FFBA: FEED-FORWARD BUNDLE ADJUSTMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose *feed-forward bundle adjustment* (FFBA), which recasts per-scene iterative BA as a single forward pass of a learned refinement model with update rules shared across scenes. Given initial cameras and geometry from a *geometric foundation model* (GFM) and fixed multi-view tracks, FFBA constructs a sparse heterogeneous graph of cameras, 3D points and 2D observations. It then refines poses, focal lengths and 3D points through a geometry-conditioned graph network stacked with geometric graph aggregation layers. These layers update observation, camera and point features via graph message propagation, and filter unreliable observations through uncertainty-guided confidence estimation. A confidence-weighted reprojection loss trains the whole framework end-to-end using only in-house data, without manual annotation. To support training across datasets and GFMs, we develop a standardized sample-construction pipeline that yields over 0.9M samples from seven datasets and three GFMs with shared multi-view tracks. Experiments on three in-domain and four cross-domain datasets show that FFBA consistently improves reprojection consistency and camera accuracy over GFM initializations. It also remains competitive with classical BA and EPO under domain shift, and reduces optimization cost by more than an order of magnitude, translating into a clear end-to-end efficiency advantage that is expected to grow with scene scale. Code will be shared upon paper acceptance.

1 INTRODUCTION

Recovering camera motion and 3D scene structure from overlapping images is a fundamental problem in computer vision. Classical *structure-from-motion* (SfM) pipelines sequentially perform feature matching, camera estimation, triangulation and geometric optimization, yielding reliable reconstructions (Snavely et al., 2006; Schönberger & Frahm, 2016). However, this multistage processing and repeated per-scene optimization create a long, input-dependent computational path that limits large-scale and low-latency deployment (Pan et al., 2024). In contrast, *geometric foundation models* (GFMs) and related feed-forward systems directly predict camera parameters, depth maps, point maps and cross-view correspondences, rapidly producing largely complete geometric initializations. Their efficient recovery of multiple geometric quantities has made feed-forward reconstruction an increasingly active area of research (Wang et al., 2024a;b; 2025a;b; 2026a;b; Yang et al., 2025).

Although feed-forward models substantially improve reconstruction efficiency, the geometric quantities they directly regress are not necessarily consistent. A fundamental multi-view constraint requires the projections of the same 3D point across views to agree with their corresponding 2D observations (Hartley & Zisserman, 2004). *Bundle adjustment* (BA) enforces this constraint by jointly refining camera parameters and 3D points to minimize reprojection error over shared observations (Triggs et al., 2000). As the central joint-estimation stage in conventional SfM, BA corrects errors accumulated during camera registration and triangulation by coordinating cameras, 3D structure and shared observations under a unified reprojection objective. This joint optimization integrates local estimates into a globally consistent reconstruction and has consequently become a natural choice for refining GFM outputs (Wang et al., 2025a; Pan et al., 2026; Wang et al., 2026b).

Classical BA, however, is typically solved through a convergence-driven iterative procedure applied independently to each scene. Each iteration linearizes the nonlinear reprojection objective and solves the resulting large sparse linear system. Although modern solvers improve scalability (Weber et al., 2023; Safari, 2025), this procedure still incurs substantial backend computation, whose cost depends on problem scale, initialization quality, numerical conditioning and convergence behavior. When

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

applied to refine GFM outputs, this additional optimization cost partially offsets the efficiency of feed-forward reconstruction. Moreover, conventional BA solves each scene in isolation, ignoring the error patterns that recur across GFM initializations and datasets. It must re-discover the correction behavior for every scene through repeated linearization, which is time-consuming and sometimes even degrades accuracy under noisy tracks or weak connectivity. This motivates a refinement model that shares update rules across scenes and corrects GFM geometry in a single forward pass.

To this end, we propose *Feed-Forward Bundle Adjustment* (FFBA), which jointly refines initial cameras and 3D structure in a single forward pass under multi-view geometric constraints. Given multiple images, we first employ a GFM to obtain the initial cameras and 3D geometries. Subsequently, we develop a multi-view track construction module to produce the 2D observations that are required for the BA task. These cameras, 3D points and observations are then formulated as distinct node types in a sparse heterogeneous graph. Moreover, we propose a novel *geometry-conditioned graph network* (GCGN), which takes the sparse heterogeneous graph as input and outputs refined cameras and 3D geometries. Specifically, the GCGN is stacked with multiple *geometric graph aggregation* (GGA) layers, each of which updates the camera, 3D points and observation confidence through graph message propagation. Finally, a confidence-weighted reprojection loss is adopted to guide the training of FFBA, which uses only the in-house data of our framework, without the need for manual annotation.

To train FFBA across datasets and GFMs, we develop a standardized sample construction pipeline. We organize scenes from DL3DV10K (Ling et al., 2024), MVImgNet (Yu et al., 2023) and ScanNet++ (Yeshwanth et al., 2023) into approximately 300K split scenes and process each with three GFMs, yielding over 900K samples. For evaluation, we compare our FFBA with classical BA and EPO (D'Urso et al., 2026) on three in-domain and four cross-domain datasets in terms of reprojection consistency, camera accuracy, and computational efficiency. Our FFBA outperforms classical BA and EPO on in-domain datasets and gains comparable results on cross-domain datasets. Considering computational efficiency, our method achieves optimization speedups of roughly one and two orders of magnitude over classical BA and EPO, respectively.

In summary, our contributions are threefold:

- We propose FFBA, a feed-forward bundle adjustment framework that recasts per-scene iterative BA as a single forward pass of a learned refinement model, jointly refining camera poses, focal lengths and 3D points under multi-view reprojection constraints across scenes.
- We design a geometry-conditioned graph network that performs geometry-aware refinement through stacked geometric graph aggregation layers with uncertainty-guided observation filtering and cascaded unsupervised losses.
- We build a standardized multi-GFM dataset and systematically evaluate FFBA, showing that it consistently improves over GFM initializations, remains competitive with classical BA and EPO under domain shift, and cuts optimization cost by more than an order of magnitude.

2 Related Work

Feed-Forward Multi-View Reconstruction. Recent learning-based methods have extended multi-view reconstruction from pairwise geometric prediction to streaming and scene-level feed-forward models. DUSt3R (Wang et al., 2024b) and MASt3R (Leroy et al., 2024) predict point maps and correspondences for image pairs and extend them to multiple views through global alignment. Fast3R (Yang et al., 2025) and CUT3R (Wang et al., 2025b) scale this paradigm to large collections and streaming inputs, respectively. Recently, VGGT (Wang et al., 2025a), π^3 (Wang et al., 2026b) and VGGT-Ω (Wang et al., 2026a) jointly recover cameras and 3D geometry with unified scene-level networks. By replacing much of the conventional sequence of matching, initialization and geometric recovery with network inference, these methods substantially shorten the reconstruction pipeline. However, their outputs are obtained primarily through direct prediction, rather than by optimizing a unified reprojection objective over all shared 2D observations in a test scene. As a result, there is no guarantee that the predicted cameras and 3D structure are jointly optimal with respect to all observations in that scene. The use of global alignment in DUSt3R (Wang et al., 2024b) and MASt3R (Leroy et al., 2024), together with the BA post-processing in VGGT (Wang et al., 2025a), further illustrates that learned feed-forward prediction and scene-level geometric refinement

play complementary roles: the former efficiently provides a coarse geometric initialization, while the latter improves global consistency between cameras and 3D structure.

Multi-View Geometric Refinement. Classical BA jointly refines camera parameters and 3D points by minimizing reprojection error (Triggs et al., 2000). Modern sparse solvers exploit its block structure to improve scalability (Demmel et al., 2021; Weber et al., 2023), but still require iterative linearization and numerical solution for each scene. Learning-based systems embed differentiable BA or iterative geometric updates into end-to-end pipelines (Tang & Tan, 2019; Teed & Deng, 2021; Wang et al., 2024a). These methods either retain explicit test-time optimization or focus on local BA rather than post-refinement of GFM initializations. Beyond standard BA, geometric post-optimization has also been formulated through learned feature alignment, 3D correspondence alignment, and edge- or depth-based objectives (Lindenberger et al., 2021; Duisterhof et al., 2025; D'Urso et al., 2026). Hybrid designs further combine sparse BA with learned point-map refinement (Chen et al., 2026). These approaches demonstrate the value of refining feed-forward geometry, but rely on per-scene optimization or a numerical BA stage. FFBA instead learns refinement rules shared across scenes and jointly updates camera parameters and 3D structure in a single forward pass.

3 Method

3.1 Problem Formulation

Given N_V cameras and N_S 3D points for an initialized scene, each camera i has a pose $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i)$, where $\mathbf{R}_i \in \mathrm{SO}(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$, and an intrinsic matrix $\mathbf{K}_i \in \mathbb{R}^{3\times3}$ containing its focal lengths and principal point. Each point j has position $\mathbf{X}_j \in \mathbb{R}^3$. The geometric state is $\boldsymbol{\xi} = (\{\mathbf{T}_i, \mathbf{K}_i\}_{i=1}^{N_V}, \{\mathbf{X}_j\}_{j=1}^{N_S})$. The observation index set $\mathcal{O} \subseteq \{1,\dots,N_V\}\times\{1,\dots,N_S\}$ contains (i,j) if point j has an available 2D observation $\mathbf{u}_{ij} \in \mathbb{R}^2$ in view i. The track of point j and the collection of all tracks are

$$\mathcal{U}_j = \{(i, \mathbf{u}_{ij}) \mid (i,j) \in \mathcal{O}\}, \qquad \mathcal{U} = \{\mathcal{U}_j\}_{j=1}^{N_S}. \tag{1}$$

The index set $\mathcal{O}$ records observation associations, while $\mathcal{U}$ stores the corresponding 2D observations grouped into tracks. Both remain fixed during refinement.

For $(i,j) \in \mathcal{O}$, the reprojection residual is

$$\Delta_{ij} = \Pi(\mathbf{K}_i\,(\mathbf{R}_i\mathbf{X}_j + \mathbf{t}_i)) - \mathbf{u}_{ij}, \tag{2}$$

where Π performs perspective division from homogeneous image coordinates to 2D coordinates. The residual Δ_{ij} is a 2D vector, and its Euclidean norm $\|\Delta_{ij}\|_2$ is the pixel reprojection error. These projection relations couple camera parameters and 3D points through shared 2D observations, providing geometric constraints for joint refinement.

Given initial geometry $\boldsymbol{\xi}^0$ and fixed tracks $\mathcal{U}$, we learn a feed-forward mapping $\Phi_{\boldsymbol{\theta}}$ with shared parameters $\boldsymbol{\theta}$ to jointly refine camera poses, focal lengths, and 3D points under reprojection constraints, while keeping the remaining intrinsics fixed. Let $\mathcal{D}$ denote the training data distribution. The training objective and inference mapping are

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{\xi}^0,\mathcal{U})\sim\mathcal{D}}\left[\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\xi}^0, \mathcal{U})\right], \tag{3}$$

$$\widehat{\boldsymbol{\xi}} = \Phi_{\boldsymbol{\theta}^\star}(\boldsymbol{\xi}^0, \mathcal{U}). \tag{4}$$

Here, $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\xi}^0, \mathcal{U})$ is the reprojection training loss for a given input, and $\boldsymbol{\theta}^\star$ denotes the learned shared parameters. Training uses reprojection consistency with the 2D observations, without ground-truth supervision of camera parameters or 3D points. At inference, the parameters remain fixed, and a single forward pass through Eq. 4 yields the refined geometric state.

3.2 Overview

As illustrated in Fig. 1, our FFBA framework contains three main components:

- **Data Preparation.** Given a set of images, a GFM first produces an initial geometry, including camera poses, intrinsics and 3D points. In parallel, a multi-view track construction module builds a

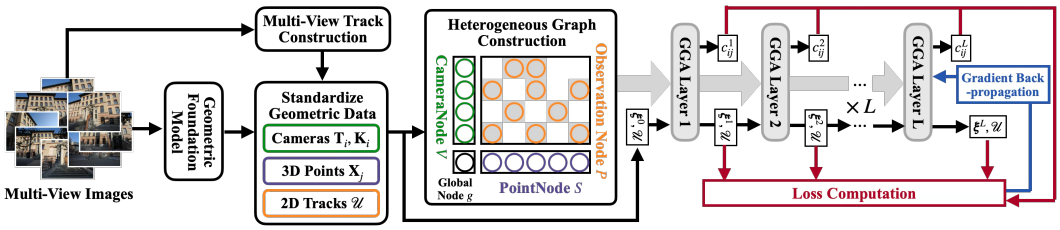


Figure 1: **Overview of FFBA.** $\boldsymbol{\xi}^0$ is the initial geometric state standardized from the GFM outputs and the constructed tracks, and $\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^L$ are the states refined by successive GGA layers. $\mathcal{U}$ denotes the fixed tracks shared by all layers, and $c_{ij}^1, \ldots, c_{ij}^L$ are the observation confidences predicted at each layer. The training loss is computed from the states and confidences of all layers, while inference is a single forward pass that maps $(\boldsymbol{\xi}^0, \mathcal{U})$ to $\boldsymbol{\xi}^L$.

set of 2D observation associations, where each track records the observations of a 3D point across multiple views. These tracks are independent of any specific GFM and remain fixed throughout refinement, serving as the multi-view reprojection constraints. The initial geometry and the fixed tracks together form the input to the refinement network.

- **Heterogeneous Graph Construction.** The initial geometry and tracks are then formulated as a sparse heterogeneous graph with four node types: camera nodes, 3D point nodes, observation nodes and a global node. Edges are induced by the observation associations, connecting each observation node to its associated camera and 3D point nodes. The graph preserves the sparse coupling structure of BA and incorporates reprojection constraints through message passing rather than explicit normal equations.
- **Geometry-Conditioned Graph Network.** The refinement backbone is composed of multiple *geometric graph aggregation* (GGA) layers. Each layer performs four steps: updating observation nodes, filtering unreliable observations, updating camera and 3D point nodes, and updating the global node. Geometric updates are applied in a residual manner with learned masks and step sizes. Training requires no ground-truth camera or point supervision. It instead uses a confidence-weighted reprojection loss accumulated over layers.

3.3 Heterogeneous Graph Construction

We adopt the heterogeneous node organization of GASfM (Brynte et al., 2024), connecting cameras and 3D points through observation nodes and aggregating scene context in a global node. This structure follows the sparse camera–point associations defined by shared 2D observations, making it suitable for modeling the coupling between geometric variables in BA. The graph is defined as:

$$\mathcal{G} = (\mathcal{N}, \mathcal{A}), \qquad \mathcal{N} = \mathcal{P} \cup \mathcal{V} \cup \mathcal{S} \cup \{g\}, \tag{5}$$

where $\mathcal{P}$, $\mathcal{V}$ and $\mathcal{S}$ contain observation nodes P_{ij} for $(i,j) \in \mathcal{O}$, camera nodes V_i and point nodes S_j, respectively; g is the global node. The relation set $\mathcal{A}$ connects each P_{ij} to V_i and S_j, allows cameras and points to send messages to g, and broadcasts global context from g to observations.

We denote node features by $\mathbf{p}_{ij}$, $\mathbf{v}_i$, $\mathbf{s}_j$ and $\mathbf{g}$. Type-specific encoders initialize them as:

$$\mathbf{p}_{ij}^0 = \phi_P(\mathbf{u}_{ij}), \quad \mathbf{v}_i^0 = \phi_V(\mathbf{t}_i \,\|\, \mathbf{q}_i \,\|\, \varphi(\mathbf{K}_i)), \quad \mathbf{s}_j^0 = \phi_S(\mathbf{X}_j), \tag{6}$$

where $\|$ denotes feature concatenation, $\mathbf{q}_i$ is the unit quaternion representing $\mathbf{R}_i$, and φ converts the initial intrinsics into horizontal and vertical field-of-view features. The encoders ϕ_P, ϕ_V, and ϕ_S normalize input coordinates and use MLPs and layer normalization to produce the initial features. Specifically, the initial global feature $\mathbf{g}^0$ is acquired using zero-mean Gaussian initialization.

3.4 Geometric Graph Aggregation

As illustrated in Fig. 2, the ℓ-th GGA layer takes the fixed tracks $\mathcal{U}$ and the geometric state $\boldsymbol{\xi}^{\ell-1}$ from the preceding layer as input (see App. A for pseudocode). It first updates observation nodes by combining geometric cues with camera, point and global features. An uncertainty-guided filtering strategy then masks unreliable observations, and the retained observation nodes are used to update the camera and point nodes. Finally, the global node is updated by aggregating the updated camera and point nodes.

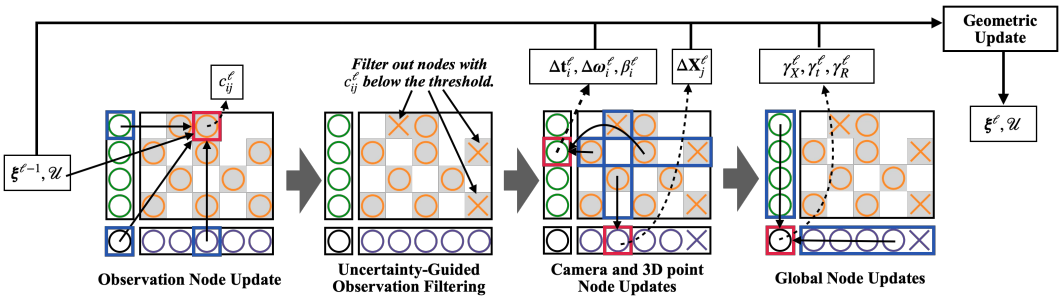


Figure 2: **Geometric graph aggregation layer.** The fixed tracks $\mathcal{U}$ are used to compute reprojection residuals for updating observation nodes (Eq. 7), and are passed unchanged to the next layer. Red and blue boxes mark the node being updated and its message sources, respectively. Dashed arrows indicate outputs of prediction heads. Crosses indicate masked observation nodes (Eq. 9) and masked camera or point nodes due to insufficient observation support (Eq. 10). The decoded geometric increments and learned step sizes update the geometric state from $\boldsymbol{\xi}^{\ell-1}$ to $\boldsymbol{\xi}^{\ell}$ (Eq. 15).

Observation Node Update. Both geometry cues and graph topology are employed for the updating of observation nodes. We first compute $\Delta_{ij}^{\ell-1}$ (Eq. 2) using the current geometries, and then update each observation node feature as:

$$\mathbf{p}_{ij}^{\ell} = \mathbf{p}_{ij}^{\ell-1} + \underbrace{\psi_P^{\ell}\left(\Delta_{ij}^{\ell-1}\right)}_{\text{reprojection residual}} + \underbrace{\mathrm{FFN}_P^{\ell}\left(\mathbf{v}_i^{\ell-1} \parallel \mathbf{s}_j^{\ell-1} \parallel \mathbf{g}^{\ell-1}\right)}_{\text{camera, point, and global features}}, \tag{7}$$

where the encoder ψ_P^{ℓ} maps the reprojection residual into the latent feature space, and FFN_P^{ℓ} fuses concatenated features into a context increment that contains two linear layers separated by a GELU activation. Unlike graph convolution that aggregates node features alone, the encoded reprojection residual is crucial for our method, which provides not only multi-view consistency but also geometric cues for the following node filtering.

Uncertainty-Guided Observation Filtering. As discussed in (Schönberger & Frahm, 2016; Lindenberger et al., 2021), unreliable observations can corrupt camera and point updates. Using the aleatoric uncertainty (Kendall & Gal, 2017), a confidence is learned for each observation node:

$$c_{ij}^{\ell} = h_c^{\ell}(\mathbf{p}_{ij}^{\ell}), \qquad c_{ij}^{\ell} \in (0, 1). \tag{8}$$

The head h_c^{ℓ} includes feature normalization, an MLP and a sigmoid output. Larger scores indicate higher predicted reliabilities, with which the active observation set is constructed as:

$$\mathcal{O}_{\text{act}}^{\ell} = \{(i,j) | (i,j) \in \mathcal{O}, c_{ij}^{\ell} \geq \tau\}, \tag{9}$$

where $\tau \in (0, 1)$ controls the degree of unreliable observation filtering.

Camera and 3D Point Node Updates. Based on the constructed active observation set, only the camera and point nodes that are associated with sufficient active observation nodes need to be updated. We first compute two binary node-update masks:

$$m_i^{V,\ell} = \mathbb{I}\left[\left|\{j \mid (i,j) \in \mathcal{O}_{\text{act}}^{\ell}\}\right| \geq \kappa_V\right], \qquad m_j^{S,\ell} = \mathbb{I}\left[\left|\{i \mid (i,j) \in \mathcal{O}_{\text{act}}^{\ell}\}\right| \geq \kappa_S\right], \tag{10}$$

where $\mathbb{I}[\cdot]$ is the binary indicator function, and κ_V and κ_S are predefined thresholds. Camera nodes and point nodes are then updated independently as:

$$\begin{aligned} \mathbf{v}_i^{\ell} &= \mathbf{v}_i^{\ell-1} + m_i^{V,\ell}\, \mathrm{FFN}_V^{\ell}\left(\mathrm{Attn}_{P\to V}^{\ell}(\{\mathbf{p}_{ij}^{\ell}\}_{j:(i,j)\in\mathcal{O}_{\text{act}}^{\ell}}, \mathbf{v}_i^{\ell-1})\right), \\ \mathbf{s}_j^{\ell} &= \mathbf{s}_j^{\ell-1} + m_j^{S,\ell}\, \mathrm{FFN}_S^{\ell}\left(\mathrm{Attn}_{P\to S}^{\ell}(\{\mathbf{p}_{ij}^{\ell}\}_{i:(i,j)\in\mathcal{O}_{\text{act}}^{\ell}}, \mathbf{s}_j^{\ell-1})\right), \end{aligned} \tag{11}$$

where $\{\mathbf{p}_{ij}^{\ell}\}_{j:(i,j)\in\mathcal{O}_{\text{act}}^{\ell}}$ (respectively $\{\mathbf{p}_{ij}^{\ell}\}_{i:(i,j)\in\mathcal{O}_{\text{act}}^{\ell}}$) contains retained observation features associated with camera i (respectively point j). FFN_V^{ℓ} and FFN_S^{ℓ} share the same architecture as FFN_P^{ℓ} with independent parameters. $\mathrm{Attn}_{P\to V}^{\ell}$ and $\mathrm{Attn}_{P\to S}^{\ell}$ are graph attention networks (Brody et al., 2022) aggregate observation features to camera nodes and point nodes, respectively.

Global Node Update. The global node is finally updated to embed scene-level information:

$$\mathbf{g}^{\ell} = \mathbf{g}^{\ell-1} + \mathrm{FFN}_g^{\ell}\big(\mathrm{Attn}_{V\to g}^{\ell}(\{\mathbf{v}_i^{\ell}\}_{i:m_i^{V,\ell}=1}, \mathbf{g}^{\ell-1}) \,\|\, \mathrm{Attn}_{S\to g}^{\ell}(\{\mathbf{s}_j^{\ell}\}_{j:m_j^{S,\ell}=1}, \mathbf{g}^{\ell-1})\big). \tag{12}$$

The messages aggregated from active camera and point nodes are concatenated and fused by FFN_g^{ℓ}, which also shares the same architecture as FFN_P^{ℓ}. The global feature provides scene-level context for the following geometry update by decoding step sizes:

$$\big(\gamma_X^{\ell}, \gamma_t^{\ell}, \gamma_R^{\ell}\big) = h_{\gamma}^{\ell}(\mathbf{g}^{\ell}), \tag{13}$$

where the head h_{γ}^{ℓ} contains an input layer normalization, an MLP and a Softplus output, γ_X^{ℓ}, γ_t^{ℓ} and γ_R^{ℓ} are learned update steps of 3D points, camera translation and rotation, respectively.

Geometry Update. The geometric corrections are decoded from camera and point features:

$$(\Delta\mathbf{t}_i^{\ell}, \Delta\boldsymbol{\omega}_i^{\ell}) = h_{\mathrm{pose}}(\mathbf{v}_i^{\ell}), \quad \Delta\mathbf{X}_j^{\ell} = h_{\mathrm{point}}(\mathbf{s}_j^{\ell}), \quad \beta_i^{\ell} = h_{\beta}^{\ell}(\mathbf{v}_i^{\ell}), \tag{14}$$

where $\Delta\mathbf{t}_i^{\ell}$, $\Delta\boldsymbol{\omega}_i^{\ell}$ and $\Delta\mathbf{X}_j^{\ell}$ are the increments of camera translation, rotation and 3D point coordinates, respectively, and β_i^{ℓ} is a scalar relative focal correction. The pose decoder h_{pose} and point decoder h_{point} are shared across layers, while the focal decoder h_{β}^{ℓ} is layer-specific. All three heads apply input layer normalization. The heads h_{pose} and h_{point} then use three-layer MLPs with GELU activations between linear layers, whereas h_{β}^{ℓ} uses a single linear layer. The geometries are then updated according to the learned node masks and step sizes:

$$\begin{aligned} \mathbf{X}_j^{\ell} &= \mathbf{X}_j^{\ell-1} + m_j^{S,\ell}\gamma_X^{\ell}\Delta\mathbf{X}_j^{\ell}, & \mathbf{t}_i^{\ell} &= \mathbf{t}_i^{\ell-1} + m_i^{V,\ell}\gamma_t^{\ell}\Delta\mathbf{t}_i^{\ell}, \\ \mathbf{R}_i^{\ell} &= \mathbf{R}_i^{\ell-1}\exp\Big([m_i^{V,\ell}\gamma_R^{\ell}\Delta\boldsymbol{\omega}_i^{\ell}]_{\times}\Big), & \mathbf{f}_i^{\ell} &= [1 + m_i^{V,\ell}\beta_i^{\ell}]\mathbf{f}_i^{0}, \end{aligned} \tag{15}$$

where $[\cdot]_{\times}$ maps a 3D rotation increment to its skew-symmetric matrix, the matrix exponential $\exp(\cdot)$ maps it to $\mathrm{SO}(3)$ for right-composition with the current rotation, and $\mathbf{f}_i^0$ denotes the initial focal lengths of camera i. The per-layer updated geometries are used to compute the per-layer reprojection residual Δ_{ij}^{ℓ} (Eq. 2), which is accumulated in the final training loss.

3.5 Confidence-Weighted Reprojection Loss

In order to reduce the influence of unreliable observations, we propose a confidence-weighted reprojection loss following aleatoric uncertainty weighting (Kendall & Gal, 2017; Wang et al., 2025a):

$$\mathcal{L} = \sum_{\ell=1}^{L} \frac{w_{\ell}}{|\mathcal{O}|} \sum_{(i,j)\in\mathcal{O}} \Big(c_{ij}^{\ell}\,\big\|\Delta_{ij}^{\ell}\big\|_2 - \alpha \log c_{ij}^{\ell}\Big), \tag{16}$$

where c_{ij}^{ℓ} is the learned confidence (Eq. 8), $\|\Delta_{ij}^{\ell}\|_2$ is the reprojection error, $\alpha > 0$ controls confidence regularization, and the weights $0 < w_1 \le \cdots \le w_L$ place greater emphasis on later layers. The first term gives high-confidence observations a stronger influence on geometric refinement, and the regularizer prevents a trivial solution of uniformly low confidences for all observations.

4 Multi-GFM Geometry Dataset with Shared Tracks

Scene Collection. We collect scenes from DL3DV10K (Ling et al., 2024), MVImgNet (Yu et al., 2023) and ScanNet++ (Yeshwanth et al., 2023) for training, validation and in-domain test sets. ETH3D (Schöps et al., 2017), LLFF (Mildenhall et al., 2019), Mip360 (Barron et al., 2022) and T&T (Knapitsch et al., 2017) are included for cross-dataset evaluation. In order to meet the GPU memory requirements of the GFM inference, we split large scenes ($\ge$ 96 views) into small scenes with random 2-96 views.

Multi-view Track Construction. To construct shared multi-view tracks independently of GFMs, we first extract local features with ALIKED (Zhao et al., 2023) and then select candidate image pairs using both sequential proximity and image retrieval. LightGlue (Lindenberger et al., 2023) is subsequently used to establish pairwise correspondences, which are geometrically verified using

MAGSAC++ (Baráth et al., 2020). A conflict-aware union–find procedure (Moulon & Monasse, 2012) then merges the verified correspondences into multi-view tracks, followed by a consistency check that removes associations with insufficient matching support.

Sample Compilation. Each split scene is processed independently by VGGT, π^3 and VGGT-Ω to obtain model-specific initial camera poses, intrinsics and depth maps. We associate the multi-view tracks with the predicted camera parameters and depths to initialize track-aligned 3D points. The resulting camera parameters and sparse 3D points, together with the shared tracks, form one sample per GFM. In total, the constructed dataset contains over 0.9M samples for training, validation and both in-domain and cross-domain testing. Detailed dataset statistics are provided in App. B.

5 Experiments

5.1 Experimental Details

Compared Methods. We compare FFBA with unrefined outputs by GFMs, classical BA implemented in PyCOLMAP v3.10 (Schönberger & Frahm, 2016), and EPO (D'Urso et al., 2026). Classical BA and our FFBA share the same inputs, while EPO performs trackless, edge-based pose optimization, without the requirement of the multi-view tracks. For EPO we use the publicly released configuration with pose-based adaptive early stopping and a maximum of 2,000 iterations.

Evaluation Metrics. We use *pixel error* and *angular error* to evaluate reprojection consistency after refinement by FFBA and classical BA. *Pixel error* is the mean Euclidean distance between predicted projections and fixed 2D observations, measured in pixels. *Angular error* is the mean angle between camera-to-point directions and the corresponding observation rays, measured in degrees($^\circ$). Camera accuracy is measured by *absolute trajectory error* (ATE), the translational (RPE-t) and rotational (RPE-R) components of adjacent-frame *relative pose error* (Sturm et al., 2012), and all-pairs relative-pose AUC (%) at 1°, 3°, and 5° (Sarlin et al., 2020). ATE and RPE-t use the ground-truth distance units, while RPE-R is measured in degrees. For efficiency analysis, we report data-preparation time, core optimization time, and end-to-end runtime in seconds (s), along with iteration counts for classical BA and EPO.

Implementation Details. We set $L = 8$ for GCGN and use 8 heads in each graph attention module. The observation and point features $\mathbf{p}_{ij}^{\ell}$ and $\mathbf{s}_{j}^{\ell}$ are 128-dimensional, while the camera and global features $\mathbf{v}_{i}^{\ell}$ and $\mathbf{g}^{\ell}$ are 256-dimensional. The encoders ϕ_P, ϕ_V, ϕ_S and ψ_P^{ℓ} contain two-layer MLPs with a 192-dimensional hidden layer and GELU activation. The hidden-layer dimensions of h_{pose} and h_{point} are $(128, 64)$ and $(64, 32)$, respectively. We set the thresholds $\tau = 0.1$, $\kappa_V = 10$, $\kappa_S = 2$, and the control factors $\alpha = 0.2$, $w_\ell = 0.8^{L-\ell}$. Training is conducted with 8 NVIDIA A100 (80 GB), and testing with an RTX 5090. Training and evaluation protocols are detailed in App. C.

5.2 In-Domain Comparison

Tab. 1 reports the comparison in reprojection and camera accuracy on three in-domain test sets. FFBA improves upon the GFM initialization (Init) across all reported metrics on all datasets. These consistent gains indicate that a shared refinement model learned across scenes can effectively correct residual geometric errors in GFM predictions, and they hold for each GFM individually (App. D). On DL3DV10K and MVImgNet, FFBA outperforms both BA and EPO in all camera metrics, which indicates that the learned feed-forward refinement generalizes across scenes and initializers within these domains, and that replacing per-scene iterative optimization with a single forward pass does not sacrifice camera accuracy when the initial geometry is relatively well conditioned.

On ScanNet++, FFBA shows superiority over BA and EPO in only a few metrics. This is expected for two reasons. First, ScanNet++ contains substantially harder samples, whose initial reprojection errors are markedly higher, so a single forward pass with a fixed update budget is less able to reach the low-error regime that iterative solvers attain through repeated linearization. Second, ScanNet++ is long-tailed in sample scale, with far fewer samples and weaker view-graph connectivity than the other two domains, so the shared refinement parameters are optimized predominantly on distributions that differ from ScanNet++. Consequently, FFBA still improves over Init in all metrics, but BA and EPO remain competitive on several camera metrics because they can exploit per-scene iteration to compensate for the harder initialization.

Table 1: **Comparison on three in-domain datasets.** Results are averaged equally over test items within each dataset. *Average* denotes the unweighted mean across datasets.

		Reprojection		Camera Pose					
Dataset	Method	Pixel ↓	Angular ↓	ATE ↓	RPE-t ↓	RPE-R ↓	AUC@1° ↑	AUC@3° ↑	AUC@5° ↑
DL3DV10K	GFM (Init)	1.1440	0.1076	0.0173	0.0126	0.0932	30.76	64.83	76.49
	BA	0.6844	0.4558	0.0201	0.0130	0.0694	36.97	67.01	77.56
	EPO	–	–	0.0174	0.0107	0.0672	33.78	66.91	77.95
	FFBA (Ours)	**0.6555**	**0.0621**	**0.0140**	**0.0080**	**0.0446**	**41.89**	**73.22**	**82.53**
MVImgNet	GFM (Init)	1.4671	0.0861	0.1026	0.1593	0.8075	9.22	42.49	59.54
	BA	0.5838	0.1773	0.1583	0.2739	1.3791	8.70	32.29	47.51
	EPO	–	–	0.1266	0.1937	1.0743	9.62	38.55	54.54
	FFBA (Ours)	**0.5806**	**0.0344**	**0.0956**	**0.1510**	**0.7052**	**14.31**	**47.62**	**63.20**
ScanNet++	GFM (Init)	2.2617	0.2640	0.0191	0.0128	0.2176	1.97	9.96	17.75
	BA	**0.5488**	2.8593	0.0118	0.0125	**0.1137**	**14.27**	**41.10**	**54.69**
	EPO	–	–	**0.0071**	0.0083	0.1240	8.48	31.83	46.80
	FFBA (Ours)	0.7310	**0.0859**	0.0099	**0.0075**	0.1164	7.81	27.67	40.56
Average	GFM (Init)	1.6243	0.1526	0.0464	0.0616	0.3727	13.99	39.09	51.26
	BA	**0.6057**	1.1641	0.0634	0.0998	0.5207	19.98	46.80	59.92
	EPO	–	–	0.0504	0.0709	0.4219	17.30	45.76	59.76
	FFBA (Ours)	0.6557	**0.0608**	**0.0398**	**0.0555**	**0.2887**	**21.34**	**49.51**	**62.10**

Table 2: **Comparison on four cross-domain datasets.** Results are averaged equally over test items within each dataset. *Average* denotes the unweighted mean across datasets.

		Reprojection		Camera Pose					
Dataset	Method	Pixel ↓	Angular ↓	ATE ↓	RPE-t ↓	RPE-R ↓	AUC@1° ↑	AUC@3° ↑	AUC@5° ↑
ETH3D	GFM (Init)	2.5174	0.2117	0.3465	0.2465	1.6636	14.91	42.39	55.25
	BA	**0.3913**	0.0825	0.3964	0.2893	**1.3317**	**36.31**	**61.46**	**69.89**
	EPO	–	–	**0.3185**	**0.2239**	1.3714	29.28	59.22	69.84
	FFBA (Ours)	0.7718	**0.0653**	0.3374	0.2354	1.5343	20.98	48.32	60.01
LLFF	GFM (Init)	1.6914	0.1055	0.1280	0.1378	0.1496	12.60	44.79	60.64
	BA	0.9026	0.0585	**0.0638**	**0.0638**	0.1124	**24.89**	**62.51**	**76.47**
	EPO	–	–	0.0673	0.0668	**0.1057**	20.91	59.38	74.04
	FFBA (Ours)	**0.8832**	**0.0567**	0.0792	0.0798	0.1418	18.83	56.13	69.55
Mip360	GFM (Init)	1.9472	0.1081	0.0665	0.0522	0.3999	17.94	54.19	68.62
	BA	**0.4472**	1.1232	**0.0410**	0.0430	**0.1411**	**53.80**	**79.83**	**86.81**
	EPO	–	–	0.0527	**0.0349**	0.2408	28.75	64.66	76.67
	FFBA (Ours)	0.6251	**0.0349**	0.0581	0.0398	0.3139	27.67	63.06	75.13
T&T	GFM (Init)	1.5890	0.1247	0.0647	0.0269	0.3434	18.62	57.46	71.26
	BA	**0.6643**	0.0653	0.0710	0.0246	0.3784	8.86	44.45	62.44
	EPO	–	–	0.0640	0.0240	0.3344	12.96	50.56	66.90
	FFBA (Ours)	0.7564	**0.0592**	**0.0625**	**0.0234**	**0.2838**	**23.54**	**61.31**	**73.86**
Average	GFM (Init)	1.9362	0.1375	0.1514	0.1159	0.6391	16.02	49.71	63.94
	BA	**0.6013**	0.3324	0.1430	0.1052	**0.4909**	**30.97**	**62.06**	**73.90**
	EPO	–	–	**0.1256**	**0.0874**	0.5131	22.97	58.46	71.86
	FFBA (Ours)	0.7591	**0.0540**	0.1343	0.0946	0.5684	22.75	57.20	69.64

5.3 Cross-Domain Comparison

Tab. 2 presents results on four cross-domain datasets. FFBA still improves upon Init across all metrics on four datasets, demonstrating that the learned refinement transfers beyond the training domains rather than merely fitting the in-domain error distribution. On T&T, FFBA achieves the best results on almost all metrics and is the only refinement method that consistently improves AUC over Init, indicating that the learned updates remain reliable when the initial geometry is already reasonably consistent. On LLFF and ETH3D, FFBA attains the lowest reprojection errors and improves over Init on all metrics, whereas BA can even degrade some camera metrics relative to Init, showing that classical per-scene optimization can be unstable under harder cross-domain initializations. On Mip360, FFBA again achieves the lowest angular error with competitive AUC, although BA retains an advantage on some camera metrics. Overall, FFBA is the only method that consistently improves over Init on every cross-domain dataset, confirming its robustness to domain shift even though it does not dominate every individual metric.

Table 4: **Ablation study of the key components of FFBA on in-domain datasets.**

Setting	Reprojection		Camera Pose					
	Pixel ↓	Angular ↓	ATE ↓	RPE-t ↓	RPE-R ↓	AUC@1° ↑	AUC@3° ↑	AUC@5° ↑
Full model	**0.6557**	**0.0608**	**0.0398**	0.0555	**0.2887**	**21.34**	**49.51**	**62.10**
(a) w/o confidence mechanism	0.6810	0.0643	0.0399	**0.0553**	0.2934	20.05	47.87	60.52
(b) w/o geometry conditioning	1.3894	0.1318	0.0448	0.0601	0.3597	14.50	40.48	52.89
(c) w/o learned update steps	0.8287	0.0792	0.0433	0.0577	0.3195	18.09	44.06	55.86

5.4 Efficiency Analysis

Table 3: Comparison of mean inference time (s).

Scope	Method	Data Prepar.↓	Optim.↓	# Iter.	Total↓
In-domain	BA	10.49	2.62	69	13.11
	EPO	**3.65**	18.82	1360	22.47
	FFBA	10.44	**0.07**	–	**10.52**
Cross-domain	BA	15.08	2.23	52	17.31
	EPO	**4.03**	11.08	966	15.11
	FFBA	14.96	**0.09**	–	**15.05**

Tab. 3 compares the average inference efficiency on both in-domain and cross-domain test datasets. The key advantage of our method lies in optimization: by replacing per-scene iterative linearization with a single forward pass of a learned refinement network, FFBA reduces the optimization cost by more than an order of magnitude relative to both BA and EPO. This is in contrast to EPO, which benefits from a trackless design and thus has a much shorter data-preparation stage, but whose pose-based adaptive early stopping still requires a drastically larger number of iterations. This leads to a dominant optimization cost that largely outweighs its preparation advantage. Classical BA, sharing the same initial geometry and tracking inputs as our method, likewise incurs a far higher optimization cost than our approach, since it must repeatedly linearize and solve the reprojection objective for each scene. As a result, although our method does not reduce the data-preparation cost, its near-constant and extremely lightweight optimization stage consistently translates into the best total inference runtime across both scopes. See App. D for detailed results.

5.5 Ablation Study

We ablate three components of FFBA, each removed individually from the full model with all other settings unchanged: (a) *w/o confidence mechanism* removes confidence prediction, observation filtering (Eq. 9) and confidence weighting in the loss (Eq. 16); (b) *w/o geometry conditioning* removes the encoded reprojection residual from the observation node update (Eq. 7); (c) *w/o learned update steps* removes h_γ^ℓ (Eq. 13) and applies the geometric increments with unit step sizes. As illustrated in Table 4, the full model achieves the best results on almost all metrics. The variant without the confidence mechanism shows modest degradation across most metrics. Without geometry conditioning, both pixel and angular errors increase substantially, and camera accuracy deteriorates across all metrics. Replacing learned update steps also yields worse results across all metrics, suggesting that adaptive scaling helps coordinate geometric updates.

6 Conclusion

We presented FFBA, a feed-forward bundle adjustment framework that recasts per-scene iterative optimization as a single forward pass of a learned, geometry-aware refinement network. Given initial cameras, intrinsics and 3D geometry from a geometric foundation model together with fixed multi-view tracks, FFBA builds a sparse heterogeneous graph, and refines poses, focal lengths and 3D points through stacked geometric graph aggregation layers with confidence-based observation filtering. Trained end-to-end with a confidence-weighted reprojection loss and no ground-truth supervision, FFBA consistently improves over GFM initializations, remains competitive with BA and EPO under domain shift, and cuts optimization cost by more than an order of magnitude, yielding an end-to-end efficiency advantage expected to grow on large-scale scenes.

Limitations and Future Work. FFBA performs a fixed number of updates without convergence guarantees, refines only focal lengths while keeping other intrinsics fixed, and scales memory with the number of observations. Future work will explore adaptive updates and error-aware stopping, broader intrinsic calibration, more memory-efficient graph representations, and tighter integration with the GFM and tracking stages.

Ethics Statement

This work uses existing multi-view datasets and pretrained GFMs, without collecting new images or conducting human-subject studies. Copyright in the source images remains with the respective rights holders, and their use and redistribution are subject to the applicable dataset licenses and terms. Any release of our processed data will also be subject to these restrictions. Applications involving people or private spaces should respect consent and privacy.

Reproducibility Statement

Secs. 3–5 describe the model architecture, data-construction pipeline, training settings, and evaluation protocol. Appendices A–C provide pseudocode, dataset statistics, and additional training details. We use fixed random seeds for dataset partitioning and model training to support reproducibility. Classical BA and FFBA share the same initial geometry and multi-view tracks for controlled comparison. Code will be released upon acceptance.

REFERENCES

Dániel Baráth, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1304–1312, 2020.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

Lucas Brynte, José Pedro Iglesias, Carl Olsson, and Fredrik Kahl. Learning structure-from-motion with graph attention networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4808–4817, 2024.

Yutong Chen, Yiming Wang, Xucong Zhang, Sergey Prokudin, and Siyu Tang. GGPT: Geometry-grounded point transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28959–28968, 2026.

Nikolaus Demmel, Christiane Sommer, Daniel Cremers, and Vladyslav Usenko. Square root bundle adjustment for large-scale reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11723–11732, 2021.

Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jérôme Revaud. MASt3R-SfM: A fully-integrated solution for unconstrained structure-from-motion. In *International Conference on 3D Vision*, pp. 1–10, 2025.

Mattia D'Urso, Christian Sormann, Mattia Rossi, and Friedrich Fraundorfer. Boosting 3D foundation models with edge-based pose optimization. In *European Conference on Computer Vision*, pp. 479–498, 2026.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, pp. 5574–5584, 2017.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78:1–78:13, 2017.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, pp. 71–91, 2024.

Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *IEEE/CVF International Conference on Computer Vision*, pp. 5987–5997, 2021.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. In *IEEE/CVF International Conference on Computer Vision*, pp. 17627–17638, 2023.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 38(4):29:1–29:14, 2019.

Pierre Moulon and Pascal Monasse. Unordered feature tracking made fast and easy. In *Conference on Visual Media Production*, pp. 1, 2012.

Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L. Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pp. 58–77, 2024.

Linfei Pan, Johannes L. Schönberger, and Marc Pollefeys. Global structure-from-motion meets feed-forward reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21880–21890, 2026.

Daniel Safari. Matrix-free shared intrinsics bundle adjustment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27017–27026, 2025.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4938–4947, 2020.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3260–3269, 2017.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3):835–846, 2006.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.

Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations*, 2019.

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16558–16569, 2021.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pp. 298–372. Springer, 2000.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21686–21697, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes L. Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21486–21499, 2026a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10510–10522, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026b.

Simon Weber, Nikolaus Demmel, Tin Chon Chan, and Daniel Cremers. Power bundle adjustment for large-scale 3D reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 281–289, 2023.

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21924–21935, 2025.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. MVImgNet: A large-scale dataset of multi-view images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9150–9161, 2023.

Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter C. Y. Chen, Qingsong Xu, and Zhengguo Li. ALIKED: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023.

A Refinement Procedure

Algorithm 1 details the GCGN refinement stage of FFBA, starting from the initial geometry and fixed tracks produced by data preparation. After graph construction and node initialization (Sec. 3.3), the GGA layers implement the feed-forward mapping in Eq. 4. Each layer updates all observation features from the current reprojection residuals and node context, then selects reliable observations for camera and point updates without changing the input tracks. The updated global feature determines scene-level step sizes, which combine with the decoded corrections and node masks to produce the geometric state for the next layer. Training evaluates the confidence-weighted loss over all layers (Eq. 16), whereas inference returns only the final geometry $\boldsymbol{\xi}^L$.

Algorithm 1 GCGN forward refinement within FFBA

Require: initial geometry $\boldsymbol{\xi}^0$, tracks $\mathcal{U}$, and learned parameters $\boldsymbol{\theta}$
Ensure: refined geometry $\boldsymbol{\xi}^L$

Construct $\mathcal{G}$ from observed associations using Eq. 5.
Initialize observation, camera, point and global features as in Sec. 3.3.
for $\ell = 1, \ldots, L$ **do**
  1. Observation update and filtering
  Compute $\Delta_{ij}^{\ell-1}$ for $(i, j) \in \mathcal{O}$ using Eq. 2.
  Update all $\mathbf{p}_{ij}^{\ell}$ using Eq. 7.
  Predict c_{ij}^{ℓ} and obtain $\mathcal{O}_{\text{act}}^{\ell}$ using Eqs. 8–9.
  Compute $m_i^{V,\ell}$ and $m_j^{S,\ell}$ using Eq. 10.
  2. Camera and point updates
  Update $\mathbf{v}_i^{\ell}$ and $\mathbf{s}_j^{\ell}$ from the same observation features using Eq. 11.
  3. Global node update
  Aggregate supported camera and point features into $\mathbf{g}^{\ell}$ using Eq. 12.
  Predict $\gamma_X^{\ell}, \gamma_t^{\ell}, \gamma_R^{\ell}$ using Eq. 13.
  4. Joint geometric update
  Decode $\Delta\mathbf{t}_i^{\ell}$, $\Delta\boldsymbol{\omega}_i^{\ell}$, β_i^{ℓ}, and $\Delta\mathbf{X}_j^{\ell}$ using Eq. 14.
  Apply masked geometric corrections to obtain $\boldsymbol{\xi}^{\ell}$ using Eq. 15.
end for
if training **then**
  Compute the multi-layer loss using Eq. 16.
end if
return $\boldsymbol{\xi}^L$

B Dataset Details

Tab. 5 summarizes the resulting dataset scale and sparse-graph sizes. The three in-domain datasets contain 181,474 source scenes, yielding 302,512 split image sets and 907,536 samples across three GFMs. Using a fixed seed, we randomly select approximately 9K samples for validation and another 9K for in-domain testing, with the remainder used for training. The four external datasets contribute 49 source scenes, resulting in 579 cross-domain test samples.

To assess problem scale, we report the average (avg) and 95th percentile (P95) of the numbers of camera nodes V, point nodes S, and observation nodes P per sample in Tab. 5. Camera and point counts reflect the size of the geometry to be optimized, while observation counts indicate the number of reprojection constraints. The average summarizes problem size across samples, whereas P95 characterizes larger problems with greater computational and memory requirements. These statistics capture variations in problem scale and constraint counts across datasets, including differences between the in-domain and cross-domain sets.

Table 5: **Dataset scale and sparse-graph statistics.**

	Scale			Camera Nodes		Point Nodes		Observation Nodes	
Dataset	Scenes	Split Scenes	Samples	V_{avg}	V_{P95}	S_{avg}	S_{P95}	P_{avg}	P_{P95}
DL3DV10K	9,802	95,965	287,895	41	93	7,369	17,381	63,732	153,431
MVImgNet	170,679	170,679	512,037	30	32	5,149	13,842	32,119	89,777
ScanNet++	993	35,868	107,604	37	56	3,019	6,446	29,282	59,851
ETH3D	25	26	78	35	74	9,796	23,933	40,955	91,385
LLFF	8	8	24	38	60	8,026	10,981	83,328	150,324
Mip360	9	55	165	41	94	15,555	39,099	87,247	190,241
T&T	7	104	312	40	80	11,076	20,363	67,894	121,081

C Training and Evaluation Protocol

Training. FFBA is trained from scratch with AdamW (Loshchilov & Hutter, 2019) on 8 A100 GPUs for about 9.7 days. We define one epoch as 500 optimizer updates, using an effective batch size of 16 samples. The learning rate increases linearly from 1×10^{-6} to 2×10^{-4} over the first 5 epochs (2.5k steps), remains constant until epoch 50 (25k steps), and then follows a staged cosine decay to 4×10^{-7} at epoch 1,000 (500k steps). The checkpoint with the lowest loss on the validation set is selected and used for all experiments.

Evaluation. Camera accuracy is measured against the ground-truth poses provided by each dataset. Estimated cameras are aligned to the reference by a Sim(3) transformation (Umeyama, 1991) before computing ATE and RPE (Sturm et al., 2012). AUC is the area under the accuracy curve of the all-pairs relative pose error, where the error of a pair is the maximum of its rotation and translation-direction errors (Sarlin et al., 2020). Each test item pairs one image set with one GFM initialization. For both in-domain and cross-domain comparisons, metrics are averaged equally over test items within each dataset, followed by an unweighted mean across datasets. Results by initializer apply the same rule to the corresponding test items. The three GFMs contribute equal numbers of items. Metrics undefined for an item are excluded from that metric only, rather than assigned zero.

D Results by GFM Initializer

Tabs. 1–3 report results by dataset. In this section, we report results by GFM in Tabs. 6–8.

In-domain comparison. Tab. 6 reports the in-domain comparison for each GFM. FFBA improves upon Init across all metrics for all three initializers, and attains the lowest angular error, ATE, RPE-t and RPE-R in every case, which confirms that the gains in Tab. 1 are not driven by a single GFM. On π^3, whose initialization is the most accurate, FFBA outperforms BA and EPO on every metric. On VGGT and VGGT-Ω, FFBA remains the best in ATE and RPE, whereas BA or EPO is marginally ahead in AUC at some thresholds while trailing behind in angular error and RPE-R.

Table 6: **In-domain comparison by GFM initializer.** Results are averaged equally over test items within each dataset and then over the three in-domain datasets.

		Reprojection		Camera Pose					
Model	Method	Pixel ↓	Angular ↓	ATE ↓	RPE-t ↓	RPE-R ↓	AUC@1° ↑	AUC@3° ↑	AUC@5° ↑
VGGT	GFM (Init)	1.7430	0.1723	0.0547	0.0716	0.4037	12.34	34.86	45.81
	BA	**0.6065**	1.7157	0.0667	0.1059	0.5549	**19.01**	45.16	**58.29**
	EPO	–	–	0.0556	0.0785	0.4413	13.87	41.15	55.50
	FFBA (Ours)	0.6804	**0.0628**	**0.0450**	**0.0627**	**0.3203**	18.66	**45.44**	58.23
π^3	GFM (Init)	1.2906	0.1119	0.0378	0.0515	0.3125	13.68	43.30	58.27
	BA	0.6088	0.6372	0.0614	0.0970	0.4953	20.48	47.97	61.24
	EPO	–	–	0.0447	0.0613	0.3645	19.35	49.68	63.68
	FFBA (Ours)	**0.5991**	**0.0553**	**0.0353**	**0.0491**	**0.2492**	**23.81**	**56.76**	**70.16**
VGGT-Ω	GFM (Init)	1.8392	0.1735	0.0466	0.0616	0.4020	15.95	39.11	49.70
	BA	**0.6018**	1.1406	0.0622	0.0964	0.5121	20.46	**47.27**	**60.22**
	EPO	–	–	0.0508	0.0729	0.4597	18.67	46.46	60.11
	FFBA (Ours)	0.6876	**0.0644**	**0.0391**	**0.0547**	**0.2966**	**21.54**	46.32	57.89
Average	GFM (Init)	1.6243	0.1526	0.0464	0.0616	0.3727	13.99	39.09	51.26
	BA	**0.6057**	1.1641	0.0634	0.0998	0.5207	19.98	46.80	59.92
	EPO	–	–	0.0504	0.0709	0.4219	17.30	45.76	59.76
	FFBA (Ours)	0.6557	**0.0608**	**0.0398**	**0.0555**	**0.2887**	**21.34**	**49.51**	**62.10**

Cross-domain comparison. Tab. 7 presents the cross-domain comparison for each GFM. FFBA still improves upon Init across all metrics for all three initializers, so the transfer observed in Tab. 2 holds regardless of which GFM provides the initial geometry. The initializer has a larger effect than the refinement method: the VGGT rows show several-fold larger ATE and RPE than the π^3 and VGGT-Ω rows for every method. On VGGT-Ω, FFBA achieves the best or second-best result on almost all metrics and is the only refinement method besides EPO that improves ATE and RPE-t over Init, whereas BA degrades both. On VGGT and π^3, BA and EPO obtain higher AUC, while FFBA retains the lowest angular error and competitive ATE.

Table 7: **Cross-domain comparison by GFM initializer.** Results are averaged equally over the four cross-domain datasets.

		Reprojection		Camera Pose					
Model	Method	Pixel ↓	Angular ↓	ATE ↓	RPE-t ↓	RPE-R ↓	AUC@1° ↑	AUC@3° ↑	AUC@5° ↑
VGGT	GFM (Init)	2.1755	0.1571	0.2904	0.2046	1.1179	11.50	42.17	56.53
	BA	**0.6135**	**0.0584**	0.2661	0.1876	**0.9425**	**29.26**	**59.25**	**71.10**
	EPO	–	–	**0.2593**	**0.1696**	0.9640	15.96	50.36	65.00
	FFBA (Ours)	0.8218	0.0586	0.2699	0.1769	1.0364	17.33	50.42	63.63
π^3	GFM (Init)	1.9326	0.1349	0.0975	0.0831	0.5316	14.73	48.40	63.12
	BA	**0.5962**	0.0596	0.0809	0.0603	**0.3440**	**32.15**	**63.56**	**75.23**
	EPO	–	–	**0.0691**	**0.0520**	0.4027	23.77	60.08	73.30
	FFBA (Ours)	0.7417	**0.0528**	0.0770	0.0610	0.4729	22.62	57.52	69.88
VGGT-Ω	GFM (Init)	1.7006	0.1205	0.0662	0.0598	0.2678	21.81	58.55	72.19
	BA	**0.5943**	0.8792	0.0821	0.0676	0.1862	**31.49**	63.37	75.38
	EPO	–	–	**0.0484**	**0.0406**	**0.1725**	29.19	**64.93**	**77.29**
	FFBA (Ours)	0.7138	**0.0507**	0.0561	0.0460	0.1960	28.32	63.67	75.40
Average	GFM (Init)	1.9362	0.1375	0.1514	0.1159	0.6391	16.02	49.71	63.94
	BA	**0.6013**	0.3324	0.1430	0.1052	**0.4909**	**30.97**	**62.06**	**73.90**
	EPO	–	–	**0.1256**	**0.0874**	0.5131	22.97	58.46	71.86
	FFBA (Ours)	0.7591	**0.0540**	0.1343	0.0946	0.5684	22.75	57.20	69.64

Efficiency. Tab. 8 reports the efficiency comparison for each GFM. The optimization time of FFBA stays between 0.07 and 0.09 s regardless of the initializer and scope, whereas BA and EPO vary with the initializer through their iteration counts. Thus, the order-of-magnitude reduction in optimization cost reported in Tab. 3 holds for every GFM. Data preparation is nearly identical for FFBA and BA because they share the same front end, and FFBA consequently achieves the best end-to-end runtime for almost all three initializers.

Table 8: **Inference efficiency by GFM initializer.** Results are averaged equally across datasets within each scope. Runtimes are in seconds.

Scope	GFM	Method	Data Preparation ↓	Optimization ↓	Iterations	End-to-End Runtime ↓
In-domain	VGGT	BA	10.21	2.61	71	12.82
		EPO	**3.42**	19.94	1448	23.36
		FFBA (Ours)	10.17	**0.07**	–	**10.24**
	π^3	BA	11.07	2.60	67	13.67
		EPO	**4.34**	17.80	1265	22.14
		FFBA (Ours)	11.00	**0.08**	–	**11.07**
	VGGT-Ω	BA	10.18	2.66	70	12.84
		EPO	**3.18**	18.71	1367	21.90
		FFBA (Ours)	10.16	**0.07**	–	**10.23**
	Average	BA	10.49	2.62	69	13.11
		EPO	**3.65**	18.82	1360	22.47
		FFBA (Ours)	10.44	**0.07**	–	**10.52**
Cross-domain	VGGT	BA	14.82	2.29	55	17.11
		EPO	**3.25**	11.57	1025	14.82
		FFBA (Ours)	14.70	**0.09**	–	**14.79**
	π^3	BA	15.84	2.31	52	18.15
		EPO	**5.99**	12.32	1048	18.32
		FFBA (Ours)	15.72	**0.09**	–	**15.81**
	VGGT-Ω	BA	14.58	2.09	50	16.67
		EPO	**2.85**	9.35	825	**12.20**
		FFBA (Ours)	14.46	**0.09**	–	14.56
	Average	BA	15.08	2.23	52	17.31
		EPO	**4.03**	11.08	966	15.11
		FFBA (Ours)	14.96	**0.09**	–	**15.05**

E AI USE STATEMENT

Large language models (LLMs) were used solely to assist with grammar, formatting, and language refinement during the preparation of this manuscript. All scientific ideas, methodology, experimental design, analysis, and interpretation of results are the original work of the authors.