VGGT-SLAM 2.0: Real-time Dense Feed-forward Scene Reconstruction

Dominic Maggio[1], Luca Carlone[1*]

MIT–SPARK

RSS 2026

Presenter: 송우석

- **github:** https://github.com/MIT-SPARK/VGGT-SLAM
- **paper:** https://roboticsconference.org/program/papers/51/

RAISE LAB

Index

- Problem & Motivation
- Method
- Experiments
- Limitation

Paradigm Shift

- simultaneously create 3D reconstruction of a scene and localize camera
  → foundational task in robotics & computer vision
- paradigm shift
  - classical multi-view geometry & optimization → utilize feed-forward geometric foundational models
- geometric foundational models + tools from classical SLAM
  → produce a much simpler SLAM system that is both easier to use and maintain
  → produce dense RGB point clouds while not requiring camera calibration be known beforhand

VGGT-SLAM (recent work)

- extends VGGT to SLAM (to process potentially thousands of frames)
  - VGGT is limited to about 60 frames on RTX 4090 due to memory constraints
  - utilize submaps (create, align), overlapping frame
  - optimize submap alignment on the SL(4) manifold due to projective ambiguity (causing submap distortions)
    → SE(3), Sim(3) are subsets of SL(4)
- problem
  - higher dimensional 15-DoF alignment → rapid drift, degenerate in planar scenes
  - factor graph only estimate homohraphy between submaps(not keyframe-level) → suboptimal
  - image retrieval for loop closure depends on external retrieval network (e.g. SALAD)

- shear
- stretch
- perspective distortion

→ not included in Sim(3)

Related Works

- Classical mapping techniques
  - Classical scene reconstruction methods

    → rely on tracking and associating features across multiple frames (+ backend optimization)
  - Learning based methods
    - incorporated in SLAM systems such as for optical flow and for neural scene representations
  - several works
    - perform classical optimization of projective alignment on the SL(3) group, and [23] discusses synchronization on the SL(4) group
- Feed-forward scene reconstruction
  - DUSt3R, MASt3R
    - take in a pair of uncalibrated monocular images and produce a 3D scene reconstruction
    - paradigm shift towards using Geometric foundation models (GFM) for SLAM
  - advanced works

    - **MASt3R-SFM:** builds on MASt3R to perform global optimization over multiple images

      → but quickly grows computationally expensive as the number of frames increases
    - **MASt3R-SLAM:** the first work to demonstrate real- time SLAM with a GFM
    - **MASt3R-Fusion:** incorporates IMU and GNSS as additional sensor measurements through classical optimization

  ⇒ all of these works are restricted by GPU memory usage to a relatively small number of frames
- Other works
  - VGGT-SLAM, VGGT-Long, SING3R-SLAM, MegaSaM, TTT3R, TALO

Related Works

- Attention layer analysis

  - **Perception Encoder:** conducts a study of the layers of a vision encoder for **downstream tasks**
  - **Easi3R (+1):** shows how an understanding of DUSt3R's layers can be used to enable **dynamic object detection**
  - **recent works:** exploring the layers of VGGT for tasks(e.g. **noise suppression, geometric interpretation**)

⇒ so far no works have studied the layers of VGGT for image retrieval verification

VGGT-SLAM 2.0 ✓

1. enforcing that the two overlapping frames in submap alignment must have the same position, rotation, and calibration, and solve for a consistent scale factor

   → to remove the issue of high-dimensional drift and planar degeneracy of VGGT submap alignment

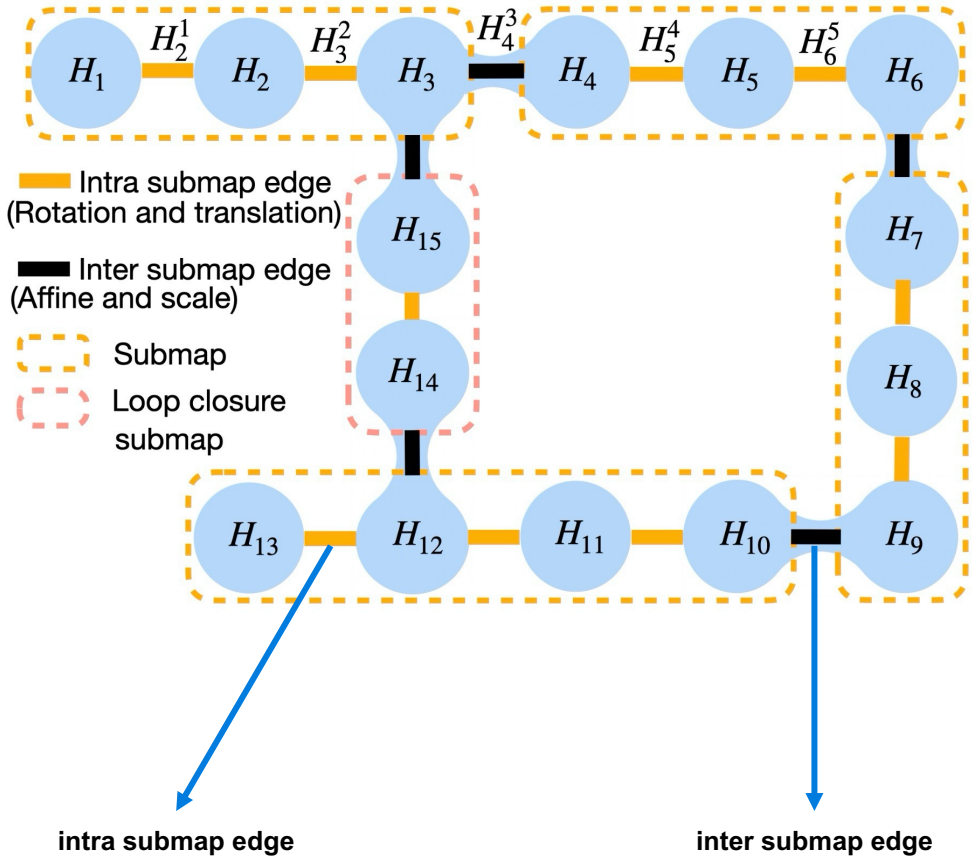
2. new factor graph structure

3. attention layers of VGGT

   → to verify the validity of a potential retrieved frame for loop closures (prevent FP, more loop closure)

VGGT-SLAM 2.0

- Factor Graph



- Node
  - all keyframes
- Edge
  - **intra:** edge inside a submap
    - non-identity rotation & translation

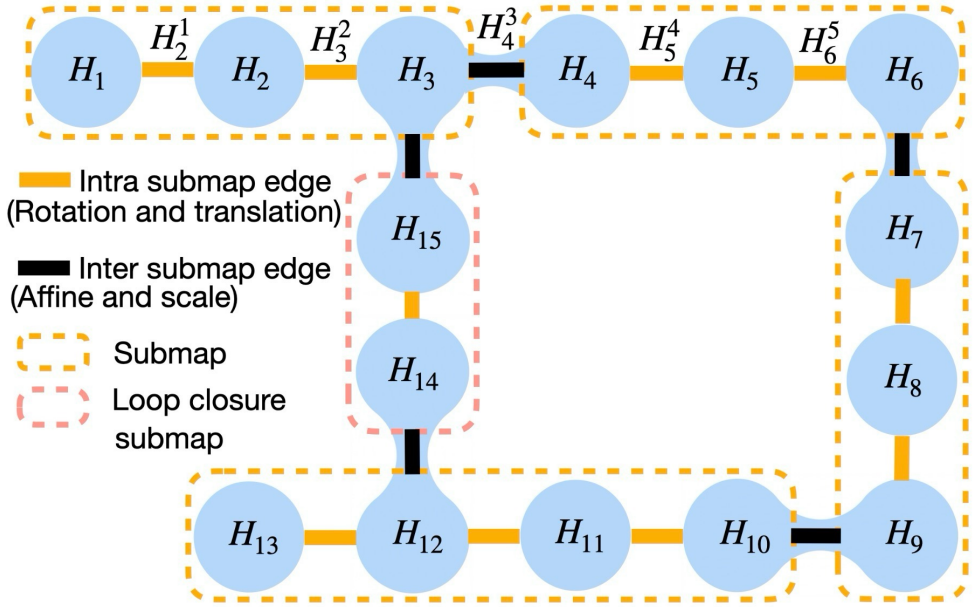
    → help correct potential drift in VGGT's translation and rotation estimates
  - **inter:** edge between submaps
    - non-identity calibration (i.e. affine) & scale variables

    → make each submap's estimate of the scale and calibration of the overlapping frame be identical (we don't know what the true calibration is)

⇒ Since both of these transformations are subsets of SL(4), we use SL(4) nodes for factor graph optimization

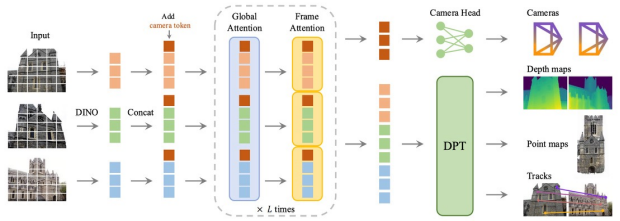
VGGT-SLAM 2.0

- Factor Graph



respective camera frame. Thus, the points $\mathbf{X}$ for frame i (*i.e.*, $\mathbf{X}_i$) are computed through back projection using only $\mathbf{K}_i^{-1}$ and $\mathbf{D}_i$, where $\mathbf{X}_i \in \mathbb{R}^{3\times h\times w}$ for image of size $h \times w$.

use predicted intrinsics & detph instead of Pointmap of VGGT



Model architecture of VGGT

$$\mathbf{X}_i^a = \mathbf{H}_j^i \mathbf{X}_j^b, \tag{1}$$

- $\mathbf{t}$: translation (3-DOF)
- $\mathbf{R}$: rotation (3-DOF)
- s: scale (1-DOF)
- $\mathbf{K}$: camera calibration (affine, 5-DOF)
- $\mathbf{v}$: projective components (3-DOF)

- full 4×4 homography matrix

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{KR} & t \\ \mathbf{v}^T & s \end{bmatrix}, \tag{2}$$

- Intra submap alignment

$$\mathbf{H}_j^i = \mathbf{T}_i^{-1}\mathbf{T}_j. \tag{3}$$

transformation matrix (rotation + translation)

- Inter submap alignment

$$\mathbf{H}_j^i = \begin{bmatrix} \mathbf{K}_i^{-1}\mathbf{K}_j & 0 \\ \mathbf{0}^T & s \end{bmatrix}, \tag{4}$$

→ relative translation and rotation between the overlapping frames is zero

1. since the overlapping frames are identical, corresponding 3D points between them are trivially known
2. unlike VGGT-SLAM, with our new problem formulation, we can easily estimate relative scale separately from other parameters of the homography

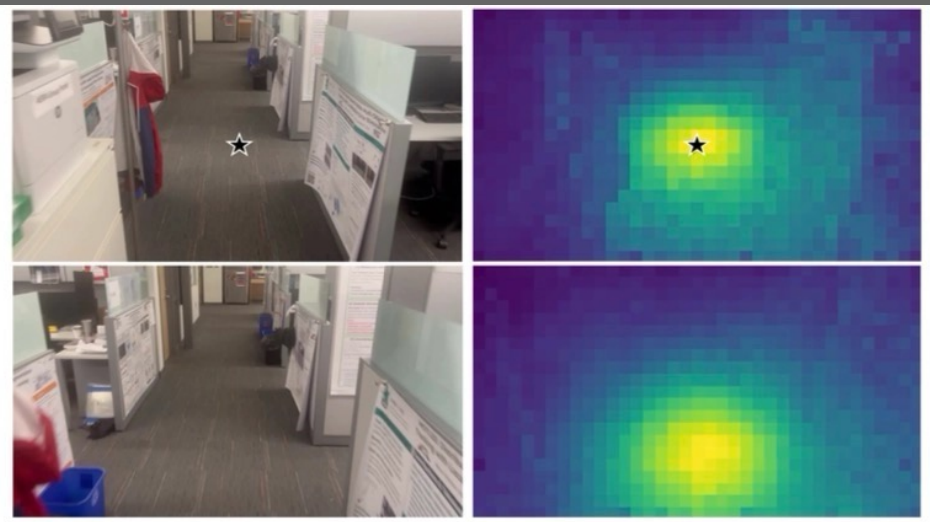
→ corresponding points seen in overlapping keyframes must have the same scale

RAISE LAB

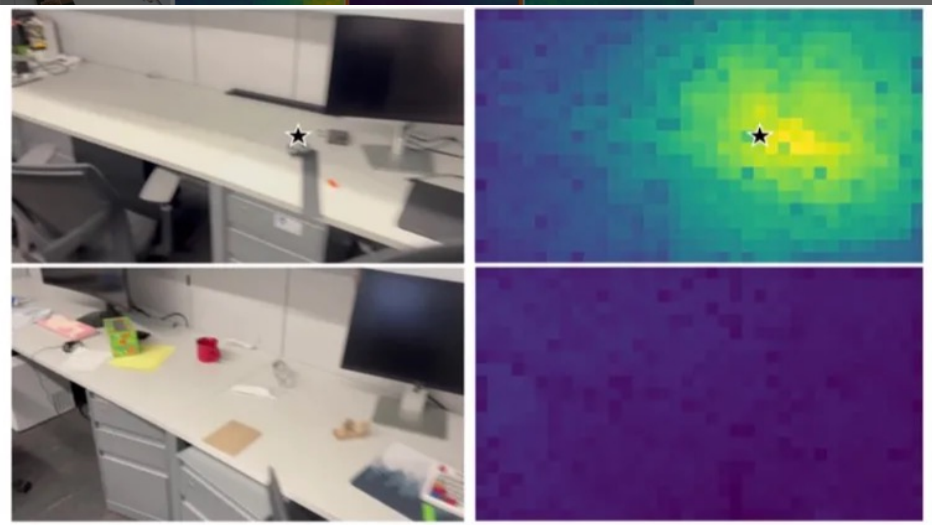
VG

- Im

…cted key token, with respect to all other query tokens



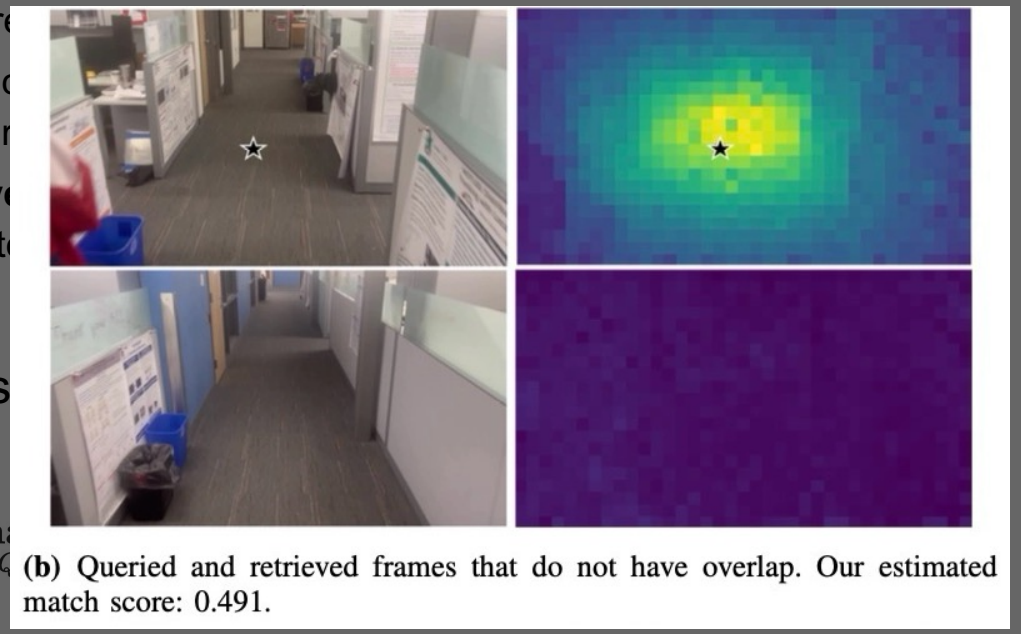
(a) Queried and retrieved frames that have overlap. Our estimated match score: 1.026.



(c) Queried and retrieved frames that do not have overlap, but SALAD incorrectly estimated a match between them. Our estimated match score: 0.550.

- Analysis
  - **Layer 22** shows larger attention values at the corr…
  - not … near…
  - **Lay…** whit…
- Match s…

$$\gamma_t = \max_{q \in Q}$$



(b) Queried and retrieved frames that do not have overlap. Our estimated match score: 0.491.

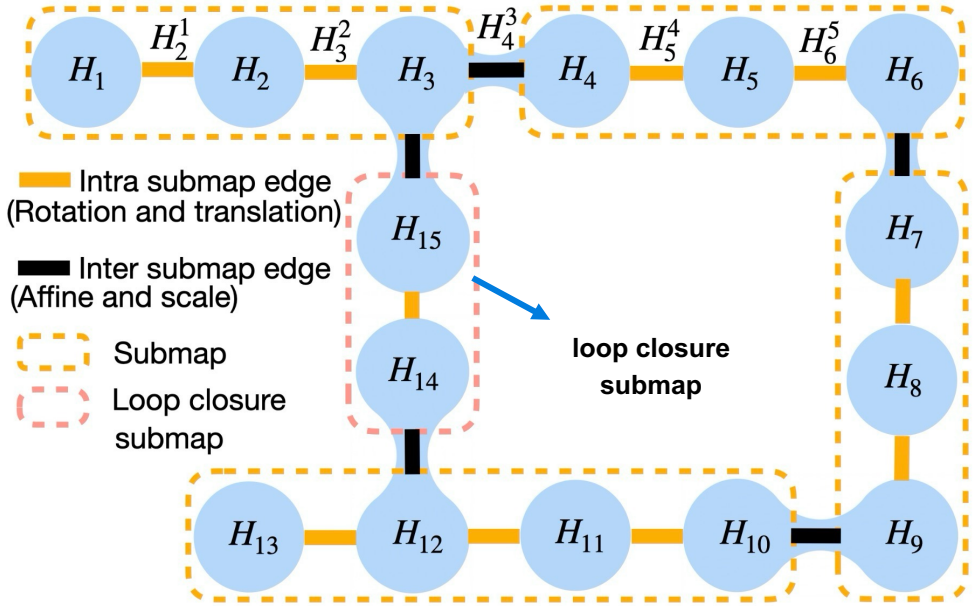
- $Q^{(1)}, K^{(1)}$: all query, key tokens (average across all heads) for image 1
- $Q^{(2)}$: all query tokens for image 2

$$\alpha_{match} = \text{Mean}_{\text{top25\%}}(\{\gamma_t\}) \quad (6)$$

RAISE LAB

VGGT-SLAM 2.0

- Loop closures and global optimization



index set of constraints (odometry + loop closures)

- Global Optimization (in VGGT-SLAM)

$$\Omega_{ij}^{\mathbf{H}} \in \mathbb{R}^{15\times 15}$$

$$\hat{\mathcal{H}} = \underset{\mathbf{H}\in \mathrm{SL}(4)}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{L}} \left\| \underbrace{\operatorname{Log}\left(\mathbf{H}_i^{-1}\mathbf{H}_j\left(\mathbf{H}_j^i\right)^{-1}\right)}_{r_{ij}} \right\|_{\Omega_{ij}^{\mathbf{H}}}^2,$$

$$\min_{\{H_i\}} \left[\sum_{\mathcal{E}_{\text{intra}}} \|r_{ij}^{\text{intra}}\|^2 + \sum_{\mathcal{E}_{\text{inter}}} \|r_{ij}^{\text{inter}}\|^2 + \sum_{\mathcal{E}_{\text{loop}}} \|r_{ij}^{\text{loop}}\|^2 \right]$$

(from ChatGPT)

- loop closure submap
  - H_{15}: retrieved frame from a prior submap
  - H_{14}: corresponding query frame in the current submap

  → inter submap edge is then added between the frame of the loop closure submap and its respective overlapping frame

- image retrieval (SALAD & verification)

  0.95 0.85

  → relax the image retrieval threshold from SALAD to get more potential loop closure candidates

  → reject false positives in challenging environments

- Recovering global reconstruction

$$\mathbf{P}_i = \mathbf{K}_i^{3\times 4}(\mathbf{H}_i^w)^{-1} \qquad (7)$$

Transformation of camera poses via homography Using $\mathbf{H}_j^i$, the camera poses can be corrected using the following [25]: $\mathbf{P}_i = \mathbf{H}_j^{i\,-1}\mathbf{P}_j$, where $\mathbf{P} \in \mathbb{R}^{4\times 4}$ is the camera matrix created from the poses and intrinsic estimates from VGGT. We can then decompose $\mathbf{P}$ to recover the camera pose.

projection matrix to extract calibration & pose

(from VGGT)

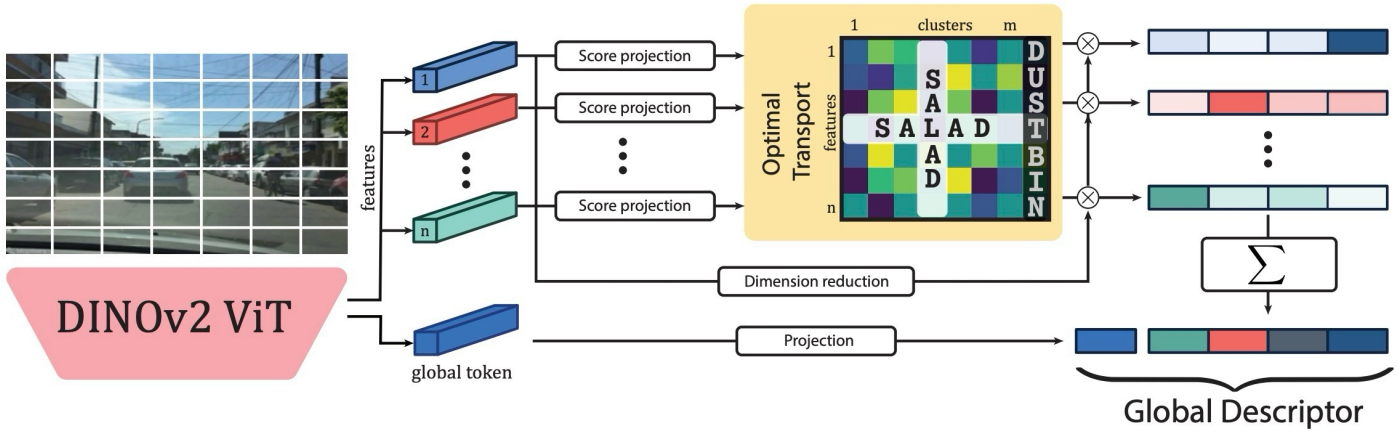
RAISE LAB

Experiments

- Pose estimation
- Loop closure verification
- Open-set semantic task
- Real-time experiments and timing results

Setup

- parameters
  - same parameter used by VGGT-SLAM
  - SALAD threshold of 0.95 (0.80 in VGGT-SLAM)
  - retrieval verification threshold of 0.85 for α_{match}



Model architecture of SALAD

Pose estimation

- datasets
  - TUM RGB-D benchmark
- metric
  - ATE RMSE
- etc
  - submap size: 32 frames (used by VGGT-SLAM)
- result
  - achieves the best overall average pose error of 4.1 cm out of all uncalibrated baselines

    → 23% lower than VGGT-SLAM

    → 22% lower than recent ViSTA-SLAM

		360	desk	desk2	floor	plant	room	rpy	teddy	xyz	avg
Calibrated	ORB-SLAM3 [43]	X	0.017	0.210	X	0.034	X	X	X	0.009	-
	DeepV2D [44]	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
	DeepFactors [45]	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
	DPV-SLAM [46]	0.112	0.018	0.029	0.057	0.021	0.330	0.030	0.084	0.010	0.076
	DPV-SLAM++ [46]	0.132	0.018	0.029	0.050	0.022	0.096	0.032	0.098	0.010	0.054
	GO-SLAM [47]	0.089	0.016	0.028	0.025	0.026	0.052	0.019	0.048	0.010	0.035
	DROID-SLAM [20]	0.111	0.018	0.042	0.021	0.016	0.049	0.026	0.048	0.012	0.038
	MASt3R-SLAM [42]	0.049	0.016	0.024	0.025	0.020	0.061	0.027	0.041	0.009	0.030
Uncalibrated	CUT3R [25]	0.174	0.592	0.546	0.662	0.467	0.911	0.051	0.845	0.129	0.486
	SLAM3R [48]	0.211	0.861	0.967	0.790	0.755	1.013	0.063	0.986	0.185	0.648
	VGGT-Long [8]	0.063	0.059	0.045	0.108	0.057	0.172	0.056	0.137	0.051	0.083
	MapAnything [27]	0.084	0.058	0.067	0.072	0.049	0.209	0.045	0.059	0.025	0.074
	DROID-SLAM [20]	0.202	0.032	0.091	0.064	0.045	0.918	0.056	0.045	0.012	0.158
	MASt3R-SLAM [42]	0.070	0.035	0.055	0.056	0.035	0.118	0.041	0.114	0.020	0.060
	ViSTA-SLAM [7]	0.104	0.030	0.030	0.070	0.052	0.067	0.023	0.080	0.015	0.052
	VGGT-SLAM Sim(3) [5]	0.123	0.040	0.055	0.254	0.022	0.088	0.041	0.032	0.016	0.074
	VGGT-SLAM SL(4) [5]	0.071	0.025	0.040	0.141	0.023	0.102	0.030	0.034	0.014	0.053
	VGGT-SLAM 2.0	0.050	0.025	0.029	0.102	0.026	0.063	0.026	0.038	0.014	0.041

TABLE I. camera pose estimation on TUM RGB-D

Method	ScanNet++	ScanNetV2 [52]	Waymo [53]
DROID-SLAM [20]	0.623	0.217	7.535
MASt3R-SLAM [42]	0.346	0.106	16.610
VGGT-SLAM [5]	1.081	0.852	13.166
VGGT-SLAM 2.0	**0.182**	**0.073**	**1.295**

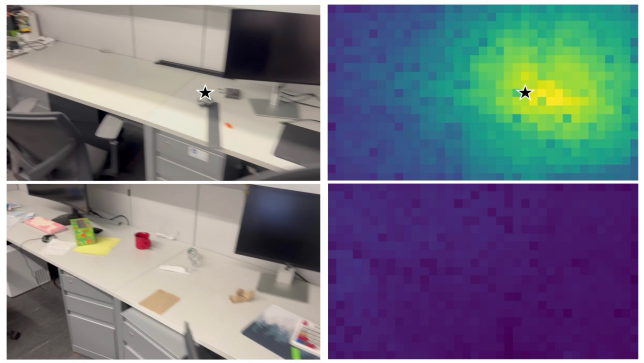
TABLE II. camera pose estimation on on ScanNet++, ScanNetV2, and Waymo

Loop closure verification

- datasets
  - Clio datasets (three scenes)
- options
  - without using retrieval verification
  - with using the proposed retrieval verification
- etc
  - SALAD threshold of 0.80 (used by VGGT-SLAM)
- result
  - image retrieval verification method can help achieve more loop closures while preventing false positives
  - office scene without verification led to false positive loop closures causing the reconstruction to diverge due to challenging similar looking areas
  - VGGT-SLAM 2.0 still substantially surpasses VGGT-SLAM pose accuracy on all three Clio scenes as VGGT-SLAM diverges due to high dimensional drift

Dataset	Without Retrieval Verification	With Retrieval Verification
Clio Cubicle	2	**5**
Clio Apartment	0	**9**
Clio Office	X	**4**

TABLE III. NUMBER OF LOOP CLOSURES ON THE CLIO DATASETS



(c) Queried and retrieved frames that do not have overlap, but SALAD incorrectly estimated a match between them. Our estimated match score: 0.550.

Method	Cubicle	Apartment	Office
VGGT-SLAM [5]	1.092	2.191	5.816
VGGT-SLAM 2.0 w/o ret. verif.	0.032	0.208	1.287
VGGT-SLAM 2.0 with ret. verif.	**0.020**	**0.078**	**0.479**

TABLE IV. RMSE OF ATE (METERS) ON CLIO SCENES

Loop closure verification

- datasets
  - LaMAR HGE phone validation scene
- metric
  - Recall@1 score
- retrieval methods
  - SALAD
  - NETVLAD
- options
  - SALAD threshold of 0.80 (used by VGGT-SLAM)
- result
  - verification procedure improves the results of both methods
  - layer 22 performs the best, the worst performing layers are its neighbors(21, 23)

	Without Retrieval Verification	With Retrieval Verification
SALAD [10]	88.45	**90.13**
NetVLAD [54]	85.92	**89.08**

TABLE V. RECALL@1 SCORES ON LAMAR HGE PHONE DATASET

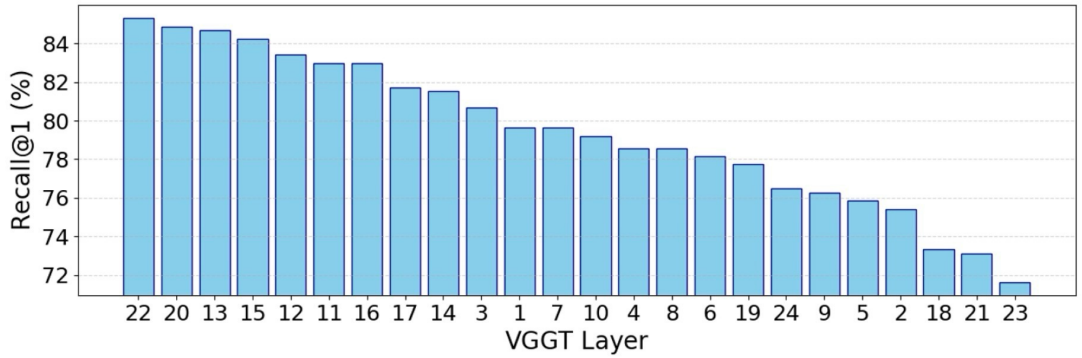
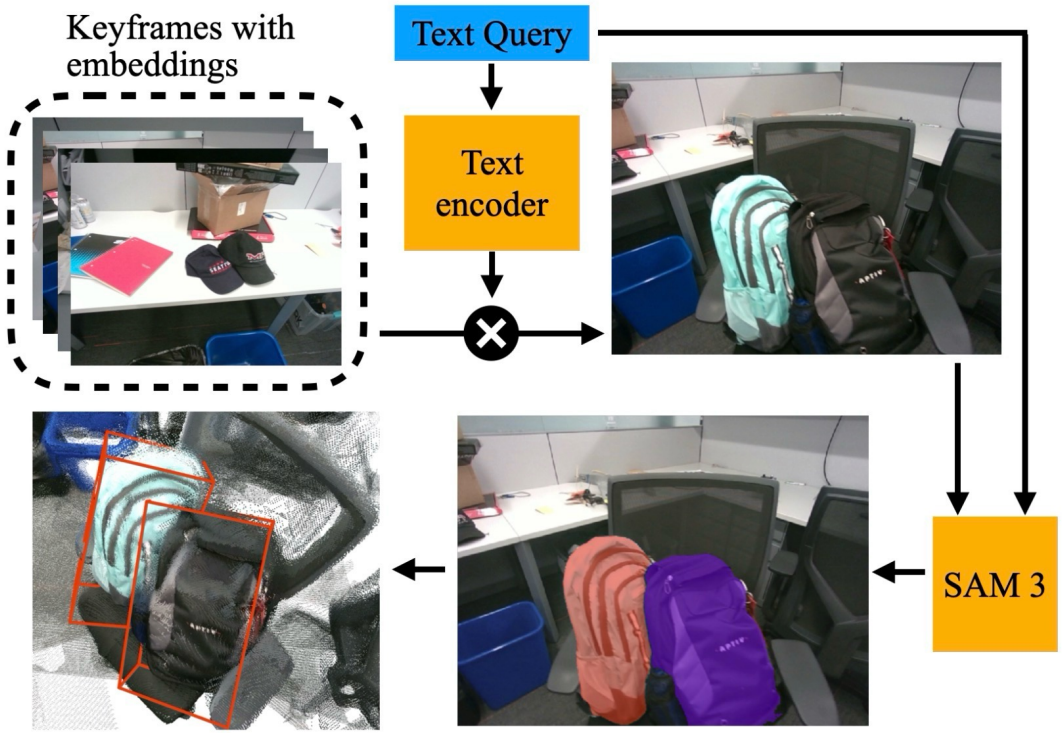
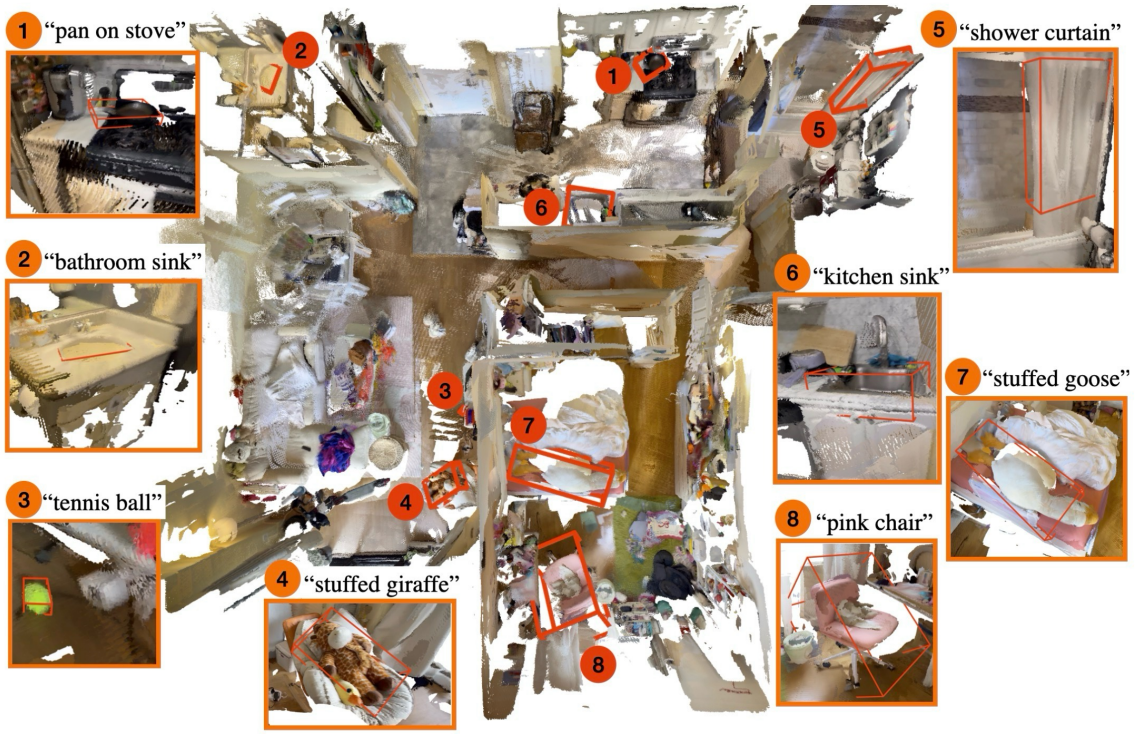


Fig. 6. Recall @1 on LaMAR dataset for all VGGT layers

Open-set semantic task



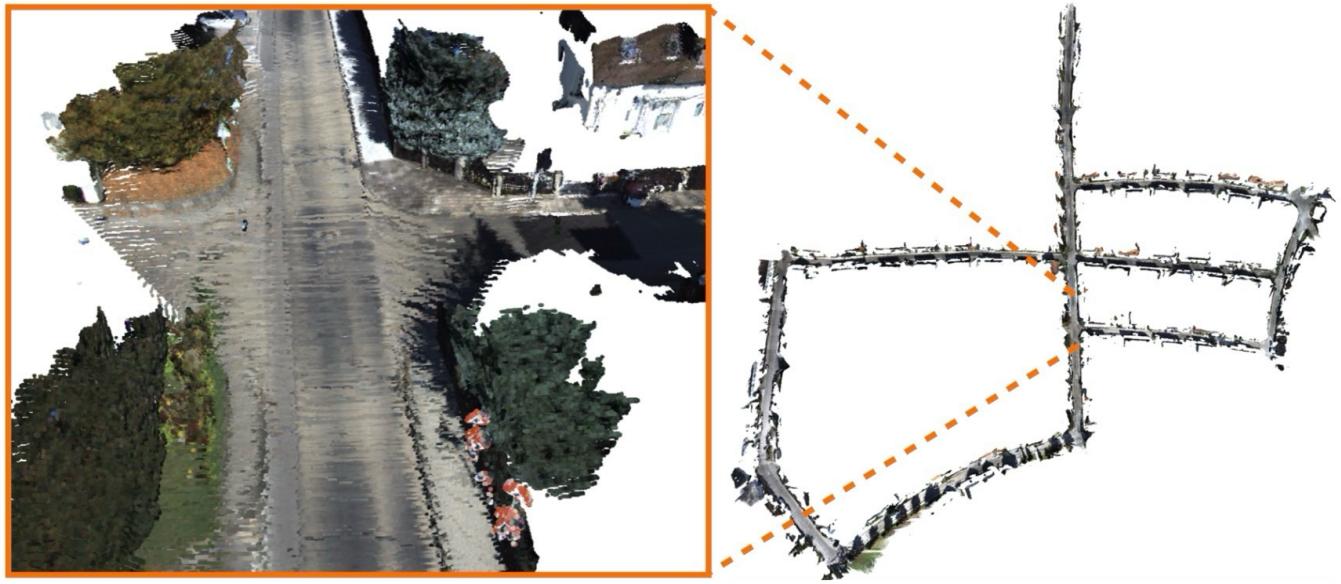
open-set querying on the Clio cubicle dataset



custom apartment using an iPhone camera

Open-set semantic task

4200 square foot barn using images captured from an iPhone



Reconstruction of Kitti outdoor driving sequence 05

Real-time experiments and timing results

- submap of size 16 frames
- using 3090 GPU
- runtime
  - w.o open-set: 8.4 FPS
  - w open-set: 6.3 FPS

  (FPS = total time / number of unique frames in the submap)

Stage	total time per submap (ms)
VGGT Inference	1248 ms
Keyframe Detection	176 ms
Loop Closure Detection	128 ms
Backend Optimization	8 ms
Semantics (optional)	368 ms

AVERAGE RUNTIME PER SUBMAP (IN MILLISECOND)

- **MASt3R-SLAM:** 7.2 FPS (comparable)
- **VGGT-SLAM:** 6.9 FPS.

- on Jetson Thor (submaps of size 4)

  VGGT-SLAM 2.0: 3.5 FPS

Limitation

- reconstruction which only saw a plain white wall

  → causing VGGT reconstruction to fail

- our backend factor-graph optimization is only over poses and not points

  → simplifies the SLAM pipeline but causes artifacts from misaligned 3D

- VGGT reconstruct all frames w.r.t. to first frame and also use the first frame to compute the scale factor of a submap

  → first frame of the submap is unequally influential