3D Reconstruction

VGGT: Visual Geometry Grounded Transformer

CVPR 2025 (Best Paper Award)

Jungwoo Yoon[1], Jiye Park[2], Geonhak Song[2], Junhyoung Lee[2],
Junhan Zang[2], Jinyeong Chae[2], Euiju Heo[2], Pseudo Lab[3]

Paper | arXiv | Code | Youtube

SOTA
AI
Review
Project

CVPR Nashville JUNE 11-15, 2025

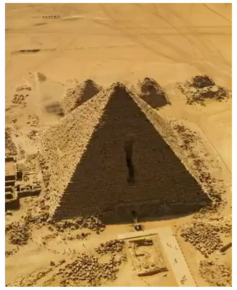
3D Reconstruction

VGGT: Visual Geometry Grounded Transformer

CVPR 2025 (Best Paper Award)

Jianyuan Wang[1, 2], Minghao Chen[1, 2], Nikita Karaev[1, 2],
Andrea Vedaldi[1, 2], Christian Rupprecht[1], David Novotny[2]
[1]Visual Geometry Group, University of Oxford, [3]Meta AI

32 Views



Paper arXiv Code Youtube

PseudoLab

SOTA
AI
Review
Project

Index

Visual Geometry Grounded Transformer

INTRODUCTION

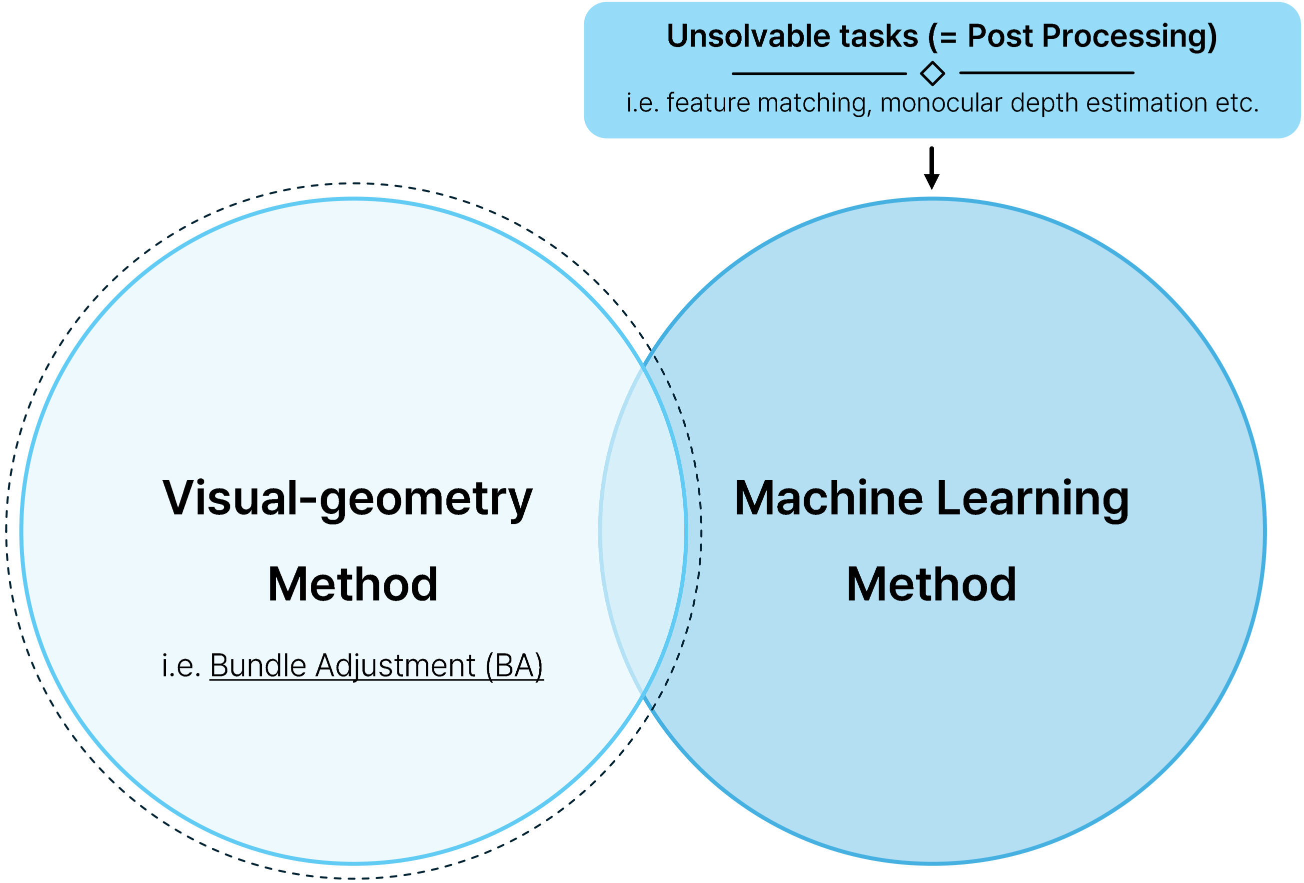
Problem Statement

3D Reconstruction

“3D reconstruction is the process of generating digital 3D representations of scenes and objects from inputs like images, video, or other sensor data.”

Problem Statement

- Problem 1. Limit of Traditional Method



⇓

Visual-geometry → increasing complexity and computational cost

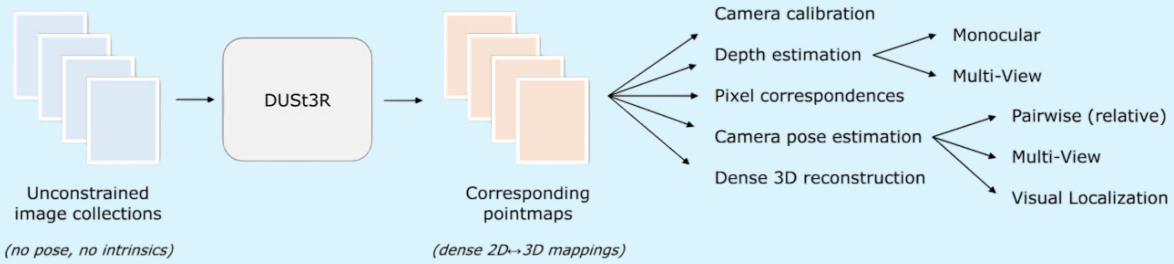
Problem Statement

- Problem 2. Limit of DL-based Method

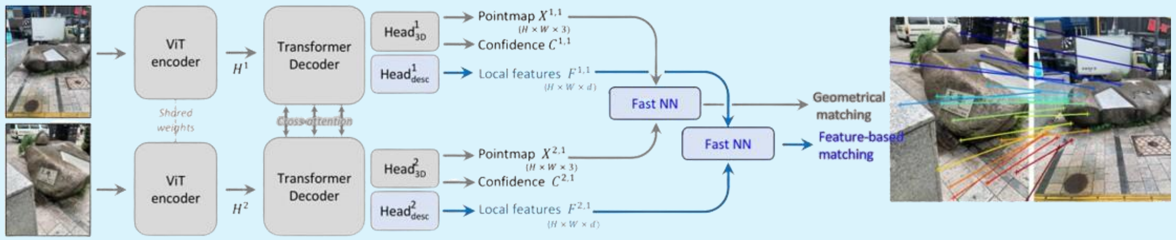
Neural Network

which directly addresses 3D reconstruction

DUST3R



MAST3R



Post processing

- **Limit of Input: processing 2 images at once** (Pairwise reconstruction)
- **Reliance on post-processing** (-> costly iterative post-optimization)

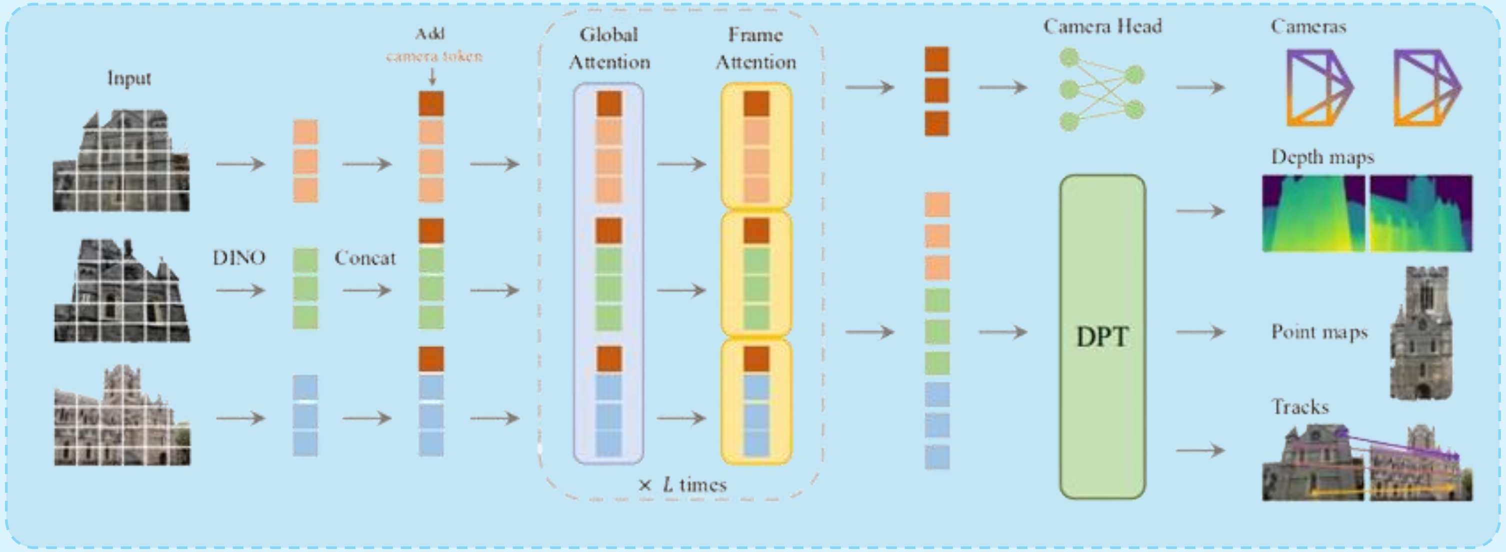
Proposed Method

VGGT: Visual Geometry Grounded Transformer

Input

1 ~ hundreds of input views of a scene

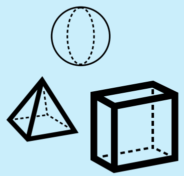
Large Transformer Model



single feed-forward

Output

A full set of 3D attribultes



Camera parameters

Depth maps

Point maps

3D point tracks

...

a single feed-forward neural network based on standard large transformer that performs 3D reconstruction from one to even hundreds of input views of a scene, predicting a full set of 3D attributes.

Contribution

- **1. Overwhelming Speed & Efficiency**

  VGGT predicts all key 3D attributes **in a single forward pass, in seconds**

- **2. Competitive Performance**

  Its prediction is **directly usable**,
  while usually outperforming alternatives even without further processing.
  Also, achieves **state-of-the-art performance** when combined with BA post-processing

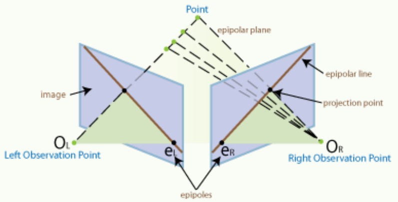
- **3. Versatility & Extensibility**

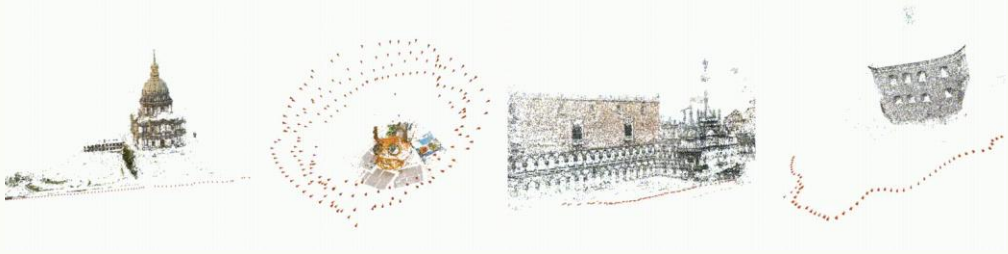
  Based on a standard large transformer as a **shared versatile backbone**,
  VGGT can be fine-tuned to solve new, specific tasks, **enhancing downstream tasks**.

Related Work

Structure from Motion (SfM)

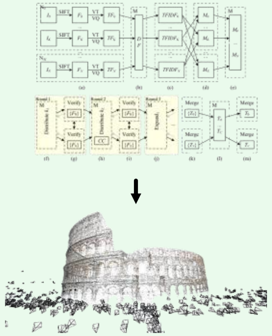
a **classic computer vision** problem that involves **estimating camera parameters and reconstructing sparse point clouds** from a set of images of a static scene captured from different viewpoints



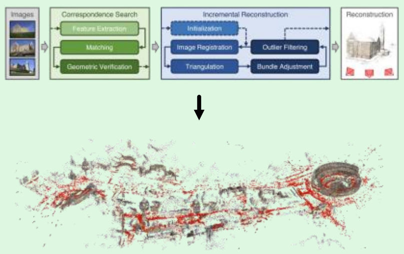


Traditional SfM

A traditional approach **consisted of multiple stages**, including image matching, triangulation, and bundle adjustment



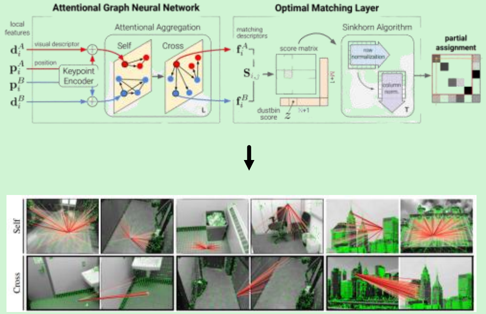
Building Rome in a day
(ACM 2011)



structure from motion revisited (COLMAP)
(CVPR 2016)

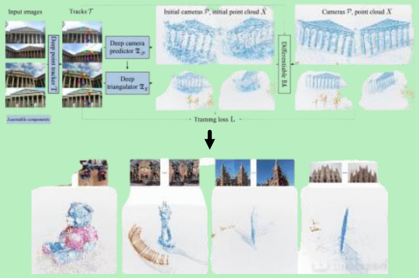
Recent SfM

A recent approach improved with **deep learning-based components**, focusing on 2 primary stages (**keypoint detection, image matching**) in particular.



Superglue: Learning feature matching with graph neural networks
(CVPR 2020)

end-to-end differentiable SfM



VGGSfM: visual geometry grounded deep structure from motion
(CVPR 2024 Highlight)

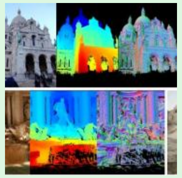
Multi-view Stereo (MVS)

a **classic computer vision** problem which aims to **densely reconstruct the geometry of a scene** from multiple overlapping images, typically **assuming known camera parameters, which are often estimated with SfM**

Traditional MVS

Traditional handcrafted MVS

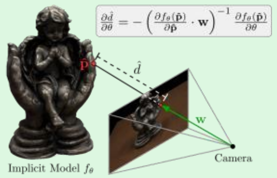
An approach to compute **geometric consistency across multiple images**, typically by **comparing image patches to estimate depth.**



Pixelwise view selection for unstructured multi-view stereo (CVPR 2016)

Global Optimization MVS

An approach formulating **the entire scene reconstruction as a single optimization** problem, aiming **to find a globally consistent 3D model** by minimizing a total energy function



Differentiable volumetric rendering (CVPR 2020)

→

Learning-based MVS

A modern approaches that **use deep neural networks to learn a mapping from input images to a 3D representation**, autonomously inferring **geometric and appearance information.**

DUST3R

DUSt3R: Geometric 3D Vision Made Easy (CVPR 2024)

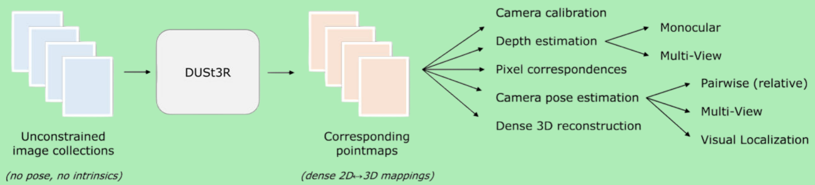
⇓

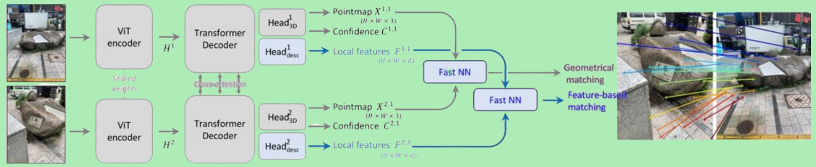
directly estimate aligned dense point clouds from a pair of views, without requiring camera parameters

⇑

MAST3R

Grounding Image Matching in 3D with MASt3R (ECCV 2024 Oral)





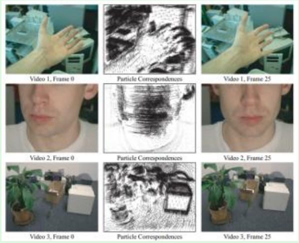
Tracking-Any-Point (TAP)

a **modern computer vision** problem aiming to predict and track points of interest(= 2D correspondences) in all other frames including across video sequences including dynamic motions, when given a video and some 2D query points.

* **TAP-Vid** : a proposed three benchmarks for TAP task and a simple baseline method, later improved to TAPIR

Origin

Particle Video



Particle video: Long-range motion estimation using point trajectories (IJCV 2008)

PIPs

Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories (ECCV 2022 Oral)

→

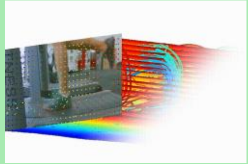
Recent TAP

CoTracker

CoTracker
(ECCV 2024)

A model which utilize correlations between different points to track through occlusions

DOT



DOT
(CVPR 2024)

A model which enable dense tracking through occlusions

TAPTR

TAPTR
(v1 – ECCV 2024
V2 – NeurIPS 2024)

A model proposed an end-to-end transformer for TAP task

LocoTrack

LocoTrack
(ECCV 2024
)

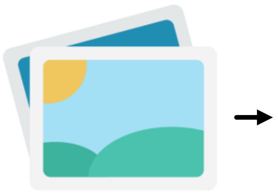
A model extended commonly used pointwise features to nearby regions

Methods

Overview & Notations

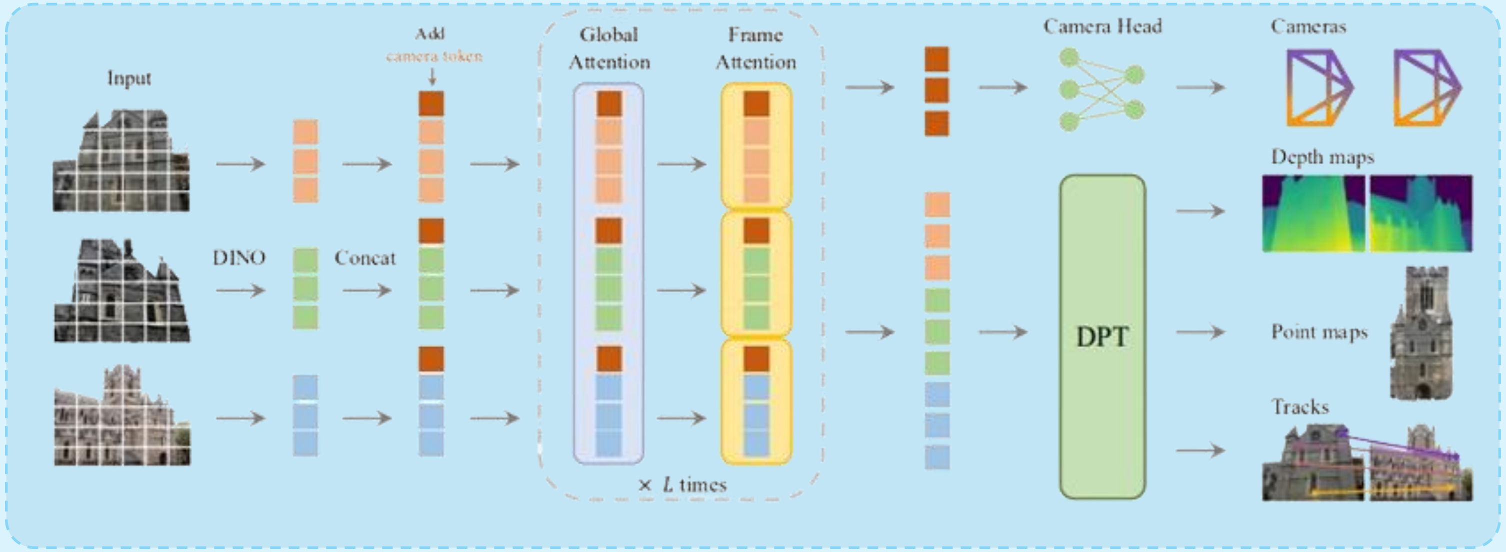
VGGT: Visual Geometry Grounded Transformer

Input



1 ~ hundreds of input views of a scene

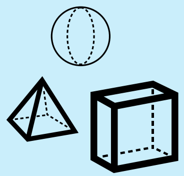
Large Transformer Model



single feed-forward

Output

A full set of 3D attribultes



Camera parameters
Depth maps
Point maps
3D point tracks
...

RGB Image

$I_i (I_i \in \mathbb{R}^{3\times H\times W})$

↓

RGB Image **Sequence**

$(Ii)_{i=1}^{N}$

Transformer

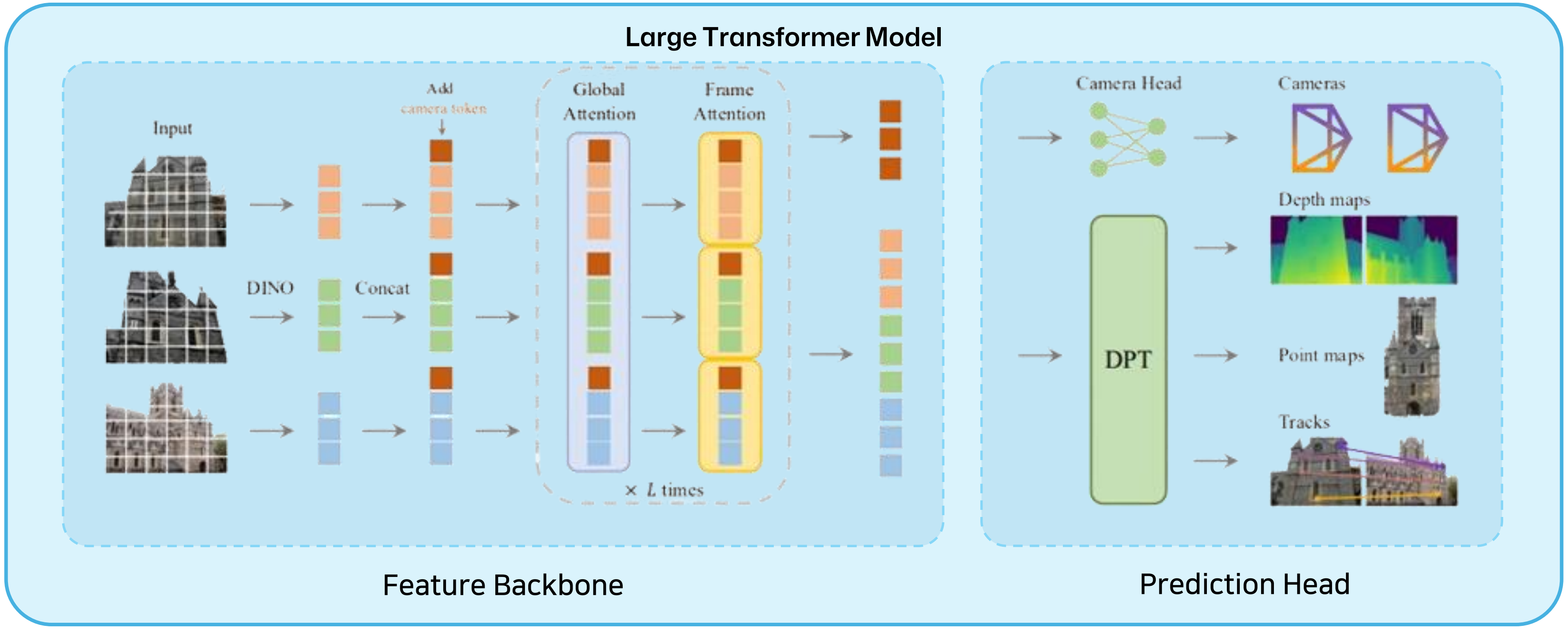
The transformer maps each RGB image I_I

$$f((Ii)_{i=1}^{N}) = (g_i, D_i, P_i, T_i)_{i=1}^{N}$$

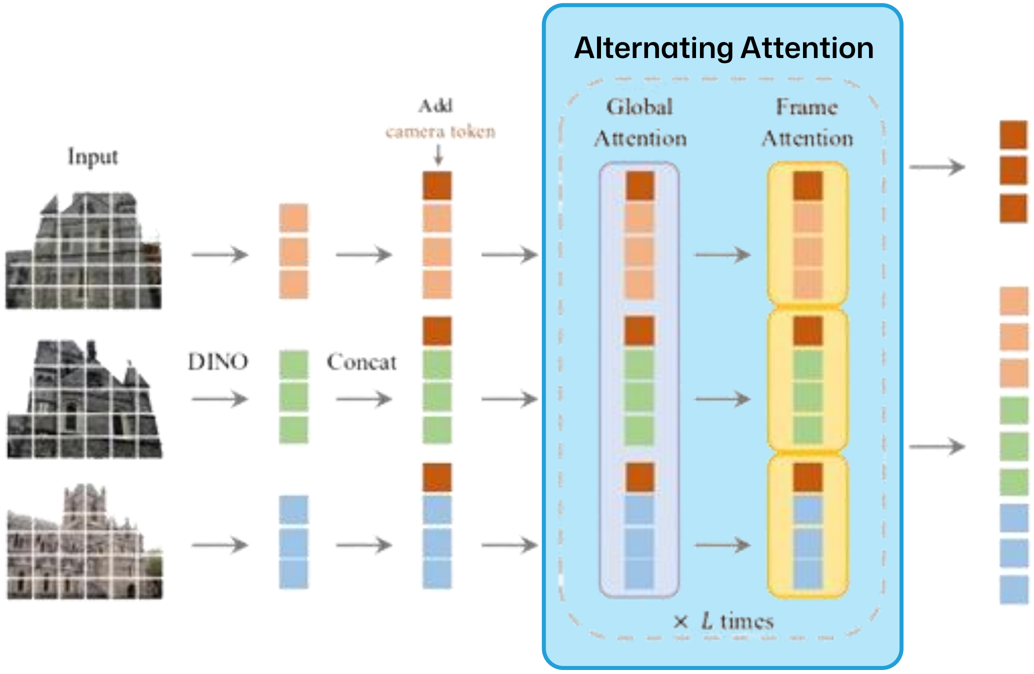
to its **camera parameters** $g_i \in \mathbb{R}$,
its **depth map** $D_i \in \mathbb{R}^{H\times W}$,
its **point map** $P_i \in \mathbb{R}^{3\times H\times W}$,
and **a grid** $T_i \in \mathbb{R}^{C\times H\times W}$
of C-dimensional features for point tracking

Feature Backbone

VGGT: Visual Geometry Grounded Transformer



Feature Backbone



X 24 times

Step 1. Image Processing

① Image Patching

Initially **patchify** each input image $\boldsymbol{I}$ into
a set of K tokens $\boldsymbol{t}^{\boldsymbol{I}} \in \mathbb{R}^{K \times C}$ through DINO.

② Image Processing

Subsequently **process** the combined
set of image tokens from al frames,
$\boldsymbol{t}^{\boldsymbol{I}} = \cup_{i=1}^{N} \{\boldsymbol{t}_i^{\boldsymbol{I}}\}$ through the **transformer network structure**,
<u>alternating frame-wise and global self-attention layers</u>.

Alternating Attention

An attention module which makes the transformer
focus within each frame and globally in an alternate fashion.

Frame-wise Self Attention

Attends to the <u>**tokens** t_k^I</u>
<u>within each frame</u> separately

Global Self Attention

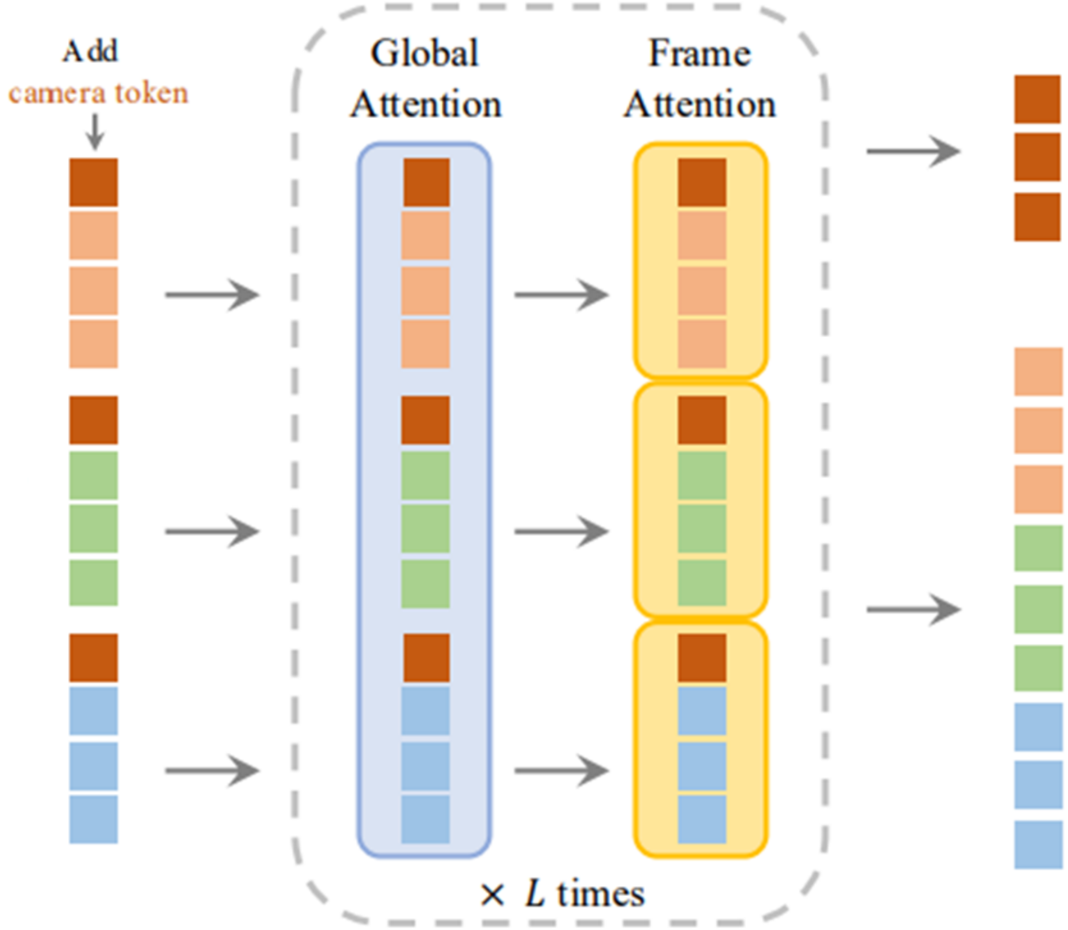
Attends to the <u>**tokens** t^I</u>
<u>across all frames</u> jointly

$\Downarrow$

Strike a **balance** between <u>integrating information across different images</u>
and <u>normalizing the activations for the tokens within each image</u>.

3. Method

Prediction Heads



Step 2. Prediction

① Input

Input Image I_i is tokenized into t_i^I,
augmented with **an additional camera token** $t_i^g \in \mathbb{R}^{1\times C}$
and **four register tokens** $t_i^R \in \mathbb{R}^{4\times C}$

↓

The concatenation of $(t_i^I, t_i^g, t_i^R j)_{i=1}^N$

↓

AA Transformer
(Global Self Attention + Frame-wise Self Attention)

② Output

$(\hat{t}_i^I, \hat{t}_i^g, \hat{t}_i^R)_{i=1}^N$

Used

Discarded

Learnable Tokens

↓

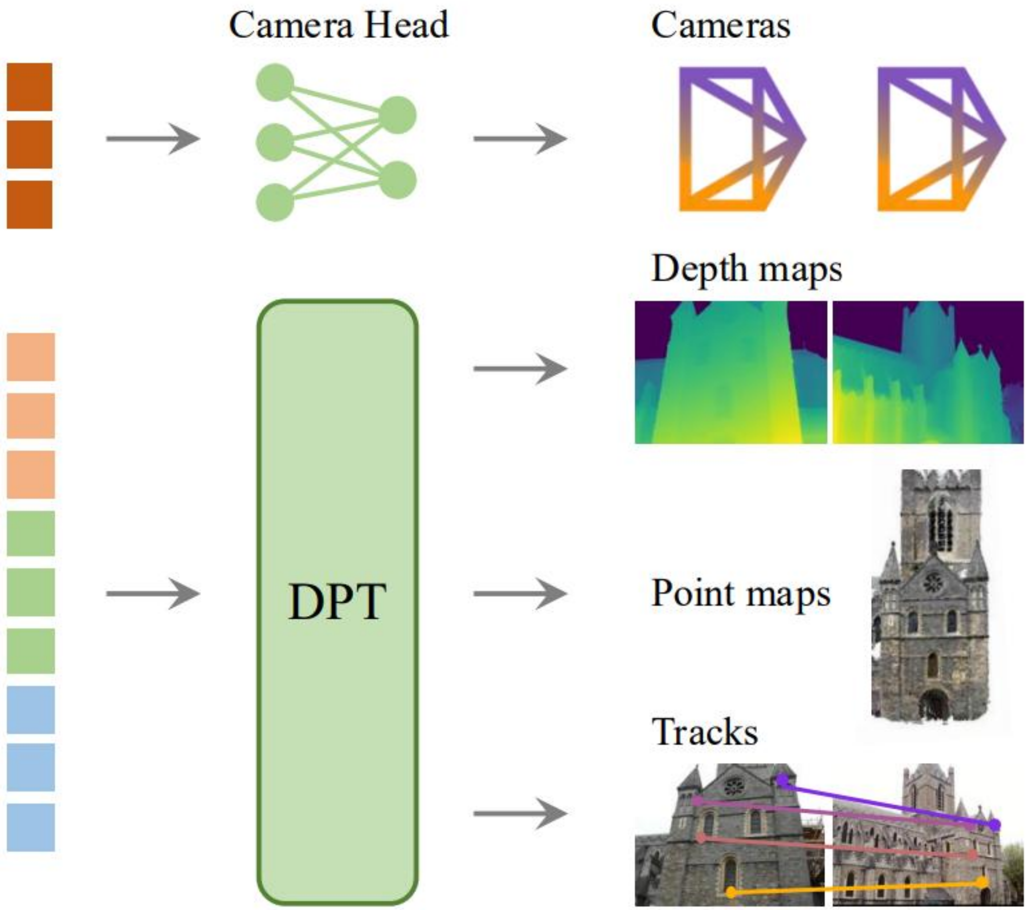
$(t_1^g := \overline{t^g},\ t_1^R := \overline{t^R})$	$(t_1^g := \overline{\overline{t^g}},\ t_1^R := \overline{\overline{t^R}}),\ i \in [2, \dots, N]$
Tokens of the first frame	Tokens of all other frames

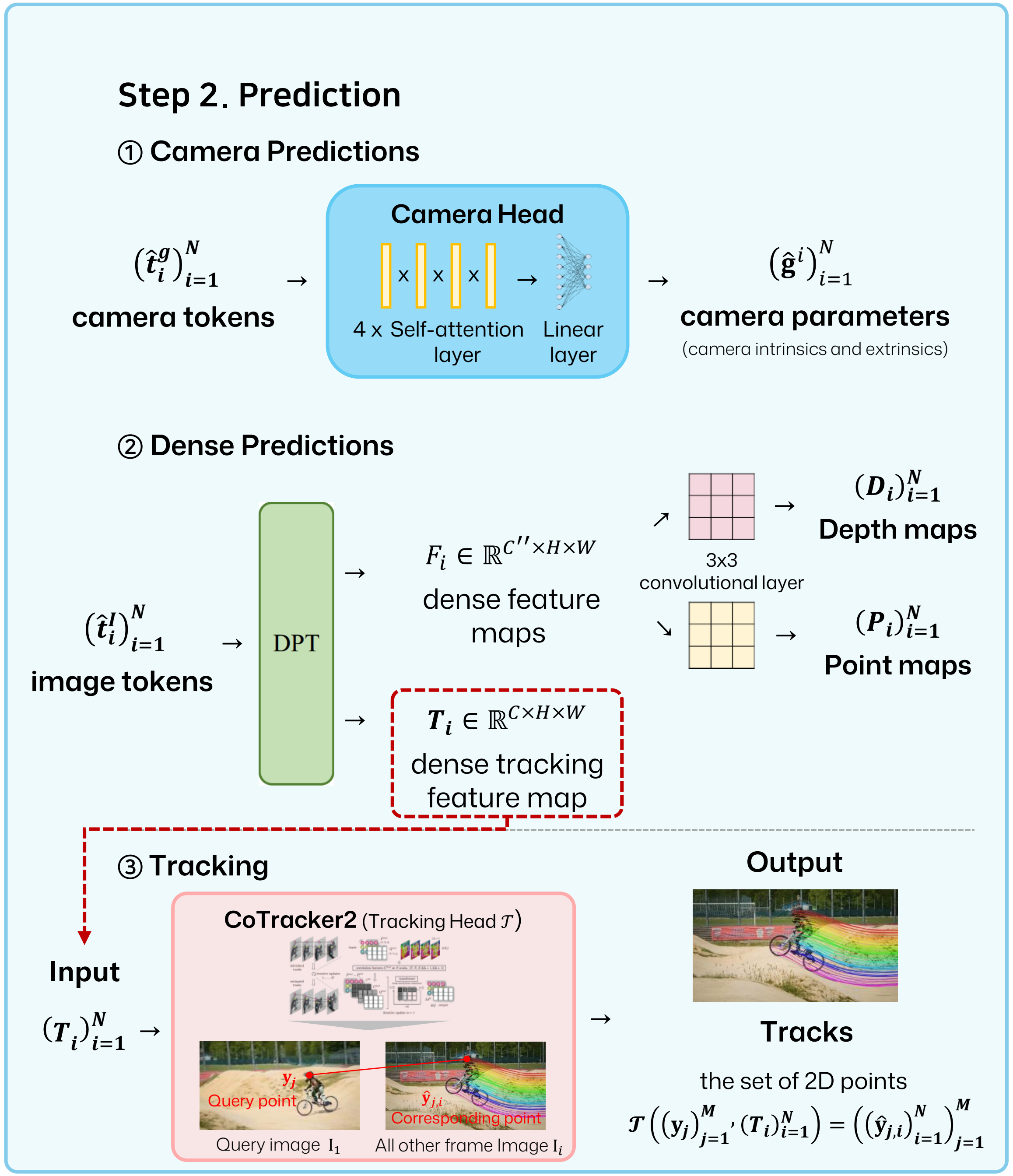
Distinguish the first frame from the rest

the refined camera and register tokens become **frame-specific**

Represent the 3D predictions
<u>**in the coordinate frame of the first camera**</u>

Prediction Heads





Experiments

Camera Pose Estimation

*** Camera Pose Estimation**

- **Task:** The task of predicting the external parameters (position & orientation) and internal settings of the cameras that captured the images.
- **Metric: AUC@30,** which combines **RRA (Relative Rotation Accuracy)** and **RTA (Relative Translation Accuracy)**

Methods	Re10K *(unseen)* AUC@30 ↑ (Not Trained)	CO3Dv2 AUC@30 ↑ (Not Trained)	Time
Colmap+SPSG [92]	45.2	25.3	∼ 15s
PixSfM [66]	49.4	30.1	> 20s
PoseDiff [124]	48.0	66.5	∼ 7s
DUSt3R [129]	67.7	76.7	∼ 7s
MASt3R [62]	76.4	81.8	∼ 9s
VGGSfM v2 [125]	78.9	83.4	∼ 10s
MV-DUSt3R [111] ‡	71.3	69.5	∼ 0.6s
CUT3R [127] ‡	75.3	82.8	∼ 0.6s
FLARE [156] ‡	78.8	83.3	∼ 0.5s
Fast3R [141] ‡	72.7	82.5	**∼ 0.2s**
Ours (Feed-Forward)	85.3	88.2	**∼ 0.2s**
Ours (with BA)	**93.5**	**91.8**	∼ 1.8s

Bundle Adjustment

Post-processing (~ 10s)

Global Alignment

Table 1. **Camera Pose Estimation on RealEstate10K [161] and CO3Dv2 [88]** with 10 random frames. All metrics the higher the better. None of the methods were trained on the Re10K dataset. Runtime were measured using one H100 GPU. Methods marked with ‡ represent concurrent work.

VGGT achieves superior performance while only operating in a feed-forward manner, requiring just 0.2 seconds across all metrics on both datasets

*** Experiment Result Analysis**

① VGGT demonstrates **significant performance advantages**, **with speed similar to the fastest variant Fast3R**

② VGGT can be **improved even further by combining it with optimization methods** from visual geometry optimization, however, **significantly faster than existing methods (only around 2 seconds even with BA)**

VGGT **directly predicts** close-to-accurate **point/depth maps**, which can serve as a **good initialization for BA**

4. Experiments

Multi-view Depth Estimation

*** Multi-view Depth Estimation**

- **Task:** The task of predicting a map that represents the distance from the camera to every pixel in the scene.
- **Metric: standard DTU metrics** → Accuracy : the smallest Euclidean distance from the prediction to ground truth
  Completeness : the smallest Euclidean distance from the ground truth to prediction
  Overall : Chamfer distance

Known GT camera	Method	Acc.↓	Comp.↓	Overall↓
✓	Gipuma [40]	**0.283**	0.873	0.578
✓	MVSNet [144]	0.396	0.527	0.462
✓	CIDER [139]	0.417	0.437	0.427
✓	PatchmatchNet [121]	0.427	0.377	0.417
✓	MASt3R [62]	0.403	0.344	0.374
✓	GeoMVSNet [157]	0.331	**0.259**	**0.295**
✗	DUSt3R [129]	2.677	0.805	1.741
✗	Ours	**0.389**	**0.374**	**0.382**

*** Experiment Result Analysis**

Others : The methods that **know ground-truth cameras** at test time

≈

Ours : The method that **do not know ground-truth cameras** at test time

the benefits of VGGT's multi-image training scheme
that teaches it to reason about multi-view triangulation natively

⇓

VGGT results comparable to methods that know ground-truth cameras at test time, naturally outperforming DUSt3R significantly

Point Map Estimation

* Point Map Estimation

- **Task:** The task of predicting a dense set of 3D points in space that define the geometric structure of the captured scene.
- **Metric:** Accuracy : the smallest Euclidean distance from the prediction to ground truth
  Completeness : the smallest Euclidean distance from the ground truth to prediction
  Overall : Chamfer distance
- **Experiment process:** ① Randomly sample 10 frames for each frame.
  ② Align the predicted point cloud to the ground truth using the Umeyama algorithm.
  ③ Filter out invalid points using the official masks.

Methods	Acc.↓	Comp.↓	Overall↓	Time
DUSt3R	1.167	0.842	1.005	$\sim$ 7s
MASt3R	0.968	0.684	0.826	$\sim$ 9s
Ours (Point)	0.901	0.518	0.709	$\sim$ 0.2s
Ours (Depth + Cam)	**0.873**	**0.482**	**0.677**	$\sim$ 0.2s

Table 3. **Point Map Estimation on ETH3D [97].** DUSt3R and MASt3R use global alignment while ours is feed-forward and, hence, much faster. The row *Ours (Point)* indicates the results using the point map head directly, while *Ours (Depth + Cam)* denotes constructing point clouds from the depth map head combined with the camera head.

* Experiment Result Analysis

Ours (Point) : The **direct predictions** from the point map head

^

Ours (Depth + Cam) : The **indirect predictions** from the **depth and camera heads**

the benefits of decomposing a complex task into simpler subproblems

(point map estimation = depth map and camera prediction)

⇓

VGGT outperforms significantly in a single feed-forward regime at only 0.2 seconds per reconstruction

Point Map Estimation

Figure 4. **Additional Visualizations of Point Map Estimation.** Camera frustums illustrate the estimated camera poses. Explore our interactive demo for better visualization quality.

Point Map Estimation

Out-of-domain Examples

① Oil paintings

② Non-overlapping frames

③ Scenes with repeating or homogeneous textures

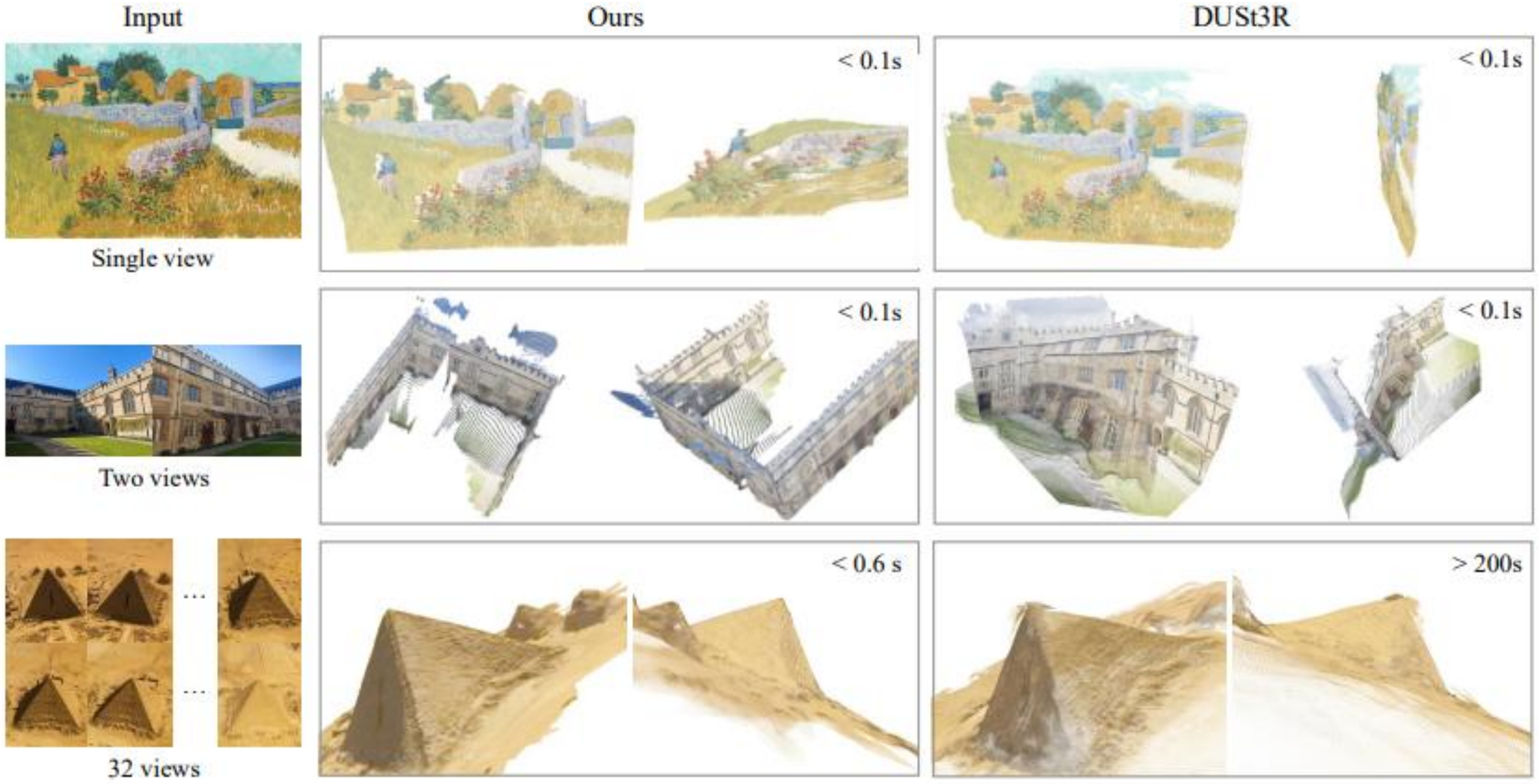


Figure 3. **Qualitative comparison of our predicted 3D points to DUSt3R on in-the-wild images.** As shown in the top row, our method successfully predicts the geometric structure of an oil painting, while DUSt3R predicts a slightly distorted plane. In the second row, our method correctly recovers a 3D scene from two images with no overlap, while DUSt3R fails. The third row provides a challenging example with repeated textures, while our prediction is still high-quality. We do not include examples with more than 32 frames, as DUSt3R runs out of memory beyond this limit.

VGGT outputs high-quality predictions and generalizes well, excelling on challenging out-of-domain examples

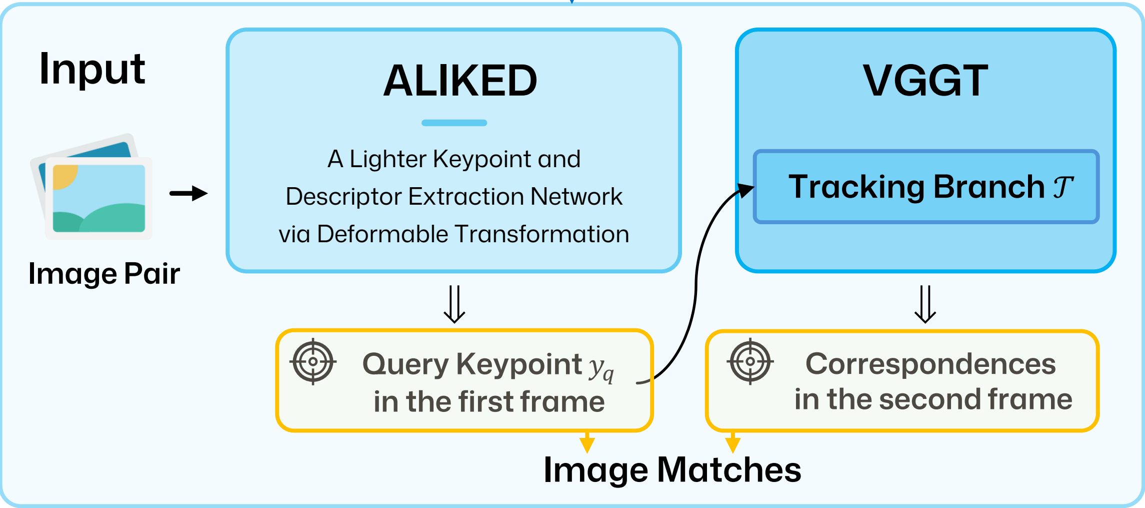
Image Matching

* Image Matching

- **Task:** The task of finding corresponding 2D pixel pairs between two images that map to the same 3D location in the real world.
- **Metric: AUC**
- **Experiment process:** ① Extract the matches for each image pair.
  ② Estimate an essential matrix using the matches.
  ③ Decompose to a relative camera pose..

Method	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
SuperGlue [92]	16.2	33.8	51.8
LoFTR [105]	22.1	40.8	57.6
DKM [32]	29.4	50.7	68.3
CasMTR [9]	27.1	47.0	64.4
Roma [33]	31.8	53.4	70.9
Ours	**33.9**	**55.2**	**73.4**

Table 4. **Two-View matching comparison on ScanNet-1500 [18, 92].** Although our tracking head is not specialized for the two-view setting, it outperforms the state-of-the-art two-view matching method Roma. Measured in AUC (higher is better).



VGGT achieves the highest accuracy among all baselines

(despite not being explicitly trained for two-view matching)

Ablation Studies

*** Ablation Studies Experiments Setup**

- **(Parameter)** an identical number of parameters, using a total of 2L attention layers.
- **(Hyperparameter)** an identical number of hyperparameters such as the hidden dimension and the number of heads
- **(Evaluation Metric) Point map estimation** reflecting the model's joint understanding of scene geometry and camera parameters

① Ablation Study for Feature Backbone

ETH3D Dataset	Acc.↓	Comp.↓	Overall↓
Cross-Attention	1.287	0.835	1.061
Global Self-Attention Only	1.032	0.621	0.827
Alternating-Attention	**0.901**	**0.518**	**0.709**

Table 5. **Ablation Study for Transformer Backbone** on ETH3D. We compare our alternating-attention architecture against two variants: one using only global self-attention and another employing cross-attention.

Alternating-Attention architecture
outperforms both baseline variants

② Ablation Study for Multi-task Learning

w. $\mathcal{L}_{camera}$	w. $\mathcal{L}_{depth}$	w. $\mathcal{L}_{track}$	Acc.↓	Comp.↓	Overall↓
✗	✓	✓	1.042	0.627	0.834
✓	✗	✓	0.920	0.534	0.727
✓	✓	✗	0.976	0.603	0.790
✓	✓	✓	**0.901**	**0.518**	**0.709**

Table 6. **Ablation Study for Multi-task Learning**, which shows that simultaneous training with camera, depth and track estimation yields the highest accuracy in point map estimation on ETH3D.

Simultaneously learning multiple 3D quantities
enhances the point map estimation performance

incorporating camera parameter estimation clearly enhances point map accuracy

Finetuning for Downstream Tasks

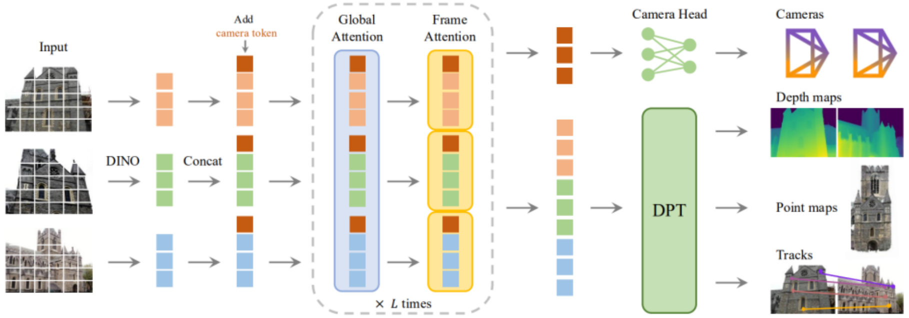
* Task 1. Feed-forward Novel View Synthesis

- **Task:** The task of instantly synthesizing and rendering how a scene would appear from a new, previously unseen camera viewpoint, based on the input images.
- **Metric: PSNR, SSIM, LPIPS**
- **Experiment process:** ① Convert the 4 input view images into tokens by DINO
  ② For the target views, encode their Plücker ray images into tokens using a convolutional layer
  ③ Concatenate tokens, representing both the input images and the target views
  ④ Process them with the AA transformer
  ⑤ Subsequently, regress the RGB colors for the target views with a DPT head

(* **Notes!**
Do not input the Plücker rays for the source images,
thus, the model is **not given the camera parameters for the input frames**.)

Method	Known Input Cam	Size	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [110]	✓	256	21.44	0.832	0.122
GS-LRM [154]	✓	256	29.59	0.944	0.051
LVSM [53]	✓	256	31.71	0.957	0.027
Ours-NVS*	✗	224	30.41	0.949	0.033

Table 7. **Quantitative comparisons for view synthesis on GSO [28] dataset.** Finetuning VGGT for feed-forward novel view synthesis, it demonstrates competitive performance even without knowing camera extrinsic and intrinsic parameters for the input images. Note that * indicates using a small training set (only 20%).



⇓

VGGT achieves competitive results on the GSO dataset,

despite not requiring **the input camera parameters**
and using **less training data than LVSM**

Finetuning for Downstream Tasks

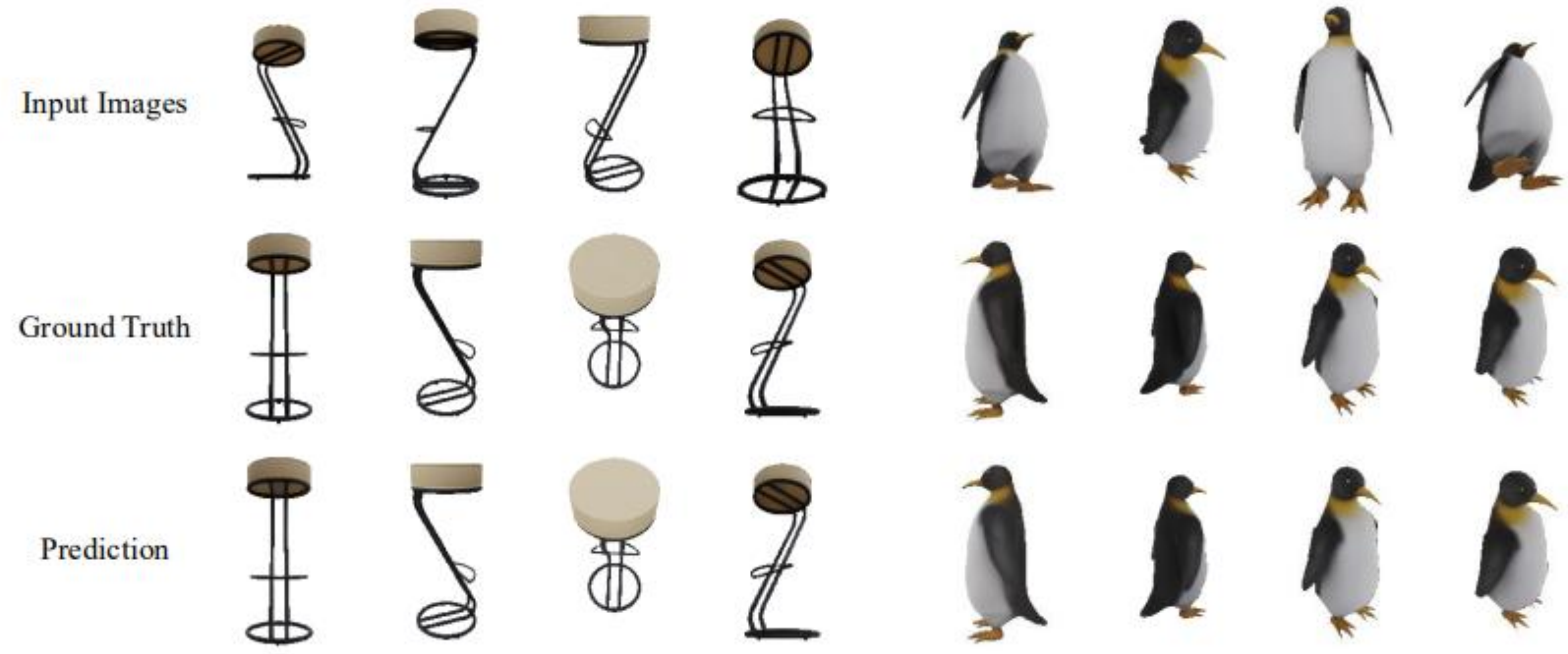


Figure 6. **Qualitative Examples of Novel View Synthesis.** The top row shows the input images, the middle row displays the ground truth images from target viewpoints, and the bottom row presents our synthesized images.

Finetuning for Downstream Tasks

* Task 2. Dynamic Point Tracking

- **Task:** The task of predicting the 2D trajectories of arbitrary points of interest across dynamic video sequences, including complex motions and occlusions.
- **Metric: Occlusion Accuracy (OA;** comprising the binary accuracy of occlusion predictions δ_{vis}^{avg} comprising the mean proportion of visible points accurately tracked within a certain pixel threshold)
  Average Jaccard (AJ; measuring tracking and occlusion prediction accuracy together**)**
- **Experiment process:** ① Adapt CoTracker2 (the SOTA model) by substituting its backbone with our pretrained feature backbone.
  (* **Note!**
  It is essential, as **VGGT is trained on unordered image collections** instead of sequential videos)
  ② Predict the tracking features $\mathcal{T}^i$ with VGGT's backbone
  ③ Enter them into the rest of the CoTracker2 architecture, finally predicting the tracks
  ④ Finetune the entire modified tracker on Kubric

Method	Kinetics			RGB-S			DAVIS		
	AJ	δ_{avg}^{vis}	OA	AJ	δ_{avg}^{vis}	OA	AJ	δ_{avg}^{vis}	OA
TAPTR [63]	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1
LocoTrack [13]	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2
BootsTAPIR [26]	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7
CoTracker [56]	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3
CoTracker + Ours	**57.2**	**69.0**	**88.9**	**72.1**	**84.0**	**91.6**	**64.7**	**77.5**	**91.4**

Table 8. **Dynamic Point Tracking Results on the TAP-Vid benchmarks.** Although our model was not designed for dynamic scenes, simply fine-tuning CoTracker with our pretrained weights significantly enhances performance, demonstrating the robustness and effectiveness of our learned features.

* Experiment Result Analysis

the integration of pretrained VGGT **significantly enhances** CoTracker's performance on the TAP-Vid benchmark

⇓

VGGT achieves strong performance demonstrating the generalization capability of its features

despite the TAP-Vid benchmark's inclusion of videos featuring rapid dynamic motions from various data sources, even in scenarios for which it was not explicitly designed.

Finetuning for Downstream Tasks

Figure 5. **Visualization of Rigid and Dynamic Point Tracking.** Top: VGGT's tracking module $\mathcal{T}$ outputs keypoint tracks for an unordered set of input images depicting a static scene. Bottom: We finetune the backbone of VGGT to enhance a dynamic point tracker CoTracker [56], which processes sequential inputs.

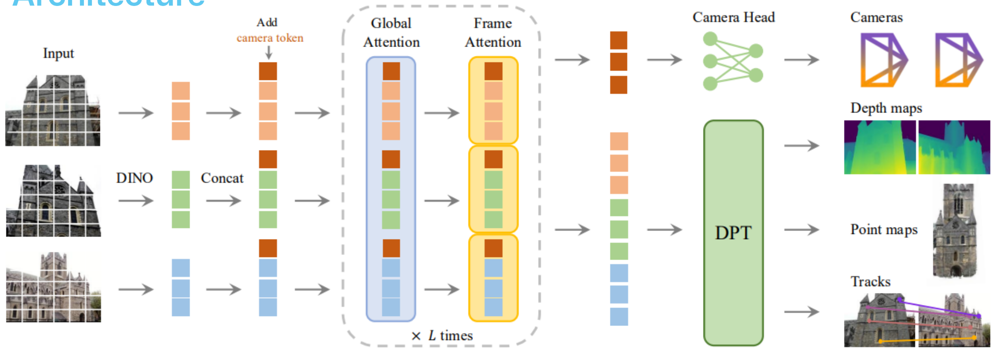
Conclusion

1 Page Summary

VGGT: Visual Geometry Grounded Transformer

"Alternating-Attention 구조 Transformer 기반의 single feed-forward 방식의 효율적인 핵심 3D 속성 추정
Camera parameter, Depth map, Point map, Point Tracking 및 하위 응용 작업에서 SOTA 달성"

Architecture



⇓

Experimental Results

Camera Estimation

Methods	Not Trained Re10K (unseen) AUC@30 ↑	Not Trained CO3Dv2 AUC@30 ↑	Time
Colmap+SPSG [92]	45.2	25.3	~ 15s
PixSfM [66]	49.4	30.1	> 20s
PoseDiff [124]	48.0	66.5	~ 7s
DUSt3R [129]	67.7	76.7	~ 7s
MASt3R [62]	76.4	81.8	~ 9s
VGGSfM v2 [125]	78.9	83.4	~ 10s
MV-DUSt3R [111] ‡	71.3	69.5	~ 0.6s
CUT3R [127] ‡	75.3	82.8	~ 0.6s
FLARE [156] ‡	78.8	83.3	~ 0.5s
Fast3R [141] ‡	72.7	82.5	~ 0.2s
Ours (Feed-Forward)	85.3	88.2	~ 0.2s
Ours (with BA)	**93.5**	**91.8**	~ 1.8s

Depth Estimation

Known GT camera	Method	Acc.↓	Comp.↓	Overall↓
✓	Gipuma [40]	**0.283**	0.873	0.578
✓	MVSNet [144]	0.396	0.527	0.462
✓	CIDER [139]	0.417	0.437	0.427
✓	PatchmatchNet [121]	0.427	0.377	0.417
✓	MASt3R [62]	0.403	0.344	0.374
✓	GeoMVSNet [157]	0.331	**0.259**	**0.295**
✗	DUSt3R [129]	2.677	0.805	1.741
✗	Ours	0.389	0.374	0.382

Point Map Estimation

Methods	Acc.↓	Comp.↓	Overall↓	Time
DUSt3R	1.167	0.842	1.005	~ 7s
MASt3R	0.968	0.684	0.826	~ 9s
Ours (Point)	0.901	0.518	0.709	~ 0.2s
Ours (Depth + Cam)	**0.873**	**0.482**	**0.677**	~ 0.2s

Novel View Synthesis

Method	Known Input Cam	Size	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [110]	✓	256	21.44	0.832	0.122
GS-LRM [154]	✓	256	29.59	0.944	0.051
LVSM [53]	✓	256	31.71	0.957	0.027
Ours-NVS*	✗	224	30.41	0.949	0.033

Dynamic Point Tracking

Method	Kinetics			RGB-S			DAVIS		
	AJ	δ^{vis}_{avg}	OA	AJ	δ^{vis}_{avg}	OA	AJ	δ^{vis}_{avg}	OA
TAPTR [63]	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1
LocoTrack [13]	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2
BootsTAPIR [26]	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7
CoTracker [56]	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3
CoTracker + Ours	**57.2**	**69.0**	**88.9**	**72.1**	**84.0**	**91.6**	**64.7**	**77.5**	**91.4**

Image Matching

Method	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
SuperGlue [92]	16.2	33.8	51.8
LoFTR [105]	22.1	40.8	57.6
DKM [32]	29.4	50.7	68.3
CasMTR [9]	27.1	47.0	64.4
Roma [33]	31.8	53.4	70.9
Ours	**33.9**	**55.2**	**73.4**

Achieves performance at least comparable to, or even surpassing, existing SOTA models.

Problem Definition

- **배경**: 전통적인 Visual Geometry 기법의 높은 계산 복잡도 및
  DL 기반 기법의 pairwise 입력 한계와 후처리 기법 사용 문제
- **핵심 문제**: 반복적 최적화 및 후처리 과정으로 인한 **느린 실행 시간과 높은 비용**

Methodology : AA(Alternating-Attention)-based Large Transformer Model

- **전체 아이디어**: 전통적 시점 기하학 (Visual Geometry) 원리를 AA 구조 대형 트랜스포머에 내재화(grounded)하여, 핵심 3D 속성을 single feed-forward로 동시 예측
- **핵심 아이디어**: 기존 attention 구조 대비 **효율적**이면서도, **프레임 간 정보 통합**이 가능한 AA 구조를 활용하여, 입력 한계 해결 및 다양한 3D 속성 예측 정확도 향상
- **작동 방식**: Global Self Attention과 Framewise Self Attention **레이어의 반복 수행**으로 프레임 간 기하학적 관계 통합 및 개별 프레임 이미지의 내부 일관성 유지

Experimental Results

- **결론**: VGGT는 AA를 구조로 3D Reconstruction을 수행하며, 이는 관련 3D Multi-task 실험으로 우수한 효율성과 일반화 성능으로 SOTA를 달성했음을 입증

The End!

Thank you for listening!

SOTA AI Review Week 6. VGGT: Visual Geometry Transformer Review