DepthAnything3

Recovering the Visual Space from Any Views

AUTHORS

Haotong Lin*, Sili Chen*, Jun Hao Liew*, Donny Y. Chen*, Zhenyu Li, Guang Shi, Jiashi Feng, Bingyi Kang†

ByteDance Seed

Core Contributions

- **Minimal Modeling:** Single ViT + per-pixel depth-ray targets.
- **SOTA Performance:** +35.7% pose AUC, +23.6% geometric accuracy vs VGGT.
- **Universal:** Works with monocular, multi-view, or video frame sequences.

Figure 1: DA3 reconstructs visual space from any number of images

Dual-DPT
NOVEL HEAD

DINOv2
BACKBONE

2 Introduction & Motivation

The Core Problem

Existing foundation models require:

- ❌ **VGGT:** Redundant multi-task targets (point maps + depth + pose) + multi-stage architecture
- ❌ **DUSt3R:** Point maps insufficient for metric consistency
- ❌ **Traditional SfM/MVS:** Brittle under textureless regions, specularities

Central Question:
Can a **SINGLE PLAIN ViT** with **MINIMAL TARGETS** (depth + rays only) suffice for unified 3D reconstruction?

YES → DepthAnything3

Key Achievement: DA3 reaches SOTA performance using **only public academic datasets**, without proprietary data.

Limitations of Prior Work

Traditional Pipelines (SfM / MVS / SLAM)
Modular systems are brittle under textureless regions, specularities, or large baselines.

Early Deep Learning Approaches
Fixed input cardinality, complex architectures, and limited scalability.

Current Foundation Models (e.g., VGGT)
Rely on multi-head task bundles, redundant targets, and heavy design overhead.

DA3 Solution Highlights

Unified Representation
Depth + Ray formulation handles arbitrary inputs (single image, stereo, video) in one model.

Minimal Architecture
Single plain ViT backbone—no complex multi-stage pipelines or redundant heads.

SOTA Performance
+35.7% pose accuracy improvement, trained only on public datasets.

3 Formulation I: Depth-Ray Representation

Problem Setup

Arbitrary Visual Inputs

Input set could be given as a single image, stereo pair, or video frame sequence.

Per-Pixel Ray Map M

Instead of global pose, the model predicts a dense map M containing ray origin t and direction d for every pixel.

Why Avoid Explicit Rotation?

Directly regressing a rotation matrix R imposes an **orthogonality constraint** ($R^T R = I$) that is difficult for neural networks to satisfy, often leading to optimization instability or degenerate poses.

- Ray directions $d \in \mathbb{R}^3$ avoid explicit orthogonality constraints, simplifying optimization.

Key: Ray direction is derived from classical formulation as $d = RK^{-1}p$, linking rotation matrix and intrinsics to rays.

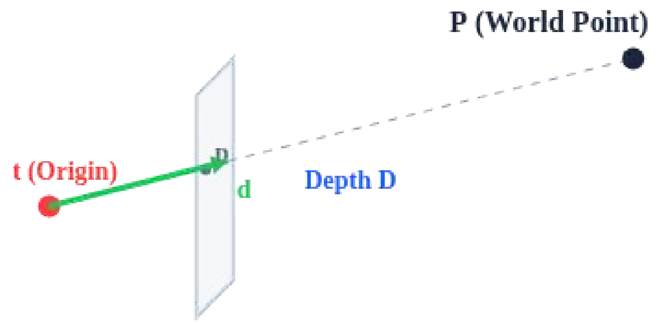
Why is d unnormalized?

d is unnormalized because $||d||$ encodes the projection transformation from K^{-1}. Normalizing would decouple the scale relationship between depth and projection. The unnormalized d preserves the pixel-to-world scale factor, which is critical for geometric consistency: $P = t + D \cdot d$ requires d's magnitude to properly fuse depth with ray direction.

```
P = t + D(u, v) d
```

```
P (3D Point) = t (Ray Origin) + D (Depth) × d (Direction)
```

GEOMETRIC DERIVATION



The direction d derived via backprojection: d = R K^{-1} p

4 Formulation II: Camera From Ray Map

Deriving Parameters from Ray Map M

Given ray map $M \in \mathbb{R}^{H \times W \times 6}$ with origins $M_{:3}$ and directions $M_{3:}$:

1 **Estimate Camera Center t_c**

$$t_c = \frac{1}{H \cdot W} \sum_{h,w} M(h, w, :3)$$

2 **Recover K, R via Homography**

Canonical ray $d_I = p$ relates to camera ray d_{cam} via $H = KR$.

$$H^* = \arg \min_{||H||=1} \sum_{h,w} ||(Hp_{h,w}) \times M(h, w, 3:)||$$

Why cross product?

Minimizes angular error—enforces directional alignment between $H \cdot p$ and predicted rays.

Solved via DLT, then decompose H^* using RQ decomposition → (K, R).

Lightweight Camera Head D_C

Challenge: Pose-from-rays optimization is computationally expensive at inference.

Solution: A dedicated camera head operating on camera tokens directly predicts (f, q, t) parameters with **negligible overhead**—bypassing expensive DLT/RQ at test time.

Camera Conditioning Tokens

Camera information is injected via tokens prepended to each view, enabling both posed and unposed inputs.

```
If pose known:
```

$$c_i = E_c(f_i, q_i, t_i)$$

Encoded via MLP E_c from FOV, quaternion, translation.

```
If pose unknown:
```

Use a shared learnable token c_ℓ.

TOKEN INTEGRATION FLOW

1 Image Patches I_i ⊕ Camera Token c_i

↓

2 ViT Backbone
Joint Processing (All Layers)

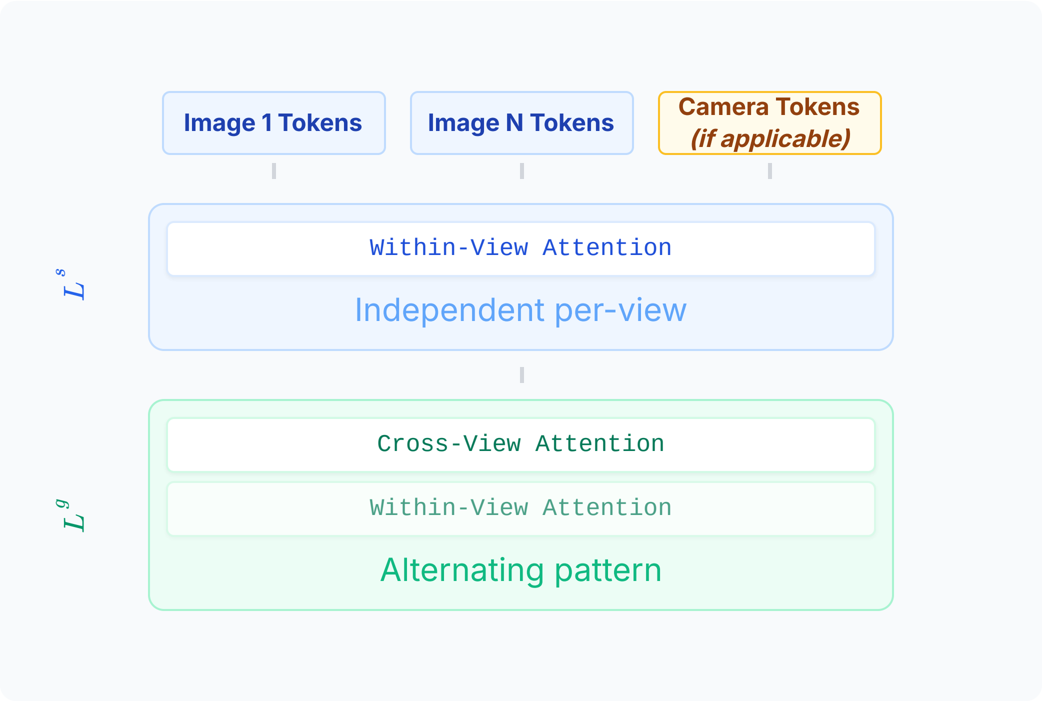
↓

3 Geometry Features | Updated c_i

5 Architecture I: Single ViT & Input-Adaptive Attention

Transformer Block Structure

Pretrained DINOv2



KEY INNOVATION: Input-Adaptive Cross-View Attention

Standard ViT handles multiple views without modification

Phase 1: Within-View (L_s layers)

- Tokens attend only within their own view
- Extract per-view features (monocular depth)
- Build local context first

Phase 2: Cross-View (L_g layers)

Alternating:

- **Cross-view:** Reorder → interleave → correspondences
- **Within-view:** Group back → refine

Standard ViT: only token ordering changes

Input-Adaptive Property

- $N_v = 1$: Monocular depth (zero overhead)
- $N_v =$ 2-18: Scales gracefully
- **Complexity:** $O(N^2)$ ViT (no 3D volumes)

Empirical optimal ratio: $L_s : L_g = 2 : 1$

Why 2:1 ratio?

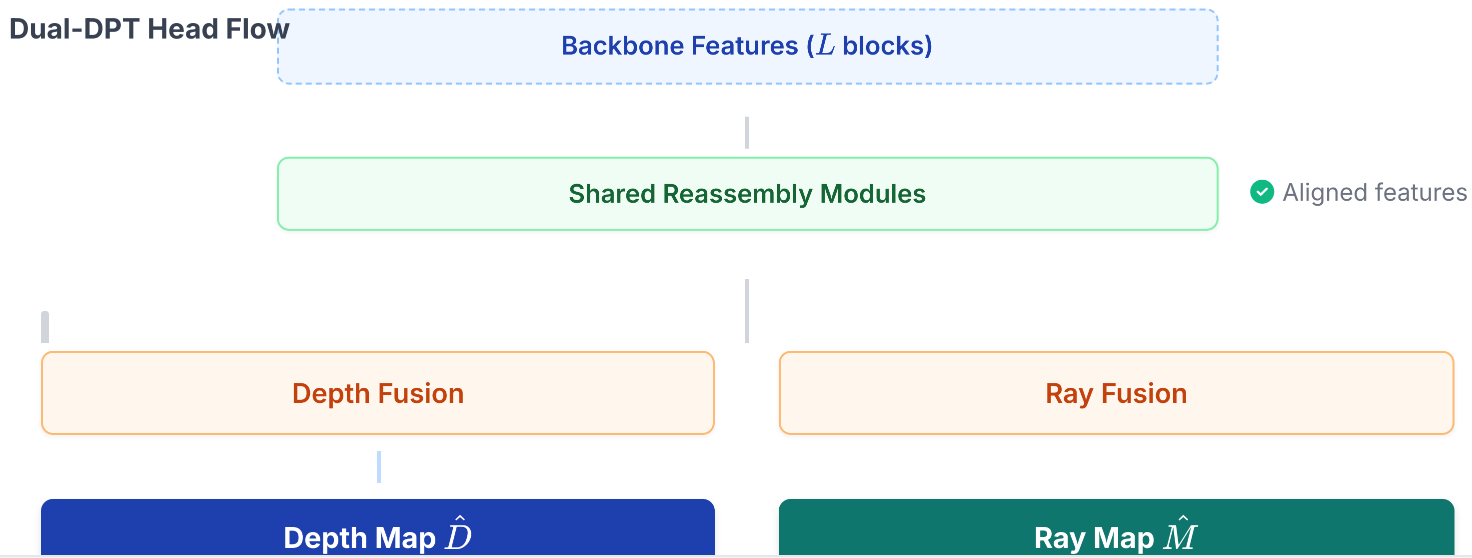
Balances local feature learning with cross-view reasoning.

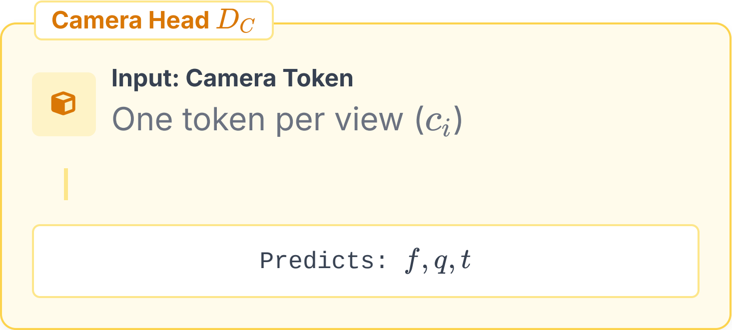
Trade-offs:

- Too few L_g → limited multi-view fusion
- Too many L_g → reduces within-view discriminability
- L_g scales as $O(N_v \cdot H \cdot W)^2$ vs $O(H \cdot W)^2$ for L_s

Tab. 7: 2:1 provides optimal accuracy-efficiency balance

Architecture II: Dual-DPT Head & Camera Head





Multi-Task Efficiency

Dual-DPT outperforms separate heads in pose and geometry (Tables 6-7).

Dual-DPT Head Design

- **Shared Reassembly:** Upsamples and concatenates multi-scale ViT features (from different transformer layers) into dense spatial representations before task-specific fusion (from DPT architecture).
- **Branch-Specific Fusion:** Two distinct paths fuse features for depth vs. ray tasks.
- **Benefit:** Encourages strong task interaction while minimizing redundant computation—outperforms separate heads with minimal parameter overhead.

Camera Head (D_C)

- Operates exclusively on **camera tokens** (one per view).
- Predicts explicit parameters: FOV $f \in \mathbb{R}^2$, quaternion $q \in \mathbb{R}^4$, translation $t \in \mathbb{R}^3$.
- **Efficiency:** Negligible computational cost—amortizes pose extraction without expensive dense ray map processing at inference.

7 Training Methodology

Teacher-Student Paradigm

Teacher: DINOv2+DPT
- **Monocular only** (single images)
- Standard DPT decoder
- No multi-view / No camera tokens
- Trained on synthetic data only

Student: DA3
- **Multi-view** (2-18 images)
- Dual-DPT + cross-view attention
- Camera tokens + camera head
- Trained on mixed real+synthetic

Synthetic → Teacher → Relative Depth

RANSAC align (s,t) → cached pseudo-labels

↓

Real+Noisy → Pseudo-Labels → Student (DA3)

Mixed Data Strategy

Real Depth | 3D Recon | Synthetic

~40% synthetic, ~30% real (ScanNet++), ~30% reconstructed (DL3DV)

KEY DETAILS

200k steps on 128 H100 GPUs

- Base res 504^2 (varied AR)
- Pose cond prob p=0.2
- Wall-clock: ~5-7 days
- Camera head D_C: ~1ms

Student Training Objective

$$L = L_D + L_M + L_P + \beta L_C + \alpha L_{grad}$$

(where $\alpha = 1, \beta = 1$)

Student: Multi-view depth + rays + pose (mixed real+synthetic data)

Teacher: $L_T = L_{grad} + L_{gl} + 0.5 L_N + L_{sky} + L_{obj}$ (monocular, synthetic only)

Student Loss Components

1. Confidence-Aware Depth Loss (L_D)

Weighted L1 loss with learned uncertainty $D_{c,p}$ that down-weights ambiguous regions.

2. Ray Map Loss (L_M)

L1 loss on ray vectors $M = (t, d)$ for direction consistency.

3. Point Consistency (L_P)

Loss on 3D points $P = t + \hat{D} \odot d$ to enforce geometric validity.

4. Camera Head (L_C)

Supervision for pose predictions (f, q, t) from camera head.

5. Gradient Loss (L_{grad})

Edge-aware smoothness (shared with teacher).

Teacher Losses (monocular)

ROE, surface normals, sky/object masks

Teacher-Student Learning Paradigm

Why Synthetic Teacher?

- Real-world depth (LiDAR/structured light) is often **sparse, noisy, and incomplete** (see Fig. 4).
- Synthetic data provides **dense, clean geometry** with perfect ground truth.

HyperSim | Virtual KITTI | TartanAir | ScanNet++ (Noisy Target)

The Teacher Model (DA3-Teacher)

- **Architecture:** Monocular DINOv2 + DPT decoder (same backbone class).
- **Target:** Scale-shift-invariant *exponential* depth (better for near-field).
- **Losses:** Gradient + Global-Local (ROE) + Surface Normal + Sky/Obj Masks.

Robust Alignment Strategy

The teacher's relative depth $\tilde{D}$ is aligned to noisy sparse real measurements D_p via robust RANSAC least squares.

$$(s,t) = \operatorname{argmin}_{s>0,t} \sum_{p\in\Omega} m_p(s\tilde{D}_p + t - D_p)^2$$

$$D_{aligned} = \hat{s}\tilde{D} + \hat{t}$$

RANSAC Benefit

Filters out gross outliers in real sensor data using Median Absolute Deviation (MAD) thresholding, preventing teacher degradation.

Noisy Input → Clean Target

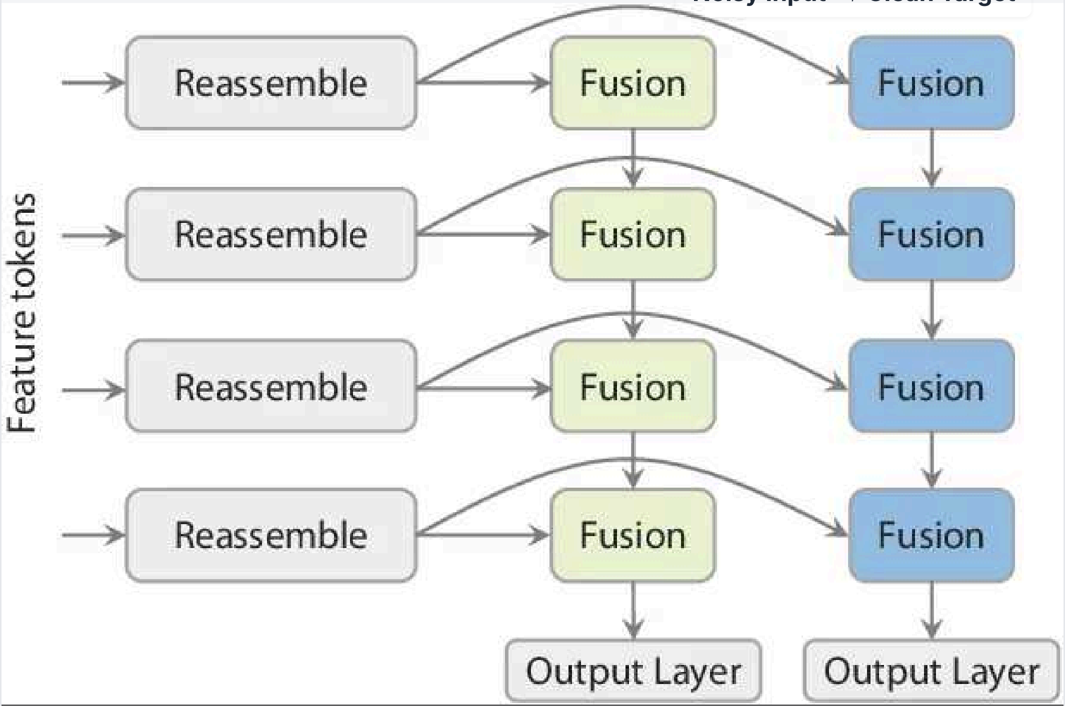


Fig 4: Data Quality & Alignment Sparse Real vs. Dense Pseudo-Label

Experiments I: Pose Accuracy & Geometry

Comparison vs VGGT

Visual Geometry Benchmark (Pose AUC)

+35.7% Improvement

Baseline context: VGGT: 46.8% avg → DA3-Large: **67.3% avg** | Relative improvement: **+44%** | Coverage: **SOTA on 18/20 settings** | (Perfect alignment = 100% AUC) | Datasets: HiRoom (synthetic), ETH3D, DTU, 7Scenes, ScanNet++ (all real-world LiDAR)

Method	Setting	HiRoom	ScanNet++	7Scenes	Avg
DUSt3R	Pose-Free	38.4	45.2	32.1	38.6
VGGT	Pose-Free	42.1	62.6	35.8	46.8
DA3-Large ⭐	**Pose-Free**	**64.5**	**85.0**	**52.3**	**67.3**

SOTA on 18/20 Settings

** DA3 sets new state-of-the-art across diverse benchmarks*

+35.7%

CAMERA POSE ACCURACY

+23.6%

GEOMETRIC ACCURACY

Evaluation Pipeline

Feed-Forward: Predict Pose + Depth

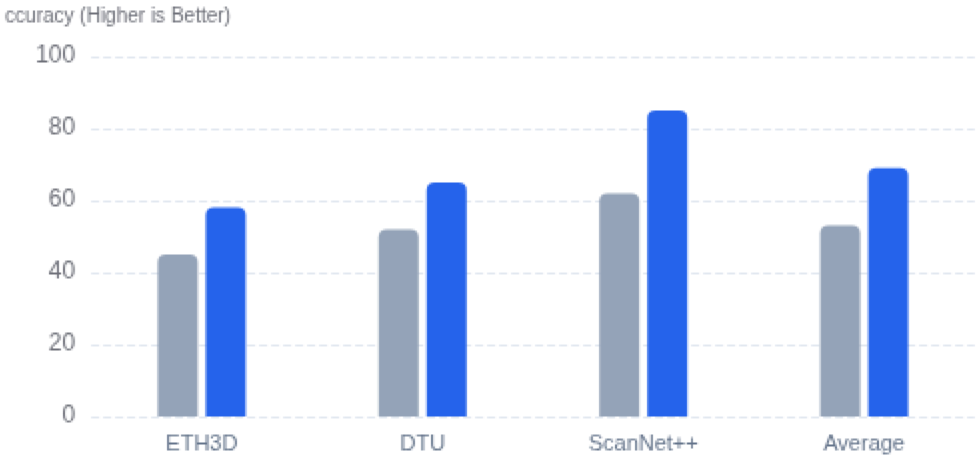
↓

Alignment: Robust RANSAC (inlier subset selection) + evo

↓

Fusion: TSDF Fusion → Point Cloud

Geometric Accuracy Gap



Experiments II: Monocular Depth & Feed-Forward 3DGS

Monocular Depth Estimation

vs Depth Anything 2

Method	KITTI (AbsRel ↓)	Sintel (AbsRel ↓)	NYUv2 (δ1 ↑)
MiDaS v3.1	0.076	0.245	0.892
Depth Anything 2	0.058	0.198	0.965
DA3-Mono (Ours)	**0.054**	**0.185**	**0.971**

Monocular Depth: +7% on KITTI vs DA2

Improvements **LIKELY** stem from (not individually ablated): (1) **depth (not disparity) target** → better for downstream 3D tasks, (2) **expanded synthetic teacher data** → broader geometry coverage, (3) **exponential encoding** → enhanced near-field discrimination.

Feed-Forward 3DGS (GS-DPT Head)

Fine-tuning Strategy: `Frozen DA3 Backbone`

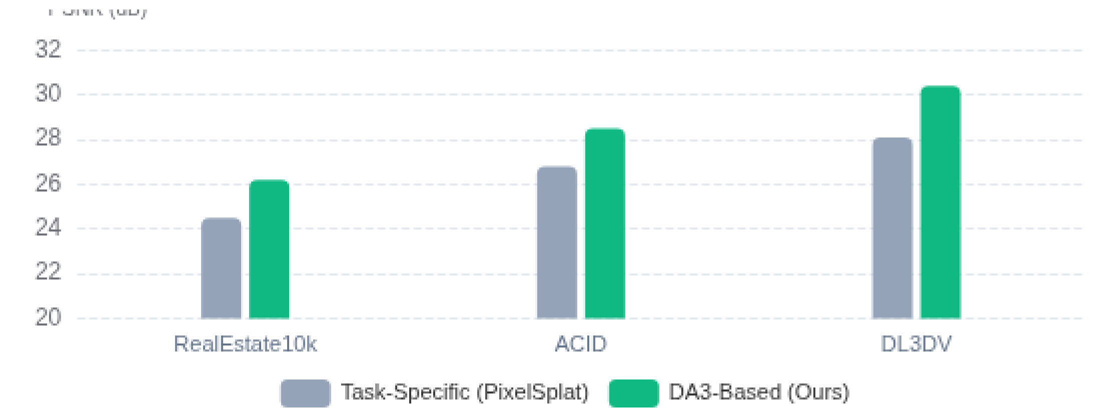
Input: Images + (Optional) Poses

↓

GS-DPT Prediction: Per-pixel Gaussians (σ, q, s, c)

Novel View Synthesis Quality (PSNR)

Pose-Adaptive



Core Findings

Geometry

FOUNDATION > TASK-SPECIFIC

Generalist backbone outperforms specialized NVS models

Adaptivity

WORKS W/ OR W/O POSE

Single model handles both settings seamlessly

Strengths & Limitations

Core Strengths

Minimal Modeling Strategy SCALABLE

Single plain ViT + depth-ray targets avoid complex bespoke architectures. Inherits scaling laws directly from DINOv2 pretraining.

Unified "Any-View" Framework

Seamlessly handles monocular, multi-view, and video inputs. Pose-optional design bridges the gap between uncalibrated images and metric 3D.

SOTA Geometry & Pose

Outperforms VGGT by huge margins (+35.7% Pose AUC). Provides robust foundation for feed-forward novel view synthesis (3DGS).

Limitations & Challenges

Computational Cost at Inference OPTIMIZATION

Extracting pose from ray maps via optimization can be slow. Pose head (DC) mitigates this but adds dependency on tokens.

Static Scene Limitation

The depth-ray formulation does not explicitly model motion or deformation fields, suggesting **POTENTIAL limitations for dynamic scenes**. *Note: The paper does not evaluate dynamic scene performance, so this remains an open question for future work.*

Memory Scaling

Trained on 2-18 views; beyond this requires memory optimization. Token budgeting strategies needed for large-scale multi-view processing.

Research Extensions

(My Proposals Inspired by DA3)

IMMEDIATE

1

Dynamic Visual Space (4D)

DA3's ray formulation **naturally extends to MOTION RAYS**: *r(t) = (t(t), d(t))*. Enables per-pixel trajectory encoding for scene flow.

Scene Flow | Video | Dynamic Nerf

NEAR TERM

2

Uncertainty & Calibration

DA3 already predicts **depth confidence D_c**. Extend to **RAY CONFIDENCE** for robust pose alignment under occlusion.

Probabilistic | Active Vision | Safety

MID TERM

3

Efficiency & Real-Time

Address DA3's **$O(N_v \cdot H \cdot W)^2$ cross-view cost** with token pruning and sparse attention patterns for L_g layers.

Edge AI | Sparsity | Latency

LONG TERM

4

Semantic & Task Coupling

Integrate language priors into **ray prediction**. Add **differentiable Bundle Adjustment** for end-to-end refinement of DA3 outputs.

Semantics | Diff. BA | Self-Supervised

Research Focus: Combine **Self-Supervised Cycle Consistency** with **Uncertainty Estimation** using DA3 backbone for robust, label-free learning.

13 Summary

Core Contributions & Rationale

Minimalism Suffices
ARCHITECTURE

A single plain ViT (DINOv2) + minimal Depth-Ray targets are sufficient for any-view geometry. No complex multi-task bundles or bespoke 3D modules needed.

Implicit Pose via Rays
FORMULATION

Predicting dense rays avoids difficult orthogonality constraints of rotation matrices (SO3). Pose emerges naturally from ray convergence.

Empirical Dominance
RESULTS

+35.7% Pose AUC vs. VGGT. Validates that scale-invariant depth + ray maps is the optimal minimal set for foundation geometry.

DepthAnything3 proves that **generalist scaling** beats **specialized engineering** for 3D visual geometry.