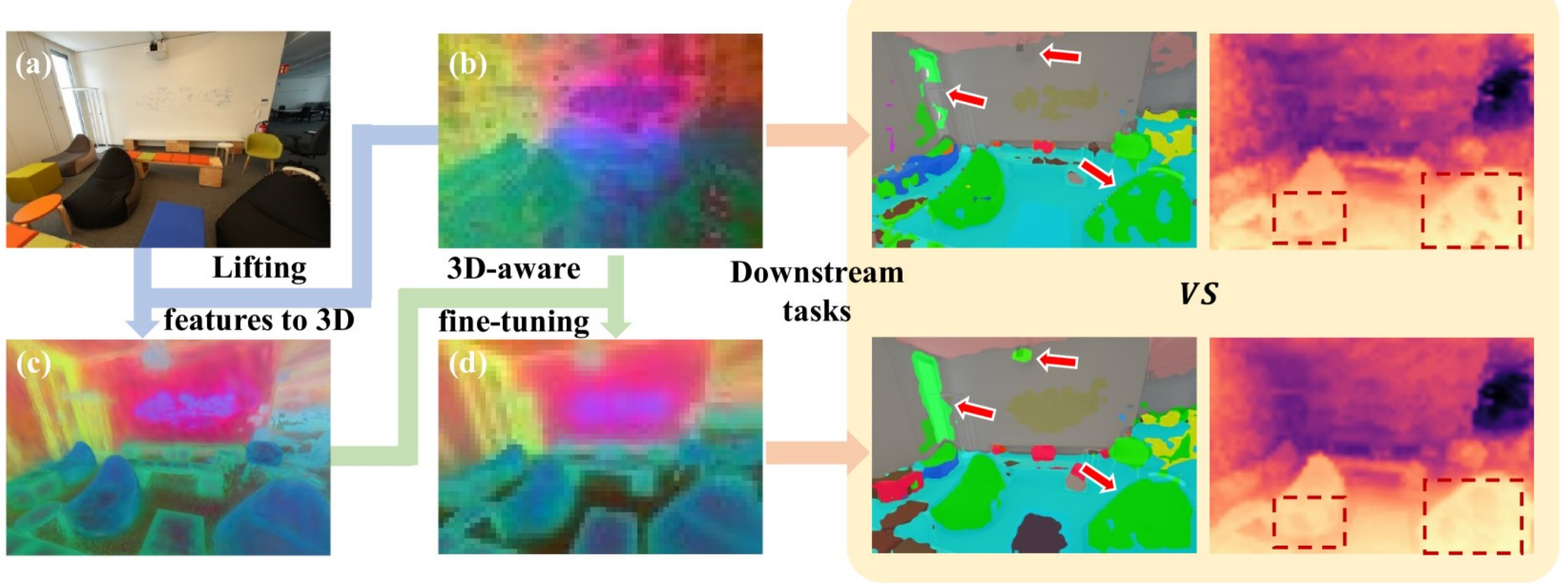


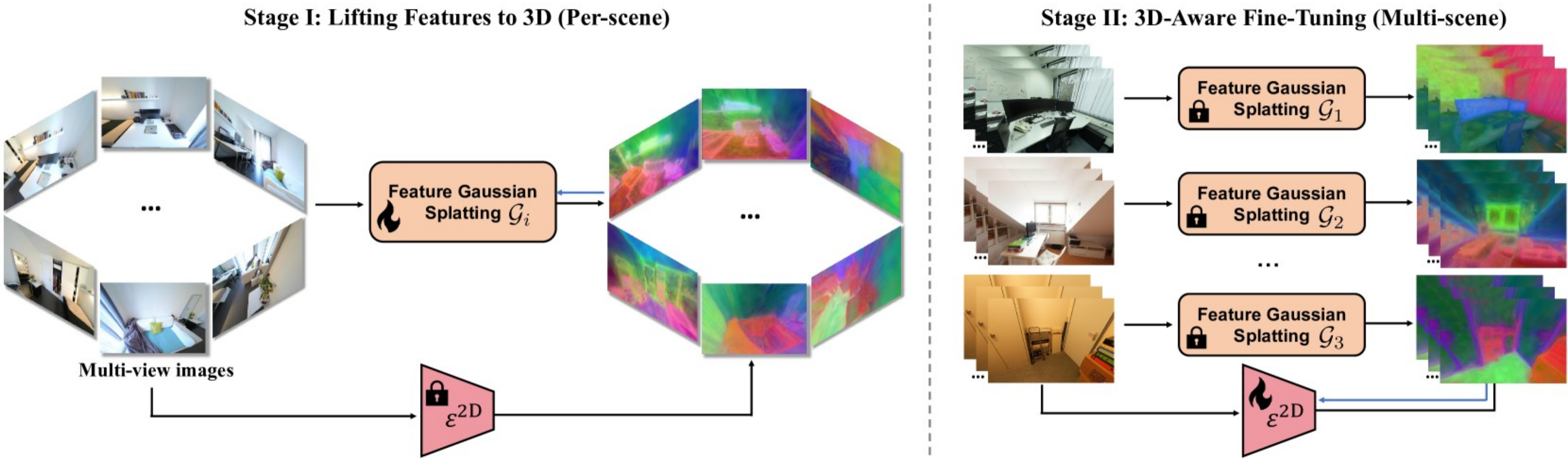
FiT3D

Improving 2D Feature Representations by 3D-Aware Fine-Tuning

Introduction to the problem

- **Problem →** Current vision models trained on 2D images lack true 3D scene understanding.
- **Motivation →** Human vision uses 3D structure cues for better understanding. Models should do the same.
- **Proposal →** A two-stage pipeline
  1. Lift 2D features to a 3D Gaussian representation
  2. Use the rendered 3D-aware features to fine-tune 2D models.

Method



- Lifting features to 3D with Feature Gaussian Splatting
- Fine-Tuning models with 3D-Awareness

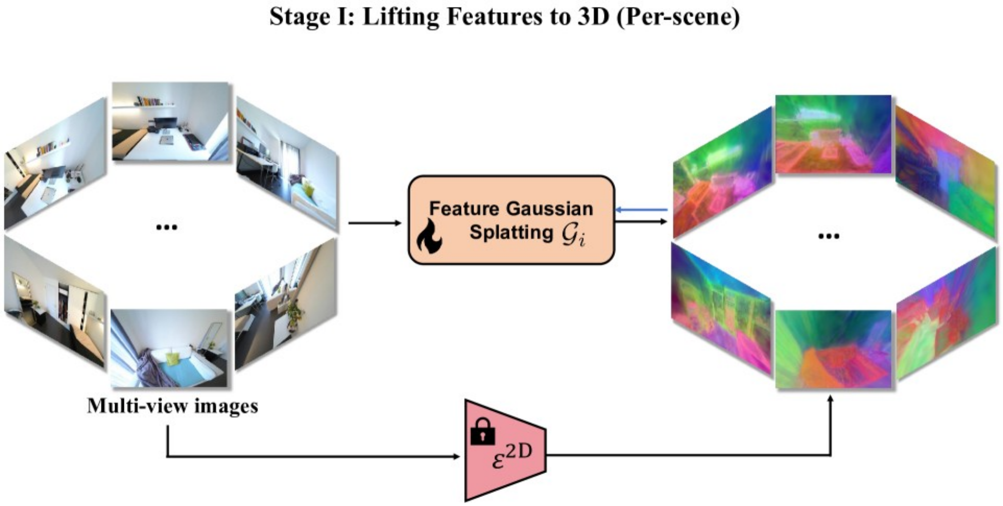
Stage 1, Lifting features to 3D: How?

- **Multi-view 2D features are encoded into 3D Gaussians**
- **A 3D Gaussian in this context represents a spatial point with appearance and feature attributes**

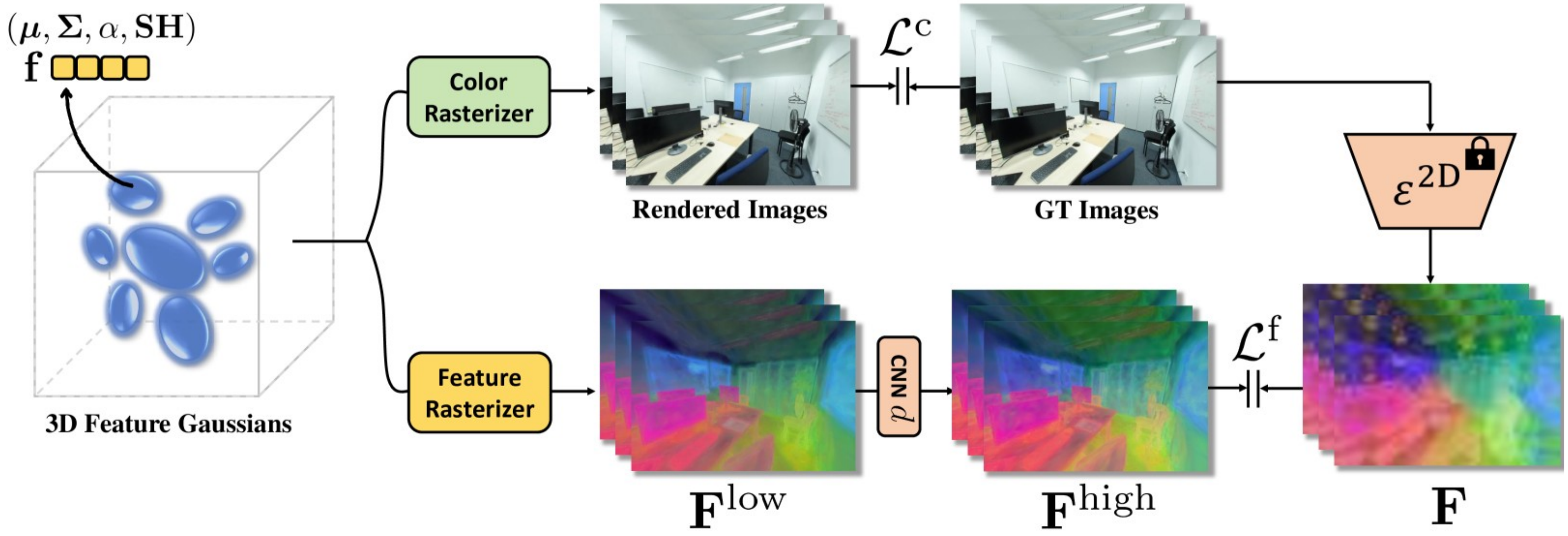
$$G = \{(\mu, s, R, \alpha, SH, f)_j\}_{1 \leq j \leq M}$$

Where:

- **Mu** is the position
- **s** is the scale, **R** the rotation
- **Alpha** is the opacity
- **SH** is the colour parameters
- **f** is a low-dim feature vector distilled from 2D features



3D Gaussian Splats and Feature Rasterization



- **To convert 3D Gaussians to a 2D feature image, we use a differentiable rasterizer: Alpha compositing, summing constribution of overlapping Gaussians**
- **A small scene-specific CNN is trained to transpose low-dim features back to high-dim space after rendering**

Stage 2, 3D-Aware Fine-Tuning

- **The Fine-Tuning algorithm is summarized as follows:**
  - Load the 3D Gaussians into CPU memory
  - Each training step:
    - Sample a training image Ii and its camera pose Pi.
    - Retrieve the corresponding 3D Gaussian G and scene-specific CNN decoder d
    - Render the 3D-aware features for the current view using G and d
    - Compute L1 loss between rendered features and 2D model output
    - Update Theta via backpropagation

Linear Probing for Downstream Tasks

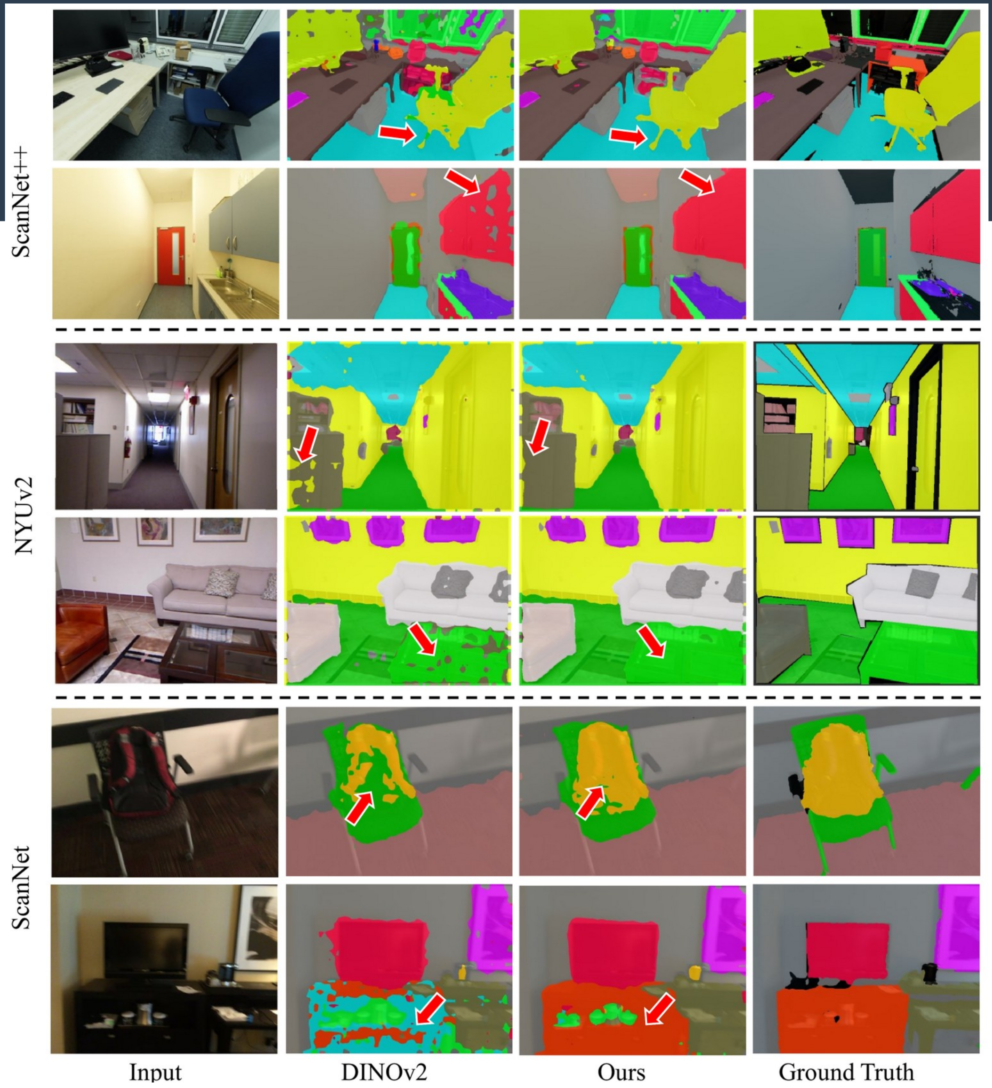
- Evaluation is done by training a shallow linear layer on top of extracted features (Linear Probing)
- Semantic segmentation → Done trough ViT tokens (patches), with the output upsampled to full resolution
- Depth Estimation → [CLS] token combined with each patch's feature to map them to depth bin probabilities and selects one from it, using cross-entropy as classification loss

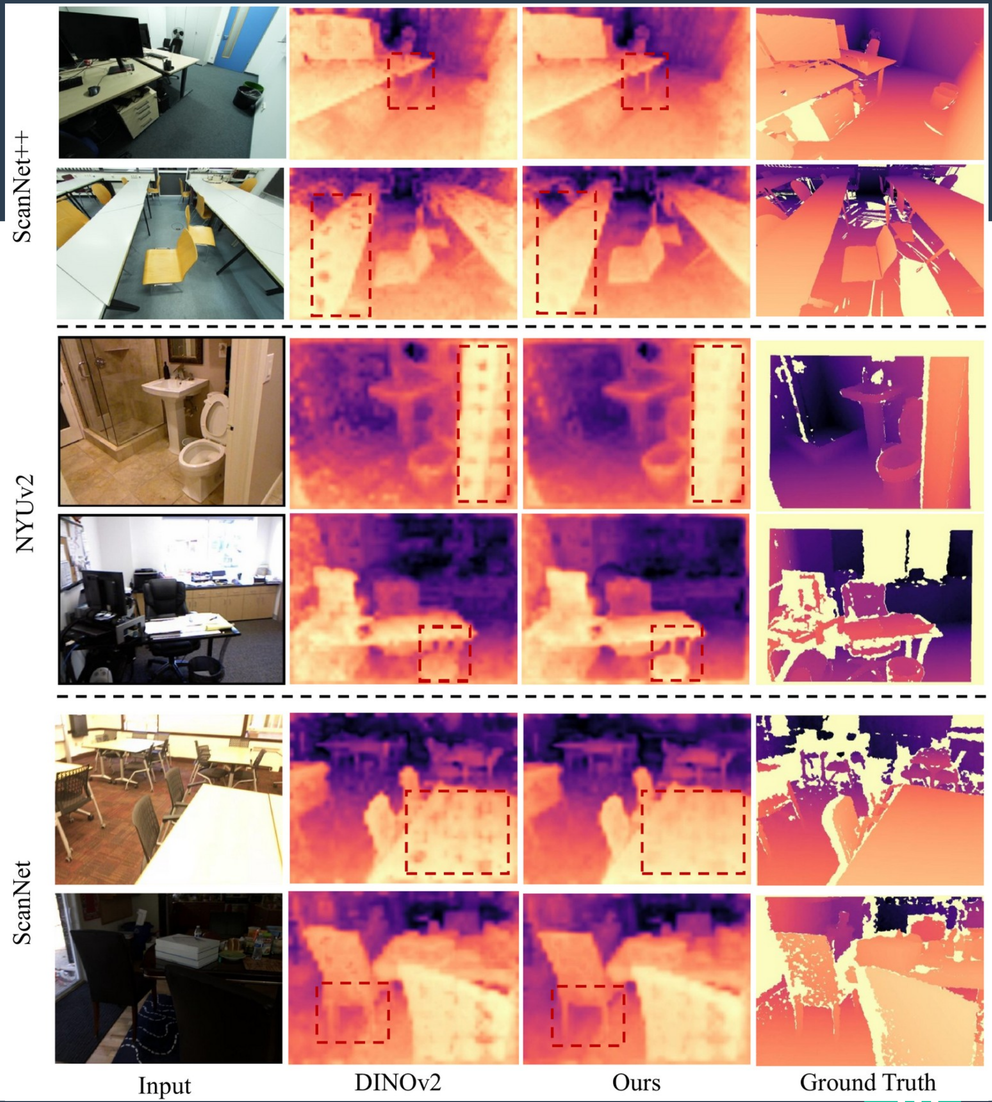
Experiments

- **Evaluation**
  - Linear probing on semantic segmentation and depth estimation
- **Models tested**
  - DINOv2 (Main)
  - CLIP, MAE, DeiT-III (for generalization)
- **injecting 3D-awareness into 2D models improves performance on vision tasks?**

Key Results

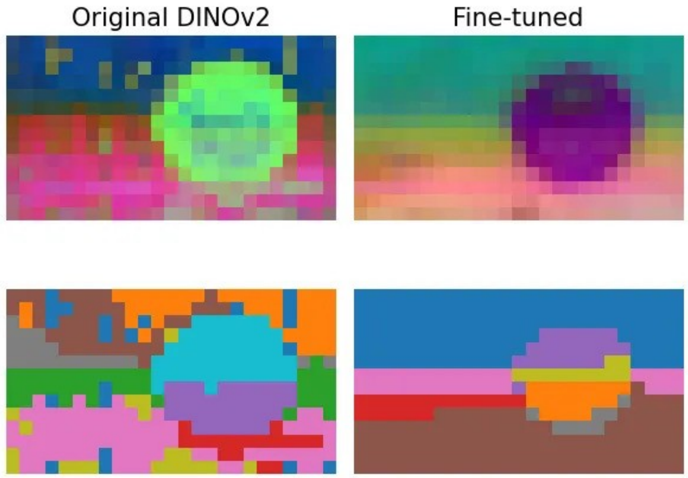
- **3D-aware features consistently outperform baseline DINOv2 on indoor datasets (ScanNet++, NYUv2, ScanNet).**
- **Semantic Segmentation (mean Intersection over Union ↑)**
  - +2.6% on ScanNet++
  - +2.0% on NYUv2
  - +1.2% on ScanNet
- **Depth Estimation (Root Mean Square Error ↓)**
  - 0.37 → 0.34 on ScanNet++
  - 0.44 → 0.42 on NYUv2
  - 0.31 → 0.29 on ScanNet
- **While still helping in generalized datasets evaluation (ADE20k, Pascal VOC, KITTI)**

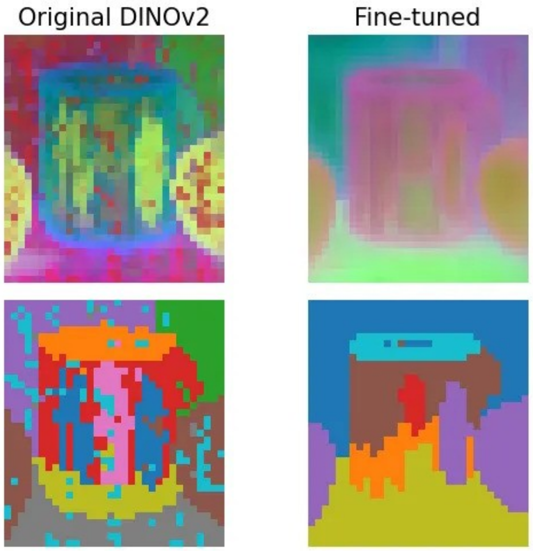

ScanNet++
NYUv2
ScanNet
Input
DINOv2
Ours
Ground Truth


ScanNet++
NYUv2
ScanNet
Input
DINOv2
Ours
Ground Truth

Reflective surfaces

- In the self-conducted experiment, surfaces with reflective properties show how the model is able to identify objects with no noise within the reflection, but misses specific dystorted elements
- The reflection is able to fool depth recognition before and after the fine tuning, showing little improvement over depth classification on a reflective surfaces





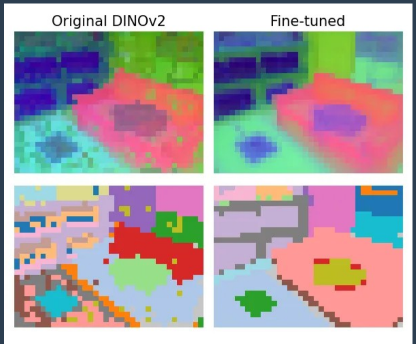
This is likely due to how Gaussian splats are calculated in the multi-view feature extraction phase

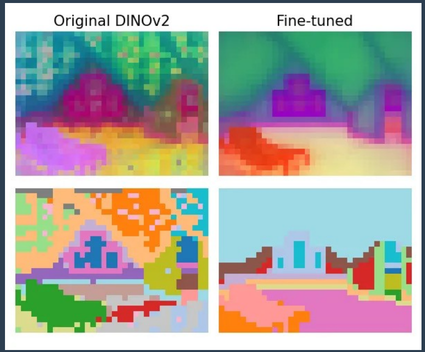
Ablation studies

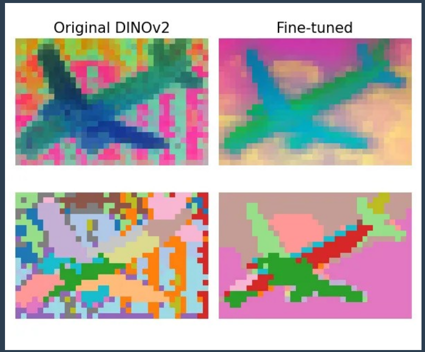
- Feature assembly strategy → The best one was to concatenate original and fine-tuned features
- Fine-tuning → 1 epoch is enough in 8.5 hours
- Classification tasks → There were no significant accuracy differences in the ImageNet results for classification

Conclusions and Key benefits

- The proposed method for augmenting 3D understanding resulted in significantly better semantic + geometric performance without labels or extra architecture
- Simple and scalable
- Fast fine-tuning, 1 epoch
- Improves multiple models
- Works out-of-domain
- No need for labeled data







Presented by Emanuele Di Sante