3D shape understanding

682: Neural Networks: A Modern Introduction

Subhransu Maji

April 21, 2026

College of
INFORMATION AND
COMPUTER SCIENCES

UMASS
AMHERST

Administrative

Midterm 2 on Tuesday, April 28 in class
Syllabus: Lecture 9 onwards (Image classification with CNNs) till today's lecture.

Homework 3 due today!

We will have the **project poster session** on May 7 (Thursday), from 4:00-7:30 PM at CSL Atrium. Everyone should attend the entire session.

Agenda (Recap)

3D representations

3D recognition architectures

- Muti-view methods
- Voxel-based methods
- Point-based methods

Recent trends

- Image to 3D

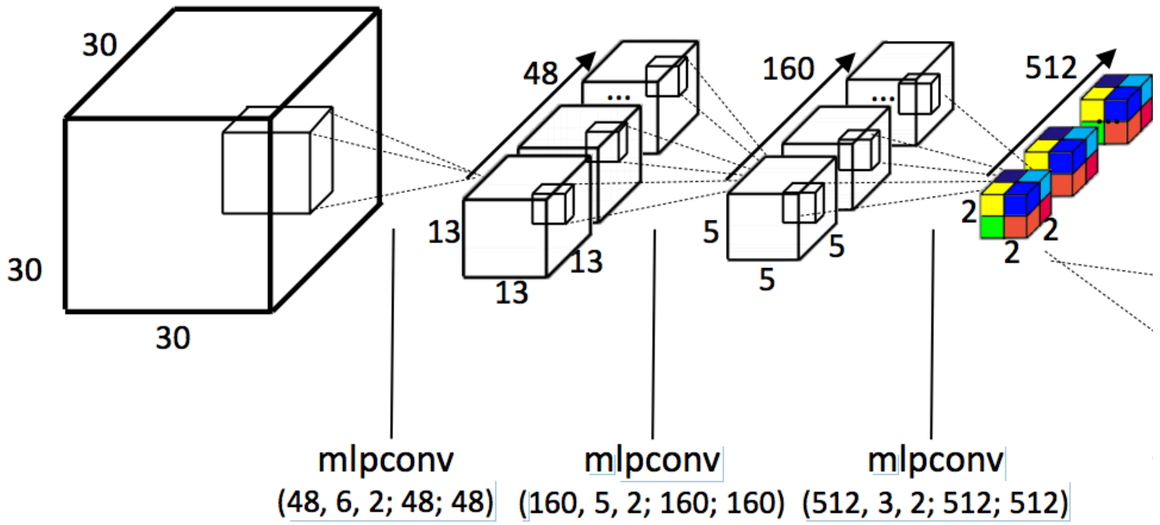
Slides credits: Hang Su & Hao Su

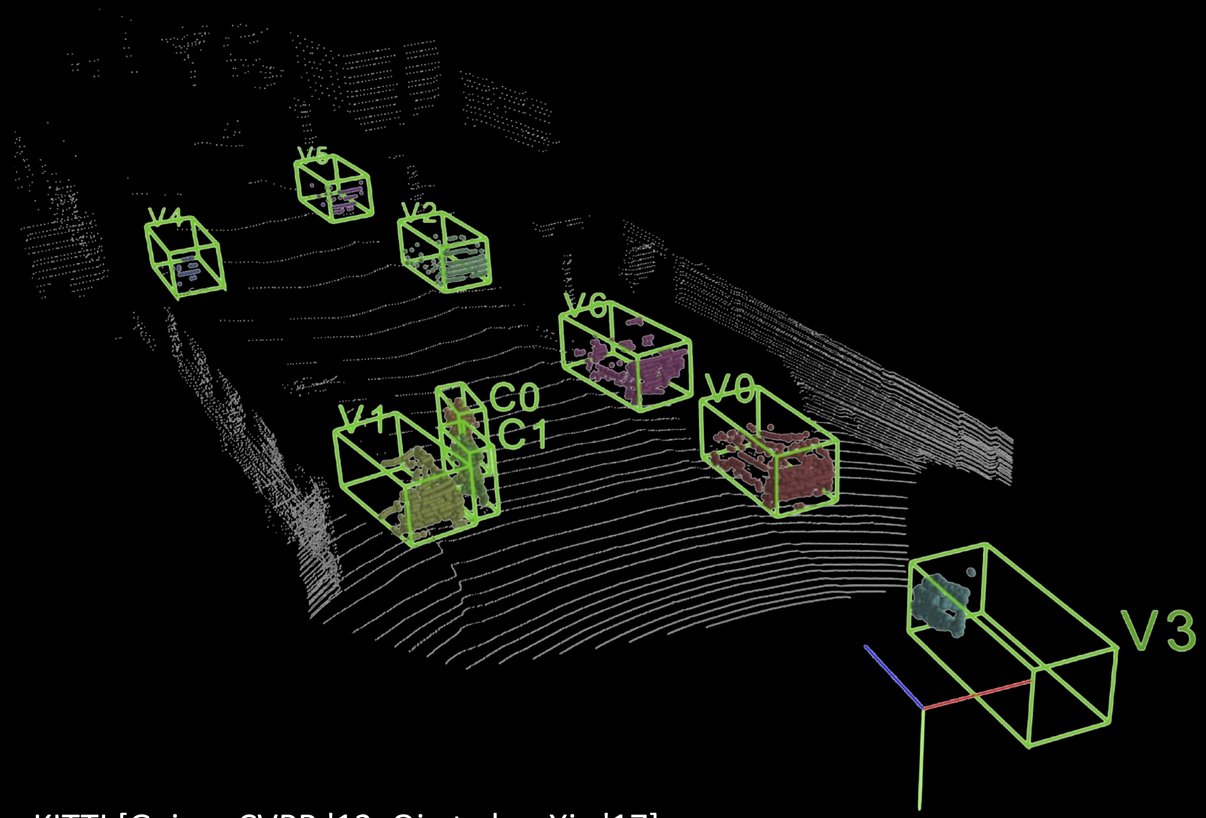
Multi-View CNN: Summary (Recap)

- A collection of 2D views is highly informative for recognizing 3D objects.
- Using 2D CNNs allows leveraging powerful image architectures and available supervision.
- MVCNNs relate 3D shapes to 2D images, enabling cross-domain applications.

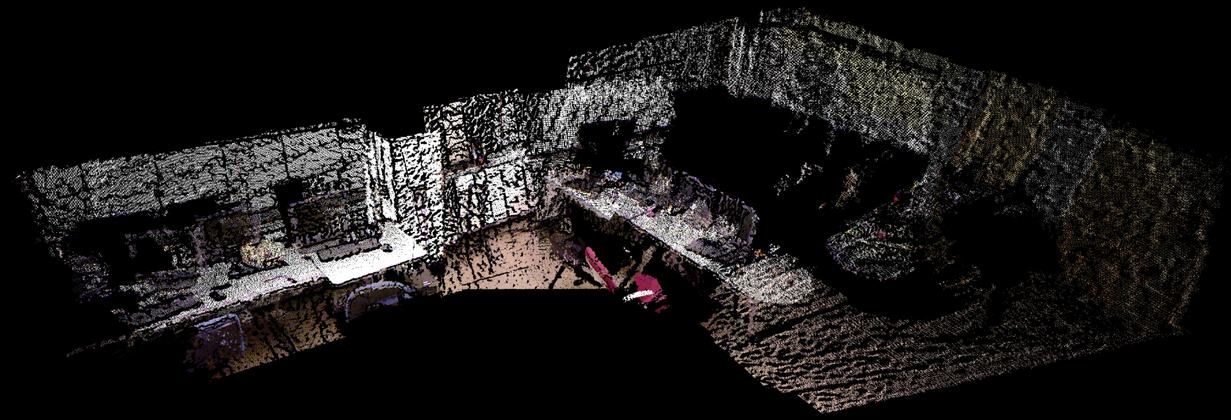
3D CNN on Volumetric Data (Recap)

3D convolution uses 4D kernels

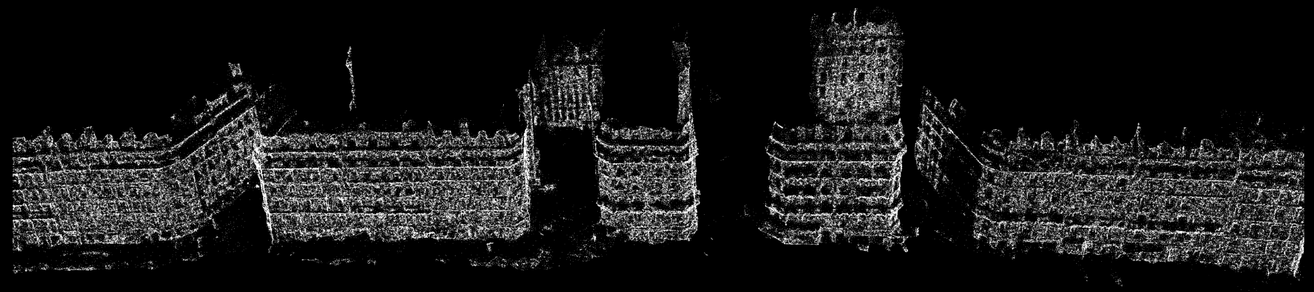




KITTI [Geiger CVPR '12, Qi et al. arXiv '17]



Cornell-RGBD [Koppula et al. NIPS '11]



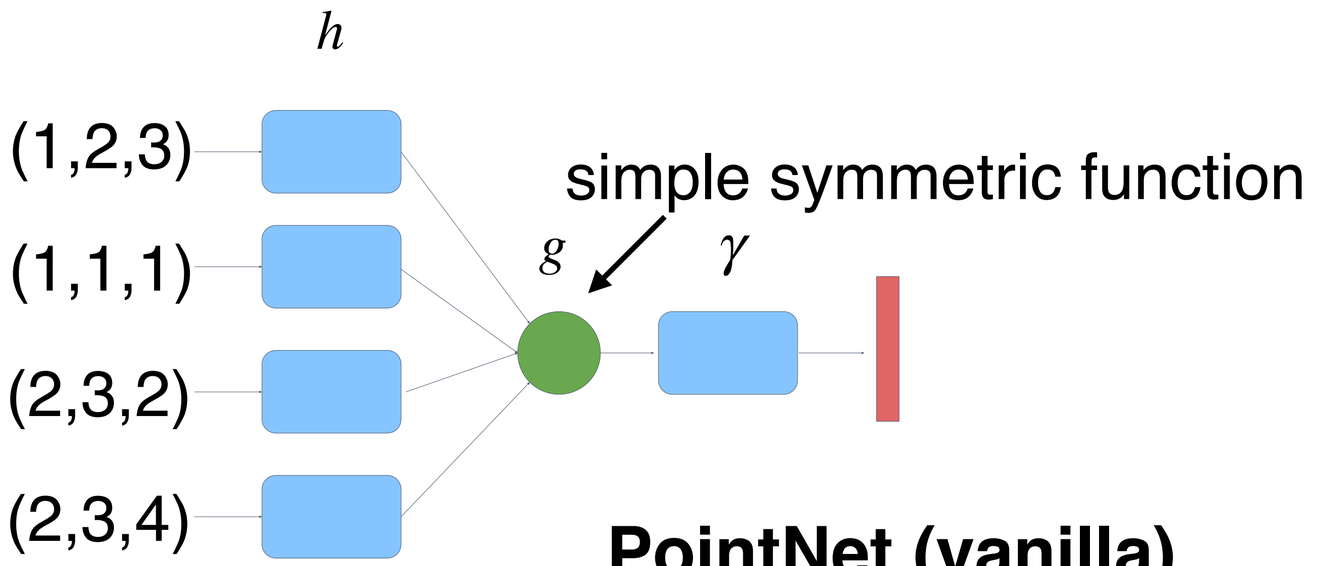
Ruemonge2014 [Riemenschneider et al. ECCV '14]

Point clouds are highly sparse, and lack of grid structure.

Construct a Symmetric Function (Recap)

Observe:

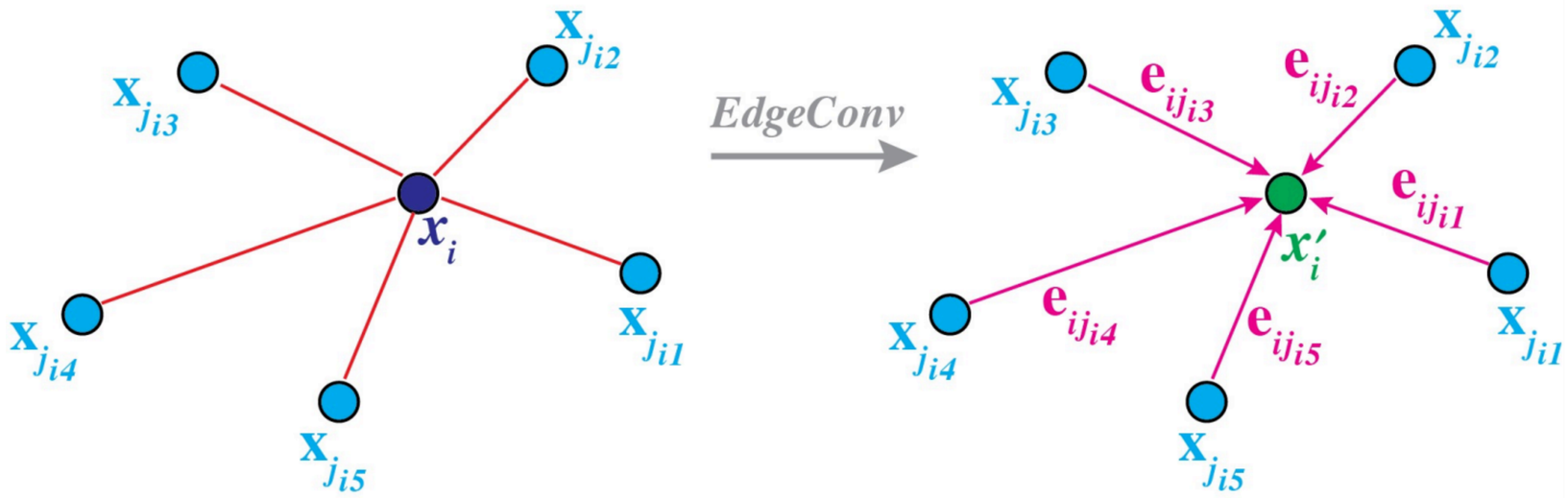
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet (vanilla)

Point Convolution As Graph Convolution (Recap)

- Points -> Nodes
- Neighborhood -> Edges
- Graph CNN for point cloud processing



Wang et al., "**Dynamic Graph CNN for Learning on Point Clouds**", *Transactions on Graphics, 2019*

Liu et al., "**Relation-Shape Convolutional Neural Network for Point Cloud Analysis**", *CVPR 2019*

High-Dimensional Filtering

$$\mathbf{v}_i' = \sum_{j \in \Omega(i)} \mathbf{W}[\mathbf{p}_j - \mathbf{p}_i]\mathbf{v}_j \longrightarrow \mathbf{v}_i' = \sum_{j \in \Omega(i)} \mathbf{W}[\mathbf{f}_j - \mathbf{f}_i]\mathbf{v}_j$$

positions

input value

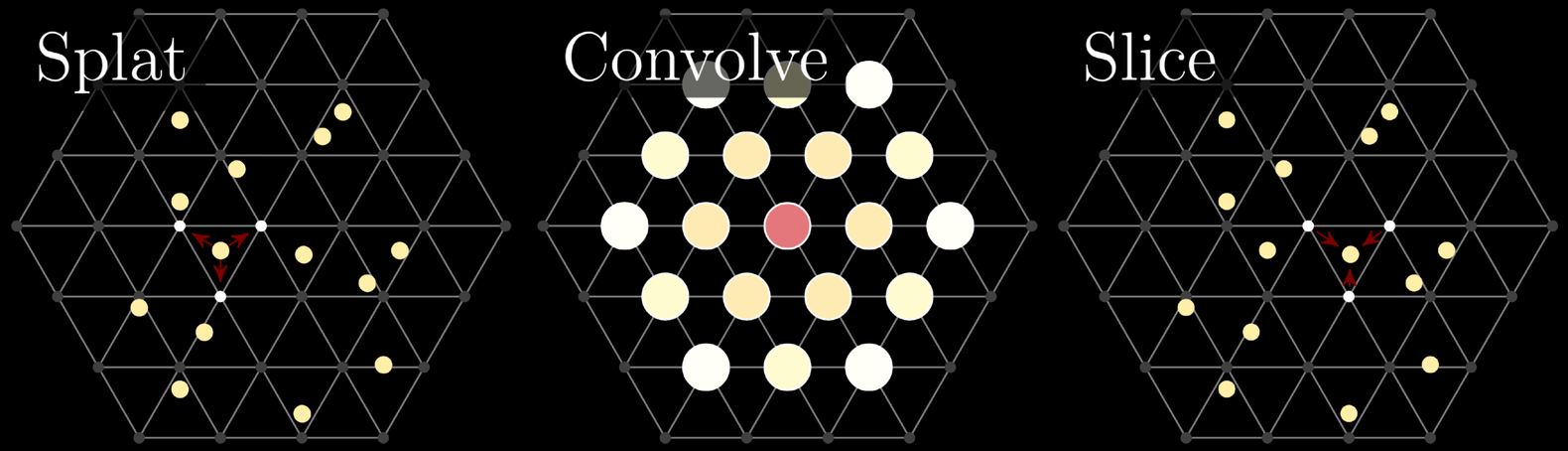
spatial convolution

high-dim. filtering

			Bilateral filter	Conv2d	
Image	pixel value $\mathbf{v}$	(r, g, b)	(r, g, b)	$\mathbf{v}(\dots)$	$\mathbf{v}(\dots)$
	pixel position $\mathbf{f}$	(x, y)	(x, y, r, g, b)	(x, y)	$(x, y, depth)$
3D Point cloud	point value $\mathbf{v}$	1	(x, y, z)	(r, g, b)	$\mathbf{v}(\dots)$
	point position $\mathbf{f}$	(x, y, z)	(x, y, z)	(x, y, z, n_x, n_y, n_z)	(x, y, z)

Efficient Sparse High-Dimensional Filtering (Recap)

Bilateral Convolution Layer (BCL) [1,2,3]



[1] A. Adams, J. Baek and M. A. Davis. Fast High-Dimensional Filtering Using the Permutohedral Lattice. Computer Graphics Forum '10
[2] V. Jampani, M. Kiefel and P. V. Gehler. Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks. CVPR '16
[3] M. Kiefel, V. Jampani and P. V. Gehler. Permutohedral Lattice CNNs. ICLR '15 workshops

Recent trends

Image (s) to 3D

- **Task:** Given input image(s), estimate the underlying 3D structure.
- **Scale ambiguity:** Outputs are often estimated only up to an unknown scale

- **Monocular depth estimation:** Single image to depth
  - Early models were ConvNet-based and trained on small datasets
  - Recent models are based on diffusion models and transformers, and are trained on massive datasets.

- **Multi-view 3D:** Multiple images as input; simultaneously estimate camera poses and the underlying geometry for a more complete 3D reconstruction
- Recent methods are often transformer-based and trained on massive datasets

Monocular depth estimation

Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation

Bingxin Ke Anton Obukhov Shengyu Huang
Nando Metzger Rodrigo Caye Daudt Konrad Schindler
Photogrammetry and Remote Sensing, ETH Zürich

Monocular depth estimation

Fine-tuned “Stable Diffusion” to model **P(depth I image)**

Training takes 2.5 days on a single NVIDIA RTX 4090 GPU

Trained on **Hypersim** (a photorealistic indoor scene dataset)

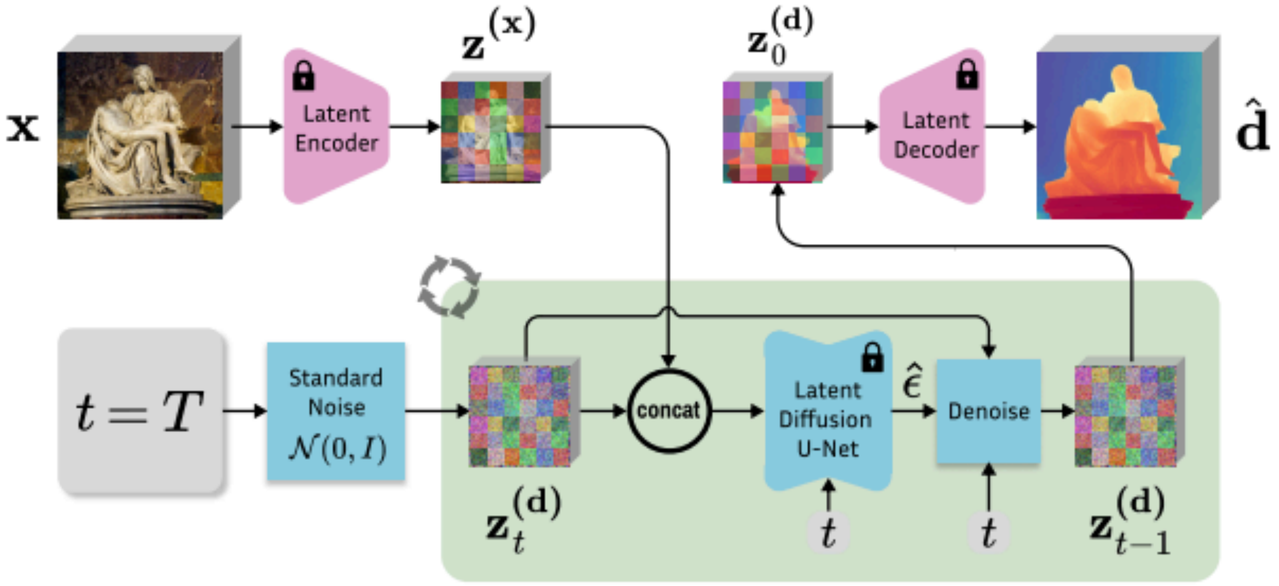
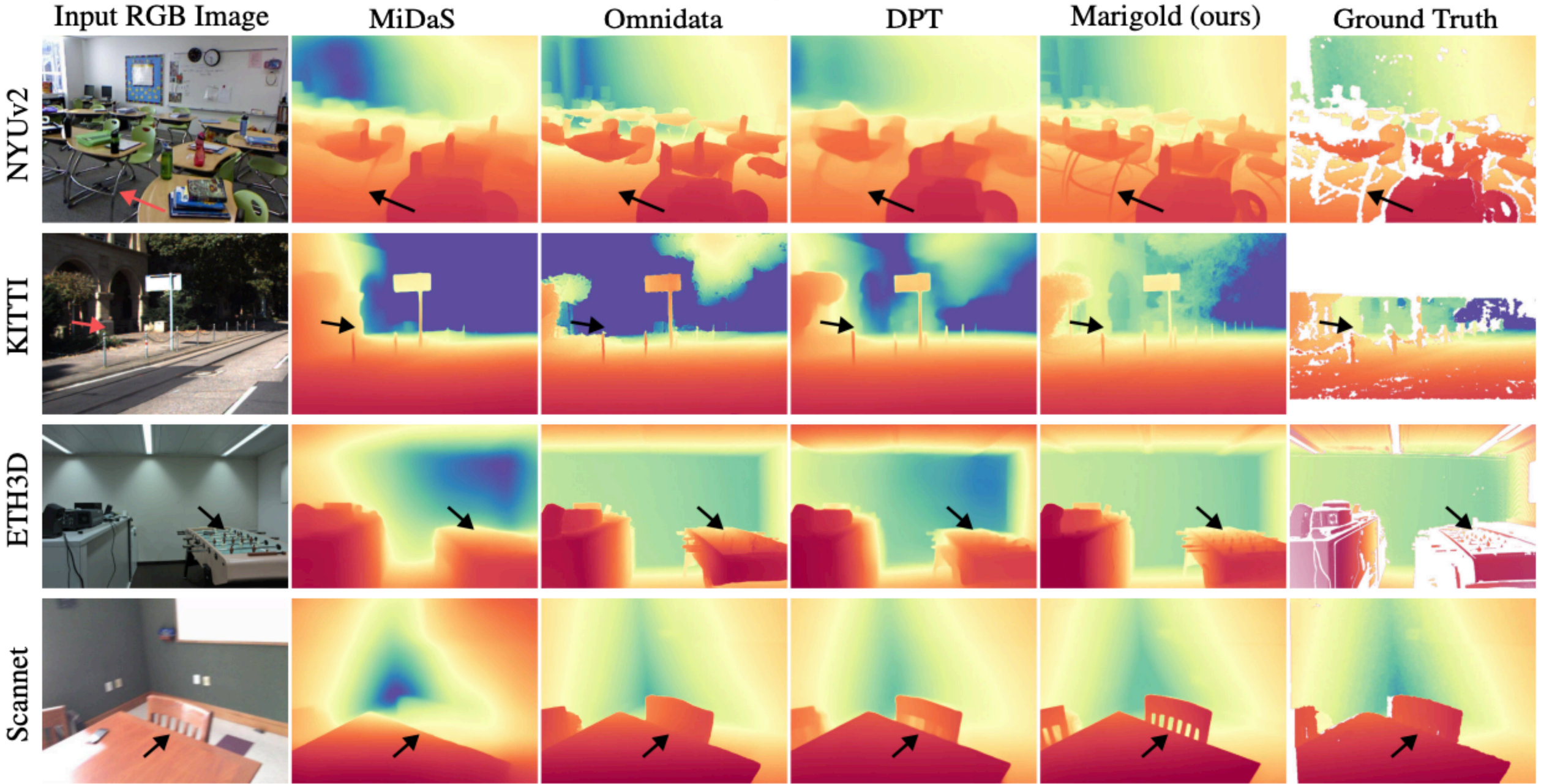


Figure 3. **Overview of the Marigold inference scheme.** Given an input image $\mathbf{x}$, we encode it with the original Stable Diffusion VAE into the latent code $\mathbf{z}^{(\mathbf{x})}$, and concatenate with the depth latent $\mathbf{z}_t^{(\mathbf{d})}$ before giving it to the modified fine-tuned U-Net on every denoising iteration. After executing the schedule of T steps, the resulting depth latent $\mathbf{z}_0^{(\mathbf{d})}$ is decoded into an image, whose 3 channels are averaged to get the final estimation $\hat{\mathbf{d}}$. See Sec. 3.4 for details.

Monocular depth estimation



DUSt3R: Geometric 3D Vision Made Easy

Shuzhe Wang*, Vincent Leroy†, Yohann Cabon†, Boris Chidlovskii† and Jerome Revaud†
*Aalto University
shuzhe.wang@aalto.fi
†Naver Labs Europe
firstname.lastname@naverlabs.com

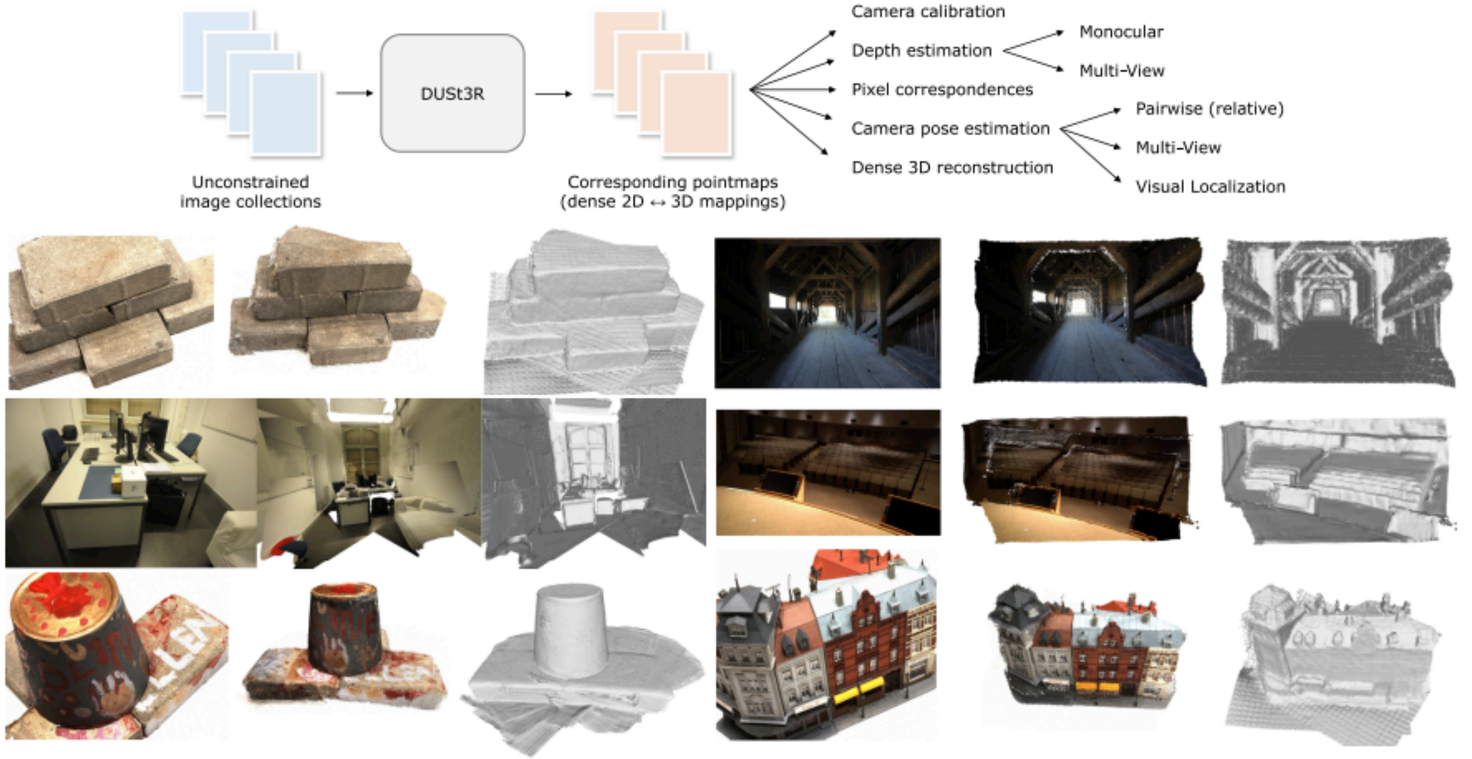


Figure 1. **Overview**: Given an unconstrained image collection, *i.e.* a set of photographs with unknown camera poses and intrinsics, our proposed method **DUSt3R** outputs a set of corresponding *pointmaps*, from which we can straightforwardly recover a variety of geometric quantities normally difficult to estimate all at once, such as the camera parameters, pixel correspondences, depthmaps, and fully-consistent 3D reconstruction. Note that DUSt3R also works for a single input image (*e.g.* achieving in this case monocular reconstruction). We also show **qualitative examples** on the DTU, Tanks and Temples and ETH-3D datasets [1, 51, 108] obtained **without** known camera parameters. For each sample, from *left* to *right*: input image, colored point cloud, and rendered with shading for a better view of the underlying geometry.

DUSt3R: Geometric 3D Vision Made Easy

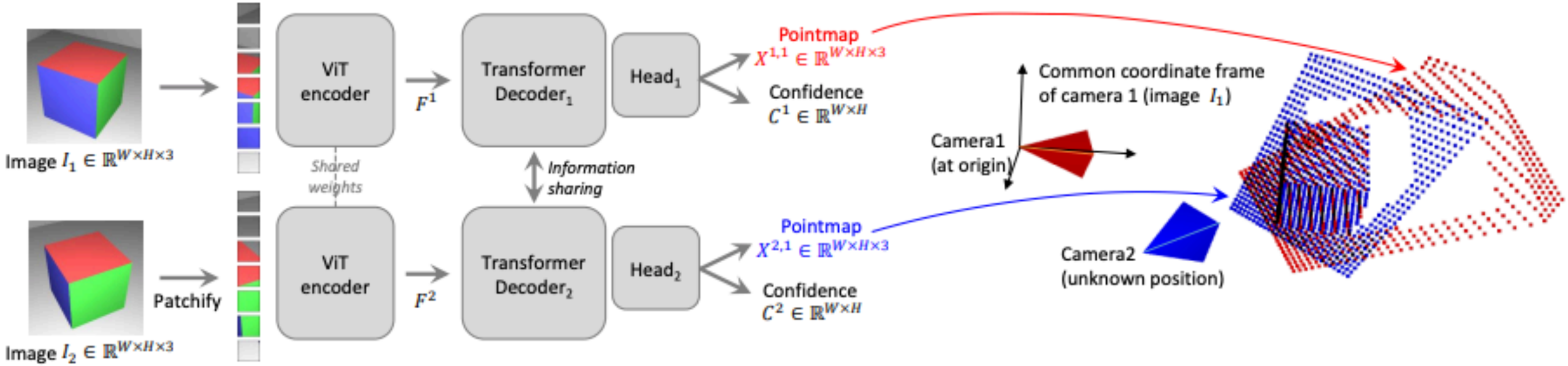
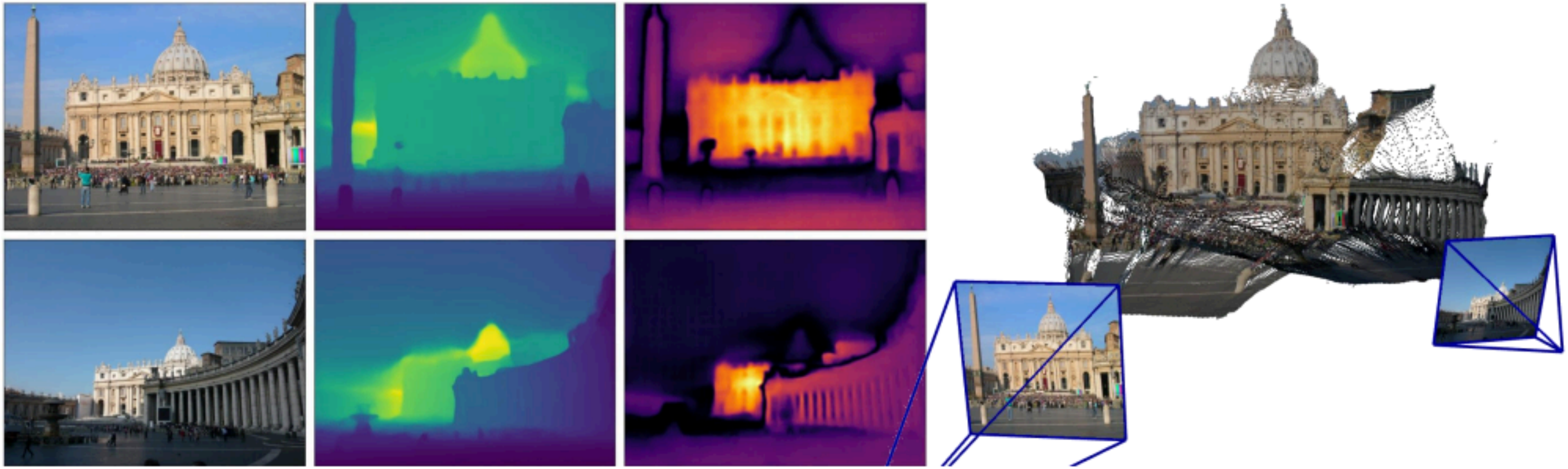
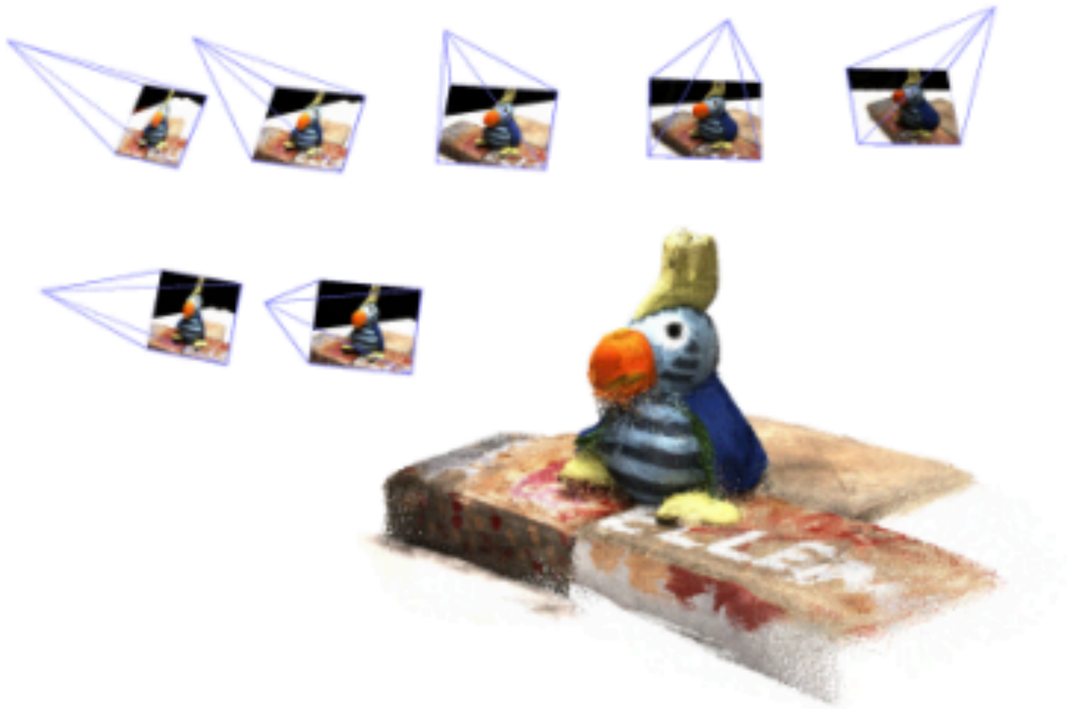


Figure 2. **Architecture of the network $\mathcal{F}$.** Two views of a scene (I^1, I^2) are first encoded in a Siamese manner with a shared ViT encoder. The resulting token representations F^1 and F^2 are then passed to two transformer decoders that constantly exchange information via cross-attention. Finally, two regression heads output the two corresponding pointmaps and associated confidence maps. Importantly, the two pointmaps are expressed in the same coordinate frame of the first image I^1. The network $\mathcal{F}$ is trained using a simple regression loss (Eq. (4))

DUSt3R: Geometric 3D Vision Made Easy



DUSt3R: Geometric 3D Vision Made Easy



VGGT: Visual Geometry Grounded Transformer

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 [2]Meta AI

Figure 1. **VGGT** is a large feed-forward transformer with minimal 3D-inductive biases trained on a trove of 3D-annotated data. It accepts up to hundreds of images and predicts cameras, point maps, depth maps, and point tracks for all images at once in less than a second, which often outperforms optimization-based alternatives without further processing.

VGGT: Visual Geometry Grounded Transformer

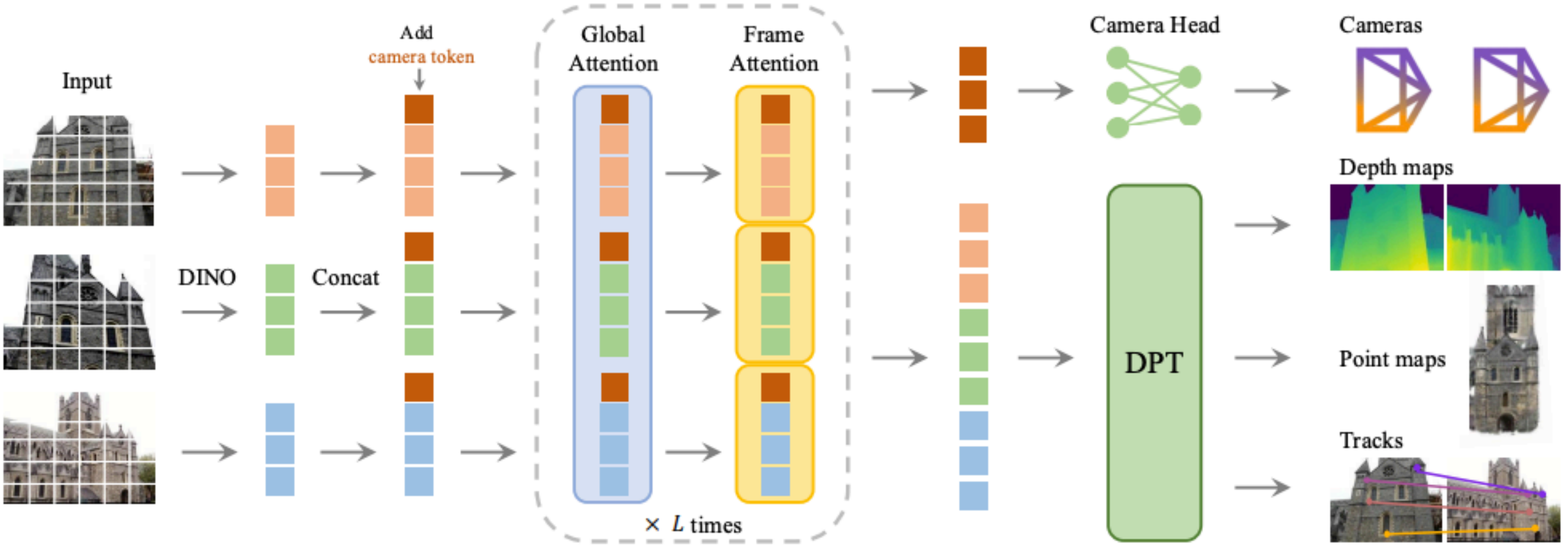


Figure 2. **Architecture Overview.** Our model first patchifies the input images into tokens by DINO, and appends camera tokens for camera prediction. It then alternates between frame-wise and global self attention layers. A camera head makes the final prediction for camera extrinsics and intrinsics, and a DPT [87] head for any dense output.

Depth Anything 3

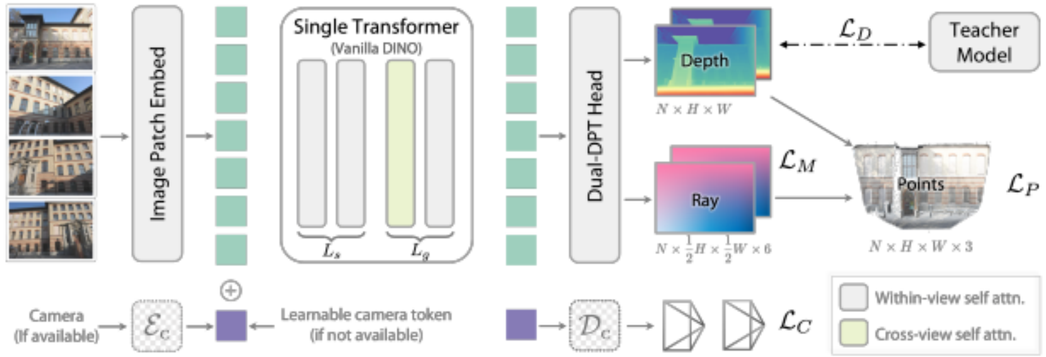


Figure 2 Pipeline of Depth Anything 3. Depth Anything 3 employs a single transformer (vanilla DINOv2 model) without any architectural modifications. To enable cross-view reasoning, an input-adaptive cross-view self-attention mechanism is introduced. A dual-DPT head is used to predict depth and ray maps from visual tokens. Camera parameters, if available, are encoded as camera tokens and concatenated with patch tokens, participating in all attention operations.

Depth Anything 3:
Recovering the Visual Space from Any Views

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

ByteDance Seed

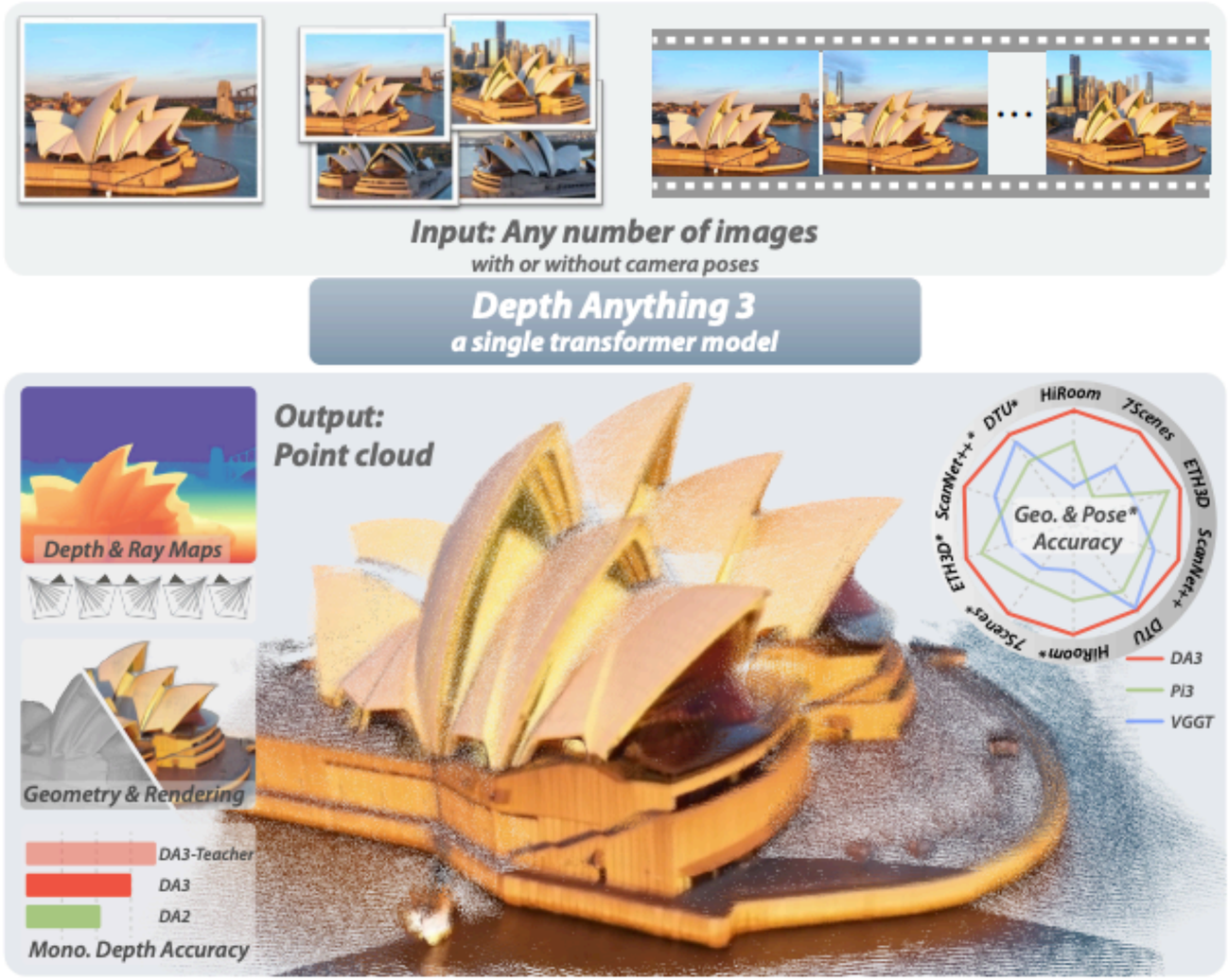


Figure 1 Given any number of images and optional camera poses, **Depth Anything 3** reconstructs the visual space, producing consistent depth and ray maps that can be fused into accurate point clouds, resulting in high-fidelity 3D Gaussians and geometry. It significantlyoutperforms VGGT in multi-view geometry and pose accuracy; with monocular inputs, it also surpasses Depth Anything 2 while matching its detail and robustness.

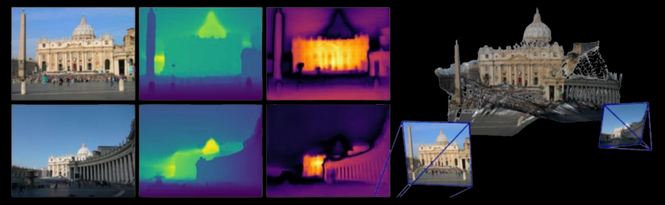
Summary

- Interests in **3D recognition models** are rapidly growing as 3D sensors and datasets become more accessible.

- Extending CNNs to 3D is non-trivial because 3D data like **polygon meshes** or **point clouds** lack regular grid structure.

- Recent advances in image to 3D combining **classical geometric pipelines** for **multi-view reconstruction**, **synthetic data**, **transformer architectures**.

- Many techniques based on **view-based methods, sparse convolutions, graph neural networks**, **transformers**, etc.



- However, **3D data** is still lacking relative to **images.**