. For statistical reliability, following [41], we divide each sequence into 10-second windows, perform initialization and evaluation in each window, and report the average results. To provide a comprehensive evaluation, we assess the initialization performance using the following metrics: (1) gravity direction error, (2) linear velocity error, and (3) absolute trajectory error (ATE) [50] over the entire 10-second window. Beyond these common metrics, we additionally report the following:

(1) **Total time** T_{tot}. This metric measures the total temporal span of sensory data consumed by continuous initialization attempts until a successful initialization is achieved.
(2) **Success rate (SR)**. We adopt stricter success criteria than those used in the compared baseline [41], requiring that: (i) the optimization converges, (ii) the state covariance is successfully recovered, and (iii) the position absolute trajectory error (ATE) of VINS tracking over the entire window remains below 0.5 m.
(3) **Scale error**. This quantifies the relative error of the estimated point cloud scale, where the ground-truth scale is computed as the ratio between the predicted translation displacement and the ground-truth displacement.

These additional metrics provide a practical assessment of the sensory data required for successful initialization and the suitability of the resulting estimates for subsequent tracking.

TABLE I: Evaluation of linear systems on EuRoC dataset: gravity error (°) ↓ / velocity error (m/s) ↓.

Algorithm	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03	Average
Dong-Si	2.42 / 0.43	10.29 / 0.75	7.54 / 0.79	2.53 / 0.33	7.41 / 0.58	7.11 / 0.63	6.22 / 0.59
Dep.(AB)	2.37 / 0.40	4.60 / 0.85	5.33 / 0.85	1.85 / 0.48	4.15 / 0.60	4.73 / 0.75	3.84 / 0.66
sqrtVINS	2.59 / 0.24	3.88 / 0.63	5.42 / 0.50	4.39 / 0.29	6.60 / 0.51	1.56 / 0.21	4.07 / **0.40**
DRT	2.28 / 0.29	4.10 / 0.60	6.68 / 0.52	1.91 / 0.21	3.83 / 0.31	3.09 / 0.51	3.65 / 0.41
Ours w/o RANSAC	2.05 / 0.37	3.12 / 0.82	6.37 / 0.74	2.02 / 0.35	4.42 / 0.63	3.47 / 0.54	3.58 / 0.58
Ours	2.27 / 0.40	3.17 / 0.81	6.11 / 0.70	2.01 / 0.35	3.99 / 0.47	3.71 / 0.65	**3.54** / 0.56

TABLE II: Overall performance on EuRoC dataset (Setting: 5 keyframes, 0.5 s window).

	Dong-Si	Dep.(AB)	Dep.(Const.)	sqrtVINS	DRT	Ours (SC)	Ours (FF)
Win. ATE ↓	1.47/0.024	1.25/**0.020**	**1.20**/0.026	**1.20**/0.022	1.52/0.031	**1.20/0.020**	3.53/0.034
Traj. ATE ↓	4.30/0.078	4.07/0.087	4.01/0.079	3.48/0.082	4.44/0.095	3.82/0.109	**3.14/0.073**
T_{tot} ↓	2.48	2.87	3.30	1.09	**1.07**	1.26	1.19
SR ↑	67.2	67.2	76.6	73.4	75.0	68.8	**81.2**

ATE: °/m T_{tot}: s SR: %.

A. Implementation Details

We compare against the following baselines:

(1) **Dong-Si**: The default initialization method of OpenVINS [12], combining Eq. (10) and Eq. (14).
(2) **Dep.(Const.)**: Our re-implementation of [40] leveraging learned monocular depth priors while using the Dong-Si method for providing the initial guess.
(3) **Dep.(AB)**: Our re-implementation of [41].
(4) **LC**: Our re-implementation of loosely-coupled method as described in [18] by aligning the estimated camera trajectory and preintegrated IMU trajectory, using Eq. (14) for refinement.
(5) **sqrtVINS**: The initialization module of [24].
(6) **DRT**: The rotation-translation-decoupled initialization method [23].

We denote our proposed method as **Ours (SC)**, which uses the proposed scale-constrained nonlinear refinement (see Eq. (19)), and **Ours (FF)**, which adopts the fully feature-free formulation (see Eq. (22)). Both methods share the same linear initialization (see Eq. (18)).

All methods are integrated into the open-source OpenVINS framework [3], and chi-square tests are applied to reject obviously erroneous solutions. For a fair comparison, the camera pose for **LC** and the depth for **Dep.(Const.)** and **Dep.(AB)** are also predicted by the VGGT model [16]. The VGGT network runs on an RTX 6000 Ada GPU, with inference taking approximately 0.24 seconds for five frames. As [40] and [41] are not open source, we reproduced these methods according to their respective papers. Note that for **Dep.(Const.)**, due to the scale normalization strategy employed by VGGT, we are unable to incorporate the shift constraint for the depth factor. Other details follow the original papers.

B. Evaluation on EuRoC Dataset

We use a setting of 5 keyframes over a 0.5 s window on the EuRoC MAV dataset [48], where the parallax is relatively small and IMU excitation is limited, yielding challenging initialization scenarios. **LC** is excluded from the comparison, as it fails to converge within such short windows.

Table I compares the linear systems in terms of gravity direction and linear velocity errors. The proposed feature-free linear system generally outperforms Dong-Si and Dep.(AB) and achieves the lowest gravity error overall, demonstrating the effectiveness of the proposed feature-free closed-form formulation. By eliminating 3D feature positions from the linear system, all methods except Dong-Si achieve greater robustness to outliers and consistently higher accuracy across sequences. However, the learning-based methods degrade on V1_03, where motion blur compromises the 3D model predictions. Incorporating RANSAC further yields a modest but consistent improvement, confirming its effectiveness in suppressing outliers.

The overall initialization performance is summarized in Table II. Win. ATE measures initialization accuracy within the short temporal window, while Traj. ATE evaluates the tracking error of subsequent VINS trajectory, serving as an indicator of initialization reliability. Accordingly, SR is determined based on Traj. ATE to filter out numerically converged but incorrect results. Ours (FF) achieves the highest success rate, enabling successful initialization and subsequent tracking in over 80% of windows while requiring less sensory data, with an average T_{tot} of only 1.2 s, demonstrating superior efficiency and reliability. Comparing the two proposed variants, Ours (SC) achieves the best Win. ATE, surpassing all baselines. Ours (FF), on the other hand, yields higher Win. ATE than Ours (SC), which is expected due to the absence of feature correspondences and thus weaker geometric constraints during optimization. Nevertheless, Ours (FF) yields the lowest Traj. ATE, suggesting that its initialization quality is sufficient to support accurate and stable VINS tracking.

C. Evaluation on TUM-VI Dataset

We further evaluate on six room sequences of the TUM-VI dataset [49], which contain ground truth from a motion capture system. Since VGGT can only accept undistorted images, we rectify the fisheye image and crop the field of view during the initialization to ensure the prediction quality. We also adopt a setting of 5 keyframes over a 0.5 s window in this dataset. Compared to EuRoC, TUM-VI provides richer motion excitation, leading to better initialization accuracy in general.

TABLE III: Evaluation of linear system on TUM-VI dataset: gravity error (°) ↓ / velocity error (m/s) ↓.

Algorithm	room1	room2	room3	room4	room5	room6	Average
Dong-Si	8.39 / 0.87	4.99 / 0.84	3.56 / 0.56	5.77 / 0.53	12.60 / 1.05	4.61 / 0.38	6.65 / 0.71
Dep.(AB)	6.72 / 0.67	2.75 / 0.72	3.14 / 0.68	5.07 / 0.55	4.33 / 0.37	1.78 / 0.38	3.96 / 0.56
Ours w/o RANSAC	3.35 / 0.32	3.23 / 0.49	2.92 / 0.55	2.60 / 0.33	3.84 / 0.26	1.95 / 0.41	2.99 / **0.39**
Ours	3.57 / 0.33	3.22 / 0.48	2.86 / 0.55	2.45 / 0.31	3.77 / 0.28	1.95 / 0.41	**2.97 / 0.39**

TABLE IV: Overall performance on TUM-VI dataset (Setting: 5 keyframes, 0.5 s window).

	Dong-Si	Dep.(AB)	Dep.(Const.)	Ours (SC)	Ours (FF)
Win. ATE ↓	1.14/0.024	1.11/**0.021**	1.14/0.031	**1.04**/0.023	3.22/0.067
Traj. ATE ↓	1.59/**0.022**	1.67/**0.022**	2.18/0.045	**1.40**/0.032	2.10/0.068
T_{tot} ↓	2.87	3.63	4.66	1.16	**1.10**
SR ↑	74.7	82.7	78.7	92.0	**94.7**

ATE: °/m T_{tot}: s SR: %.

As shown in Table III, the advantage of our linear system becomes more pronounced under stronger IMU excitation, achieving approximately 25% reduction in gravity direction error compared to Dep.(AB). For overall performance in Table IV, both variants achieve comparable accuracy with only 1.1 s data duration, while improving the SR to over 90%.

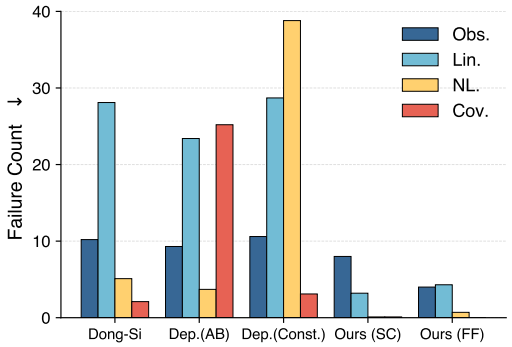

Fig. 3: Failure source distribution across methods. Obs.: insufficient observations. Lin.: rank-deficient linear system. NL.: divergent optimization. Cov.: covariance recovery failure.

To better understand total data consumption T_{tot}, we analyze the failure sources of each method and categorize them into four types as shown in Fig. 3. All baselines exhibit a large number of failures caused by insufficient observations and rank-deficient linear systems. Although Dep.(AB) eliminates the estimation of feature positions in the linear system, it still relies on feature correspondences implicitly and thus suffers from the same limitations. In addition, Dep.(AB) and Dep.(Const.) show increased failures in covariance recovery and nonlinear optimization respectively, stemming from the underlying feature-based formulation. Owing to a robust closed-form solution and reduced state dimension, Ours (FF) and Ours (SC) significantly reduce failures across all categories, thus achieving lowest T_{tot}. We also analyze the per-attempt runtime in Fig. 4. Learning-based methods exhibit similar runtime dominated by VGGT inference, and are slower than Dong-Si which requires no network forward pass. Ours (FF) takes slightly longer in nonlinear refinement than Ours (SC), since removing feature correspondences yields a flatter optimization landscape that requires more iterations to converge. However, this comparison reflects only the cost of a single attempt. As shown in Fig. 3, the proposed methods require significantly fewer attempts to succeed, thereby achieving the lowest overall computational time for successful initialization.

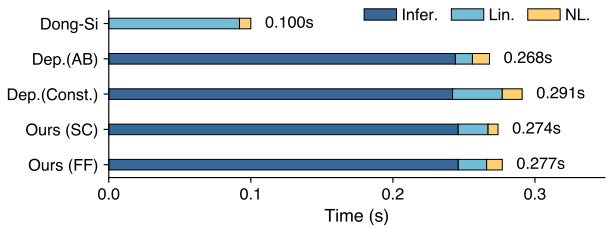

Fig. 4: Per-attempt initialization runtime breakdown. Infer.: VGGT inference. Lin.: linear system. NL.: nonlinear refinement. Note that the proposed methods need significantly fewer attempts.

D. Ablation Study

To validate individual components of the proposed method, we conduct extensive ablation studies on TUM-VI dataset, as summarized in Table V. In addition to standard initialization metrics, we report the scale errors estimated from both the linear system and nonlinear refinement stages, reflecting the scale accuracy of the recovered 3D scene. Unless otherwise specified, the default configuration employs the feature-free formulation with 100 sampled points, RANSAC outlier rejection, and region-partitioned scale.

We first analyze the influence of the number of sampled points K in the linear system. As K increases from 10 to 1000, gravity and velocity errors generally decrease, indicating that more points provide better geometric constraints. However, a larger K leads to a higher computational cost and numerical instability, since more low-confidence points may be included. As can be seen when $K = 1000$, it results in longer T_{tot} and lower SR. From an accuracy perspective, Pts. 500 achieves the best performance, while we also observe that reasonable accuracy can be attained with only 10 points. In summary, we select $K = 100$ for the highest SR.

We then evaluate the remaining components. RANSAC slightly improves both SR and accuracy by filtering outliers in the predicted point cloud. Global Scale, which applies a single scale factor to the entire point cloud rather than region-partitioned scales, achieves comparable performance, indicating that VGGT produces consistent scale predictions across the scene. Since LC fails to initialize within the

TABLE V: Ablation study on TUM-VI. We evaluate sampled points, RANSAC, regional scale, and scale constraints.

	Grav. Err. Lin. (°) ↓	Vel. Err. Lin. (m/s) ↓	Scale Err. Lin. (%) ↓	Scale Err. NL. (%) ↓	Win. ATE (°/m) ↓	Traj. ATE (°/m) ↓	T_{tot} (s) ↓	SR (%) ↑
Pts. 10	2.83	0.35	38.03	29.62	3.05/0.075	1.83/0.058	1.10	86.7
Pts. 20	3.08	0.40	41.83	30.08	3.35/0.077	1.96/0.063	1.03	89.3
Pts. 50	3.19	0.39	43.48	28.63	3.46/0.071	2.15/0.070	1.08	89.3
Pts. 100 (Ours)	**2.97**	**0.39**	**42.48**	**25.72**	**3.22/0.067**	**2.10/0.068**	**1.10**	**94.7**
Pts. 200	2.70	0.39	42.20	24.32	3.00/0.064	2.01/0.061	1.26	88.0
Pts. 500	2.46	0.37	41.10	20.27	2.69/0.054	1.98/0.058	1.26	86.7
Pts. 1000	2.47	0.40	43.50	21.38	2.67/0.055	2.38/0.068	1.48	80.0
w/o RANSAC	2.99	0.39	42.13	25.12	3.22/0.065	2.14/0.071	1.05	88.0
Global Scale	2.89	0.40	41.13	24.87	3.17/0.066	2.22/0.067	1.10	89.3
LC (1s)	1.25	0.21	13.73	-	0.81/0.028	1.46/0.027	4.71	40.0
w/ SC	3.21	0.41	41.62	8.80	1.04/0.023	1.40/0.032	1.16	92.0
w/o SC	4.07	0.50	47.15	-	1.23/0.026	1.63/0.035	2.88	82.7

TABLE VI: Ablation of feed-forward 3D model on linear system: gravity error (°) ↓ / velocity error (m/s) ↓.

Model	room1	room2	room3	room4	room5	room6	Average
Ours (VGGT)	3.57 / 0.33	3.22 / 0.48	2.86 / 0.55	2.45 / 0.31	3.77 / 0.28	1.95 / 0.41	2.97 / 0.39
Ours (DAv3)	1.84 / 0.21	2.32 / 0.59	2.21 / 0.55	2.29 / 0.31	1.53 / 0.15	1.36 / 0.27	1.92 / 0.35
Ours (π^3)	2.08 / 0.25	1.57 / 0.34	2.20 / 0.43	1.41 / 0.15	2.54 / 0.22	1.02 / 0.24	**1.80 / 0.27**

default 0.5 s window, we additionally report LC (1s) with an extended 1 s window. Even under this relaxed setting, LC exhibits significantly lower SR and longer T_{tot}, highlighting the advantage of our feature-free formulation. Finally, the comparison between w/ SC and w/o SC validates the effectiveness of scale-constrained nonlinear refinement. Incorporating scale constraints improves both accuracy and SR, and yields only 8% scale error after refinement. An additional ablation on the temporal window length is provided in Tab. S1 of the supplementary material, showing that longer windows further improve accuracy and success rate.

The proposed method is designed to be agnostic to the choice of feed-forward 3D geometry model. To validate this property and to examine the impact of different 3D predictions on initialization performance, we incorporate alternative models into the pipeline. Specifically, we replace the default VGGT [16] model with Depth Anything 3 (DAv3) [51] and π^3 [52] while keeping all other components unchanged.

As shown in Table VI, Ours (DAv3) and Ours (π^3) consistently achieve lower errors across all sequences than Ours (VGGT), demonstrating that the proposed initialization framework can directly benefit from improved feed-forward geometric predictions. This confirms that the formulation is not tied to a specific 3D model and naturally scales with the quality of the underlying geometry. Among the evaluated models, π^3 yields the best overall performance. We use its latest π^3X variant, which supports conditional injection of known camera intrinsics during inference. In contrast, models without this capability rely on internally inferred intrinsics that may deviate from the true camera parameters, leading to additional geometric inconsistencies in the predicted 3D output. Detailed ablations on π^3 are provided in Tab. S2 of the supplementary material.

E. Evaluation on Self-Collected Dataset

To validate the generalization and applicability of the feature-free formulation, we collect a dataset that contains various challenging real-world scenarios, as shown in Fig. 5. The data are recorded by a RealSense D455 camera, which provides RGB images at 30 Hz and consumer-grade IMU measurements at 200 Hz. Each sequence is approximately 25 seconds long, with figure-eight motion patterns designed specifically for initialization validation. Since no ground truth is available, we qualitatively evaluate initialization success by examining whether the VINS pose tracking severely drifts after initialization. Due to the challenging nature of the scenarios, all methods use a 2 s initialization window. Table VII shows qualitative results.

(a) Indoor-Structured (b) Indoor-Degraded (c) Outdoor

Fig. 5: Example scenes from our self-collected dataset: (a) indoor office with complex depth, (b) visually degraded indoor environment, and (c) outdoor.

In Indoor-Structured scenarios, all learning-based methods succeed, demonstrating strong geometric inference capability of the 3D foundation model. In Indoor-Degraded scenarios, only 50–60 features are detected per frame on average. Under such conditions, all baselines often diverge severely or even fail to satisfy initialization requirements, whereas our feature-free formulation succeeds reliably, as shown in Fig. 6.

TABLE VII: Success rate evaluation on self-collected dataset.

Category	Seq.	Ours (FF)	Dep.(Const.)	Dep.(AB)	Dong-Si	LC	sqrtVINS	DRT
Indoor-Structured	office_01	✓	✓	✓	✓	✓	✓	✓
	office_02	✓	✓	✓	✓	✓	✓	✓
Indoor-Degraded	printer_01	✓	✗	✓	✗	✗	✗	✗
	printer_02	✓	✗	✗	✗	✗	✗	✗
	lift_01	✓	✗	✗	✗	✗	✓	✓
	lift_02	✓	✗	✗	✗	✗	✗	✗
Outdoor	outdoor_01	✓	✗	✗	✓	✗	✓	✗
	outdoor_02	✓	✓	✗	✗	✗	✗	✗
	outdoor_03	✓	✓	✓	✗	✓	✓	✓
	outdoor_04	✗	✗	✗	✗	✗	✗	✗
SR (%) ↑		**90**	40	40	30	30	50	40

✓: successful initialization without obvious drift. ✗: failure or severe drift.

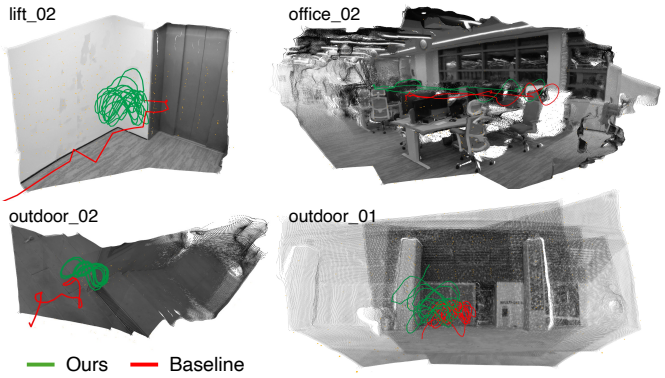

Fig. 6: Trajectory comparison on representative sequences. Green: Ours. Red: Baseline (only one shown for clarity). The baseline exhibits significant drift in degraded scenarios, while Ours (FF) maintains stable tracking across all categories.

However, we find that Ours (FF) might also fail in extremely degraded environments (e.g., blank white walls), as the input images lack sufficient geometric cues for the 3D model. In Outdoor scenarios, all methods fail on 'outdoor_04' sequence due to the open-sky scene, where the 3D model produces unreliable predictions. Further dataset samples and additional qualitative trajectory comparisons across all sequences are provided in Fig. S2 and Fig. S3 of the supplementary material.

VI. Conclusion and Future Work

We present a feature-free monocular visual-inertial initialization framework that leverages point clouds predicted by feed-forward 3D models. Using the predicted scene geometry, we derive an efficient closed-form linear solver and a compact nonlinear refinement formulation. Extensive experiments on public datasets demonstrate that our method achieves success rates exceeding 90% while requiring only about 1.2 s of sensory data for initialization. The low scale error further indicates that the metric scale of the 3D model can be reliably recovered. Results on a self-collected dataset further validate the generalization across various scenarios, with robust performance particularly in visually degraded environments.

Our method still has limitations. While the feature-free formulation achieves higher efficiency and success rates than its scale-constrained counterpart, it can yield slightly lower pose accuracy; in practice, the two formulations can be selected according to application requirements. A more fundamental limitation lies in the generalization of current feed-forward 3D foundation models: in open-sky scenes with sparse or ambiguous visual cues, the predicted geometry becomes unreliable and degrades initialization, which is a challenge shared by all learning-based baselines evaluated here. Since our formulation is model-agnostic, future improvements in foundation models will directly translate into better initialization. Developing a fully differentiable, feed-forward VINS is another promising direction toward further robustness.

Acknowledgments

This work is partially supported by "The MBZUAI-WIS Joint Program for Artificial Intelligence Research".

References

[1] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3565–3572.

[2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[3] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4666–4672.

[4] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: LiDAR-Inertial-Camera Odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5848–5854.

[5] Y. Zhang, F. Zhu, Q. Cai, J. Lv, Z. Xu, X. Chen, and X. Zhang, "Towards More Precise and Robust Positioning in Urban Environments Through an Enhanced FGO-Based GNSS RTK Framework," *IEEE Transactions on Intelligent Vehicles*, vol. 9, pp. 7603–7616, 2024.

[6] B. Yi, V. Ye, M. Zheng, Y. Li, L. Müller, G. Pavlakos, Y. Ma, J. Malik, and A. Kanazawa, "Estimating Body

and Hand Motion in an Ego-sensed World," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 7072–7084.

[7] C. Kong, J. Fort, A. Kang *et al.*, "Aria Gen 2 Pilot Dataset," arXiv preprint arXiv:2510.16134, 2025.

[8] L. Schmid, M. Abate, Y. Chang, and L. Carlone, "Khronos: A Unified Approach for Spatio-Temporal Metric-Semantic SLAM in Dynamic Environments," in *Robotics: Science and Systems (RSS)*, 2024.

[9] C. Kassab, M. Mattamala, L. Zhang, and M. Fallon, "Language-EXtended Indoor SLAM (LEXIS): A Versatile System for Real-time Visual Scene Understanding," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 988–15 994.

[10] H. Zhang, C.-C. Chen, H. Vallery, and T. D. Barfoot, "GNSS/Multisensor Fusion Using Continuous-Time Factor Graph Optimization for Robust Localization," *IEEE Transactions on Robotics*, vol. 40, pp. 4003–4023, 2024.

[11] Z. Xu, F. Zhu, Z. Zhang, C. Jian, J. Lv, Y. Zhang, and X. Zhang, "PO-GVINS: A Tightly Coupled GNSS-Visual-Inertial Navigation Framework Using Pose-Only Representation," *IEEE Robotics and Automation Letters*, vol. 10, pp. 10 830–10 837, 2025.

[12] T.-C. Dong-Si and A. I. Mourikis, "Estimator Initialization in Vision-Aided Inertial Navigation with Unknown Camera-IMU Calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 1064–1071.

[13] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-Inertial Odometry With Structural Regularity of Man-Made Environments," *IEEE Transactions on Robotics*, vol. 35, pp. 999–1013, 2019.

[14] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, pp. 1874–1890, 2021.

[15] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "DUSt3R: Geometric 3D Vision Made Easy," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 697–20 709.

[16] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "VGGT: Visual Geometry Grounded Transformer," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 5294–5306.

[17] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa, "Continuous 3D Perception Model with Persistent State," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 10 510–10 522.

[18] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[19] R. Mur-Artal and J. D. Tardós, "Visual-Inertial Monocular SLAM With Map Reuse," *IEEE Robotics and Automation Letters*, vol. 2, pp. 796–803, 2017.

[20] A. Martinelli, "Closed-Form Solution of Visual-Inertial Structure from Motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[21] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous State Initialization and Gyroscope Bias Calibration in Visual Inertial Aided Navigation," *IEEE Robotics and Automation Letters*, vol. 2, pp. 18–25, 2017.

[22] C. Campos, J. M. Montiel, and J. D. Tardós, "Fast and Robust Initialization for Visual-Inertial SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1288–1294.

[23] Y. He, B. Xu, Z. Ouyang, and H. Li, "A Rotation-Translation-Decoupled Solution for Robust and Efficient Visual-Inertial Initialization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 739–748.

[24] Y. Peng, C. Chen, K. Wu, and G. Huang, "sqrtvins: Robust and Ultrafast Square-Root Filter-Based 3D Motion Tracking," *IEEE Transactions on Robotics*, vol. 41, pp. 6570–6589, 2025.

[25] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[26] V. Leroy, Y. Cabon, and J. Revaud, "Grounding Image Matching in 3D with MASt3R," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 71–91.

[27] X. Chen, Y. Chen, Y. Xiu, A. Geiger, and A. Chen, "TTT3R: 3D Reconstruction as Test-Time Training," arXiv preprint arXiv:2509.26645, 2025.

[28] H. Wang and L. Agapito, "3D Reconstruction with Spatial Memory," arXiv preprint arXiv:2408.16061, 2024.

[29] R. Murai, E. Dexheimer, and A. J. Davison, "MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 16 695–16 705.

[30] D. Maggio, H. Lim, and L. Carlone, "VGGT-SLAM: Dense RGB SLAM Optimized on the SL(4) Manifold," arXiv preprint arXiv:2505.12549, 2025.

[31] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3D V2: A Versatile Monocular Geometric Foundation Model for Zero-Shot Metric Depth and Surface Normal Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 579–10 596, 2024.

[32] N. Keetha, N. Müller, J. Schönberger, L. Porzi, Y. Zhang, T. Fischer, A. Knapitsch, D. Zauss, E. Weber, N. Antunes, J. Luiten, M. Lopez-Antequera, S. R. Bulò, C. Richardt, D. Ramanan, S. Scherer, and P. Kontschieder, "MapAnything: Universal Feed-Forward Metric 3D Reconstruction," arXiv preprint arXiv:2509.13414, 2025.

[33] H. Wang and L. Agapito, "AMB3R: Accurate Feed-Forward Metric-Scale 3D Reconstruction with Backend," arXiv preprint arXiv:2511.20343, 2025.

[34] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 16 558–16 569.

[35] C. Wang, K. Ji, J. Geng, Z. Ren, T. Fu, F. Yang, Y. Guo, H. He, X. Chen, Z. Zhan, Q. Du, S. Su, B. Li, Y. Qiu, Y. Du, Q. Li, Y. Yang, X. Lin, and Z. Zhao, "Imperative Learning: A Self-Supervised Neuro-Symbolic Learning Framework for Robot Autonomy," *The International Journal of Robotics Research*, p. 02783649251353181, 2025.

[36] Y. Yuan, Z. Chen, K. Li, W. Wang, and H. Zhao, "SLAM-Former: Putting SLAM into One Transformer," arXiv preprint arXiv:2509.16909, 2025.

[37] X. Zuo, N. Merrill, W. Li, Y. Liu, M. Pollefeys, and G. Huang, "CodeVIO: Visual-Inertial Odometry with Learned Optimizable Dense Depth," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 14 382–14 388.

[38] N. Merrill and G. Huang, "Visual-Inertial SLAM as Simple as A, B, VINS," arXiv preprint arXiv:2406.05969, 2024.

[39] L. Wang, L. Guo, Z. Xu, Q. Wang, F. Gao, and X. Chen, "LiDAR-VGGT: Cross-Modal Coarse-to-Fine Fusion for Globally Consistent and Metric-Scale Dense Mapping," arXiv preprint arXiv:2511.01186, 2025.

[40] Y. Zhou, A. Kar, E. Turner, A. Kowdle, C. X. Guo, R. C. DuToit, and K. Tsotsos, "Learned Monocular Depth Priors in Visual-Inertial Initialization," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 552–570.

[41] N. Merrill, P. Geneva, S. Katragadda, C. Chen, and G. Huang, "Fast Monocular Visual-Inertial Initialization Leveraging Learned Single-View Depth," in *Robotics: Science and Systems (RSS)*, 2023.

[42] P. G. Savage, "Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 19–28, 1998.

[43] T. Lupton and S. Sukkarieh, "Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.

[44] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation," in *Robotics: Science and Systems (RSS)*, 2015.

[45] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency Analysis and Improvement of Vision-aided Inertial Navigation," *IEEE Transactions on Robotics*, vol. 30, pp. 158–176, 2014.

[46] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, pp. 932–945, 2008.

[47] R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron, "Learning Single Camera Depth Estimation Using Dual-Pixels," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7628–7637.

[48] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC Micro Aerial Vehicle Datasets," *The International Journal of Robotics Research*, vol. 35, pp. 1157–1163, 2016.

[49] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1680–1687.

[50] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251.

[51] H. Lin, S. Chen, J. Liew, D. Y. Chen, Z. Li, G. Shi, J. Feng, and B. Kang, "Depth Anything 3: Recovering the Visual Space from Any Views," arXiv preprint arXiv:2511.10647, 2025.

[52] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He, "π^3: Scalable Permutation-Equivariant Visual Geometry Learning," arXiv preprint arXiv:2507.13347, 2025.