FVL2025第十期学习讲座

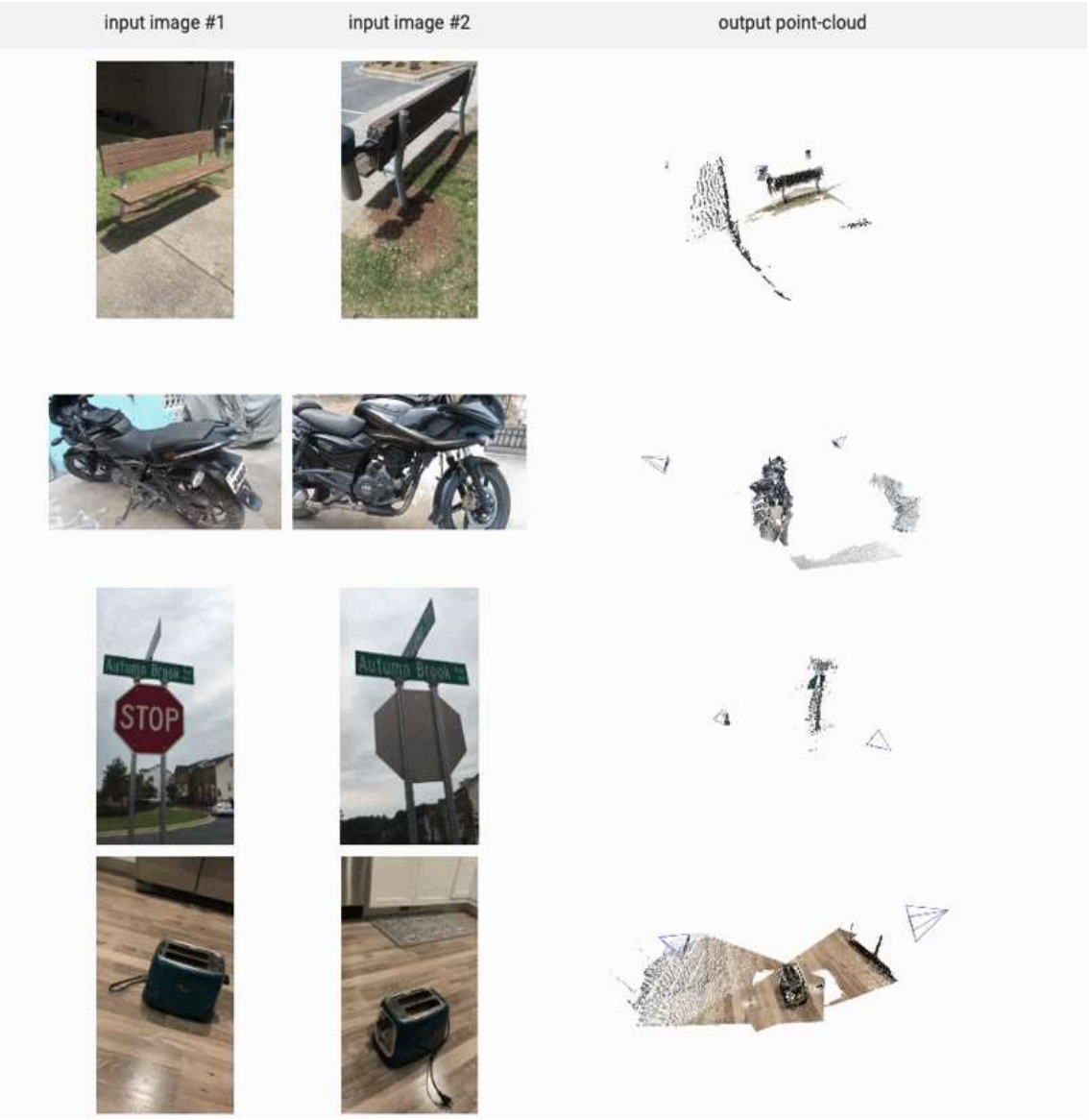
端到端三维重建

• 院系：智能工程学院 • 专业：控制科学与工程 • 汇报人：国翔宇 • 汇报时间：2025.05.25

1. DUSt3R: Geometric 3D Vision Made Easy 2024 CVPR

2. Easi3R: Estimating Disentangled Motion from DUSt3R Without Training 2025 arXiv

3. VGGT: Visual Geometry Grounded Transformer CVPR 2025 (Oral)

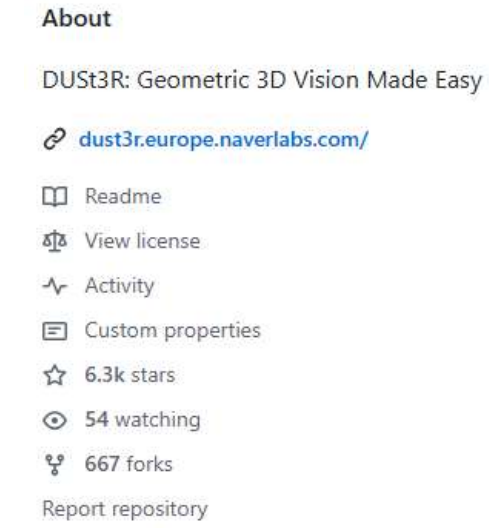


DUSt3R: Geometric 3D Vision Made Easy **CVPR 2024**

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

Dense and **U**nconstrained **St**ereo **3**D **R**econstruction of arbitrary image collections

给定一个不受约束的图像集合，即一组具有未知相机姿势和内在特性的照片，DUSt3R 输出一组相应的点图，从中可以直接恢复通常难以一次性估计的各种几何量，例如相机参数、像素对应关系、深度图和完全一致的 3D 重建。



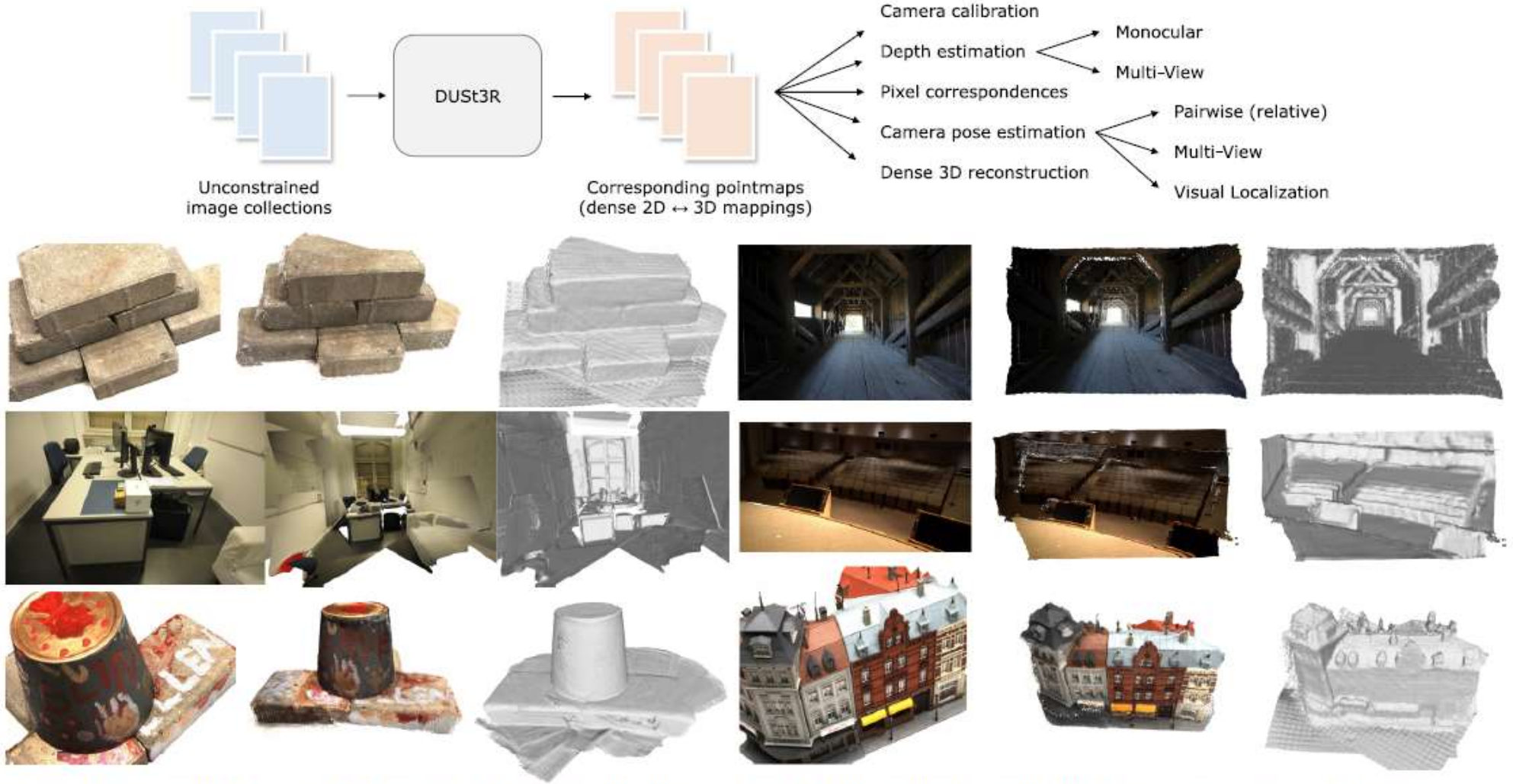


Figure 1. **Overview**: Given an unconstrained image collection, *i.e.* a set of photographs with unknown camera poses and intrinsics, our proposed method **DUSt3R** outputs a set of corresponding *pointmaps*, from which we can straightforwardly recover a variety of geometric quantities normally difficult 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.

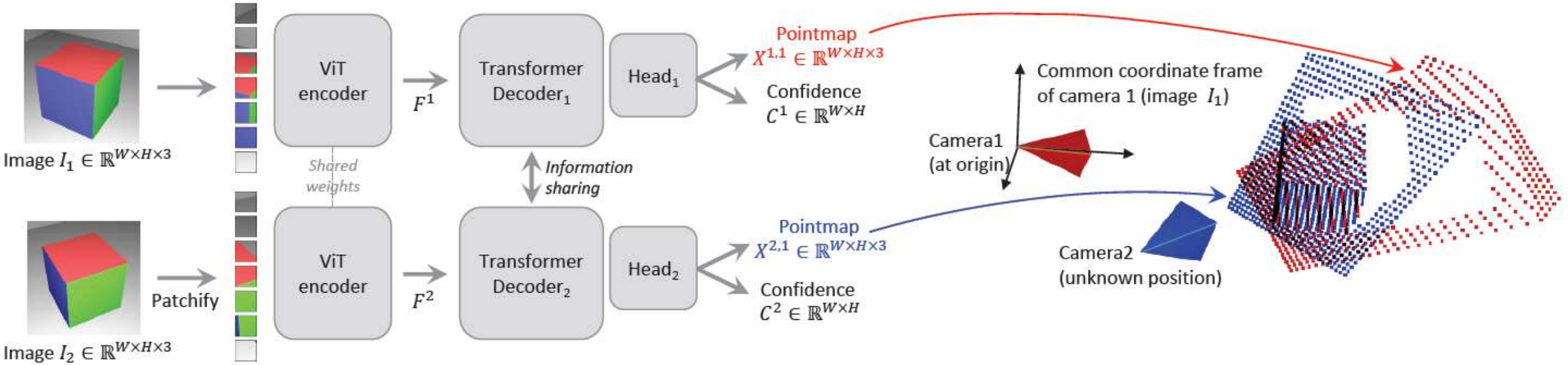


Figure 2. **Architecture of the network $\mathcal{F}$.** Two views of a scene (I^1, I^2) are first encoded in a Siamese manner with a shared ViT encoder. The resulting token representations F^1 and F^2 are then passed to two transformer decoders that constantly exchange information via cross-attention. 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))

$$X^{n,m} = P_m P_n^{-1} h\left(X^n\right) \tag{1}$$

$$\ell_{\text{regr}}(v,i) = \left\| \frac{1}{z} X_i^{v,1} - \frac{1}{\bar{z}} \bar{X}_i^{v,1} \right\| . \tag{2}$$

$$\text{norm}(X^1, X^2) = \frac{1}{|\mathcal{D}^1| + |\mathcal{D}^2|} \sum_{v\in\{1,2\}} \sum_{i\in\mathcal{D}^v} \|X_i^v\| . \tag{3}$$

$$\mathcal{L}_{\text{conf}} = \sum_{v\in\{1,2\}} \sum_{i\in\mathcal{D}^v} C_i^{v,1} \ell_{\text{regr}}(v,i) - \alpha \log C_i^{v,1}, \tag{4}$$

Global Alignment

Pairwise graph. $\{I^1, I^2, \ldots, I^N\}$ $\mathcal{G}(\mathcal{V}, \mathcal{E})$

- Existing off-the-shelf image retrieval methods
- Through network （inference takes ≈40ms on a H100 GPU）

Global optimization

$$\chi^* = \arg\min_{\chi, P, \sigma} \sum_{e \in \mathcal{E}} \sum_{v \in e} \sum_{i=1}^{HW} C_i^{v,e} \left\| \chi_i^v - \sigma_e P_e X_i^{v,e} \right\|. \quad (5)$$

Figure 3. **Reconstruction examples** on two scenes never seen during training. From left to right: RGB, depth map, confidence map, reconstruction. The left scene shows the raw result output from $\mathcal{F}(I^1, I^2)$. The right scene shows the outcome of global alignment (Sec. 3.4).

Dataset mixture. DUSt3R is trained with a mixture of eight datasets: Habitat [104], ARKitScenes [25], MegaDepth [56], Static Scenes 3D [110], Blended MVS [162], ScanNet++ [166], CO3Dv2 [94] and Waymo [122]. These datasets feature diverse scene types: indoor, outdoor, synthetic, real-world, object-centric, etc. Table 8 shows the number of extracted pairs in each datasets, which amounts to 8.5M in total.

Datasets	Type	N Pairs
Habitat [104]	Indoor / Synthetic	1000k
CO3Dv2 [94]	Object-centric	941k
ScanNet++ [166]	Indoor / Real	224k
ArkitScenes [25]	Indoor / Real	2040k
Static Thing 3D [110]	Object / Synthetic	337k
MegaDepth [56]	Outdoor / Real	1761k
BlendedMVS [162]	Outdoor / Synthetic	1062k
Waymo [122]	Outdoor / Real	1100k

Table 8. Dataset mixture and sample sizes for DUSt3R training.

yocabon on Mar 4, 2024 · edited by yocabon Edits Contributor

Hi,

We trained dust3r on A100 gpus (with 80GB of vram)
for the training at 224x224 resolution, we used 4, but I'd recommend to use 8 (it increases the effective batch size to 128 - we also tried that and there's barely any difference).
512 linear and dpt both were trained using 8 A100 gpus.

About timings:

224: `~0.59s` per batch, `8*100_000` pairs per per epoch, that's `6_250` steps if running on 8 gpus, or `~1.02` hours per epoch. About `102.4` hours total (test passes/saving checkpoints will increase that a bit, but they are not long compared to training).
512 linear: `~0.63s` per step (accum_iter=2, so 2 steps per effective batch), `8*10_000` pairs per per epoch, `2_500` steps per epoch, `26.25` minutes per epoch, `87.5` hours total
512 dpt: `~0.52s` per step (accum_iter=4, so 4 steps per effective batch), `8*10_000` pairs per per epoch, `5_000` steps per epoch, `43` minutes per epoch, `65` hours total. Note, the first epoch is much slower when using dpt.

23

Relation between depthmaps and pointmaps. As a result, the depth value $D^1_{i,j}$ at pixel (i, j) in image I^1 can be recovered as

$$D^1_{i,j} = \bar{X}^{1,1}_{i,j,2}. \quad (8)$$

Therefore, all depthmaps displayed in the main paper and this appendix are straightforwardly extracted from DUSt3R's output as $X^{1,1}_{:,:,2}$ and $X^{2,2}_{:,:,2}$ for images I^1 and I^2, respectively.

Easi3R: Estimating Disentangled Motion from DUSt3R Without Training

Xingyu Chen[1] Yue Chen[1] Yuliang Xiu[1,2] Andreas Geiger[3] Anpei Chen[1,3]
[1]Westlake University [2]Max Planck Institute for Intelligent Systems
[3]University of Tübingen, Tübingen AI Center
easi3r.github.io

引入了一个简单、高效的免训练4D重建方法Easi3R。在推理时应用attention adaptation，消除了从头开始预训练或网络微调的需求。

发现DUSt3R中的注意力层内在地编码了关于相机和物体运动的丰富信息。通过分解注意力图，可以实现准确的动态区域分割、相机姿态估计以及4D密集点云重建。

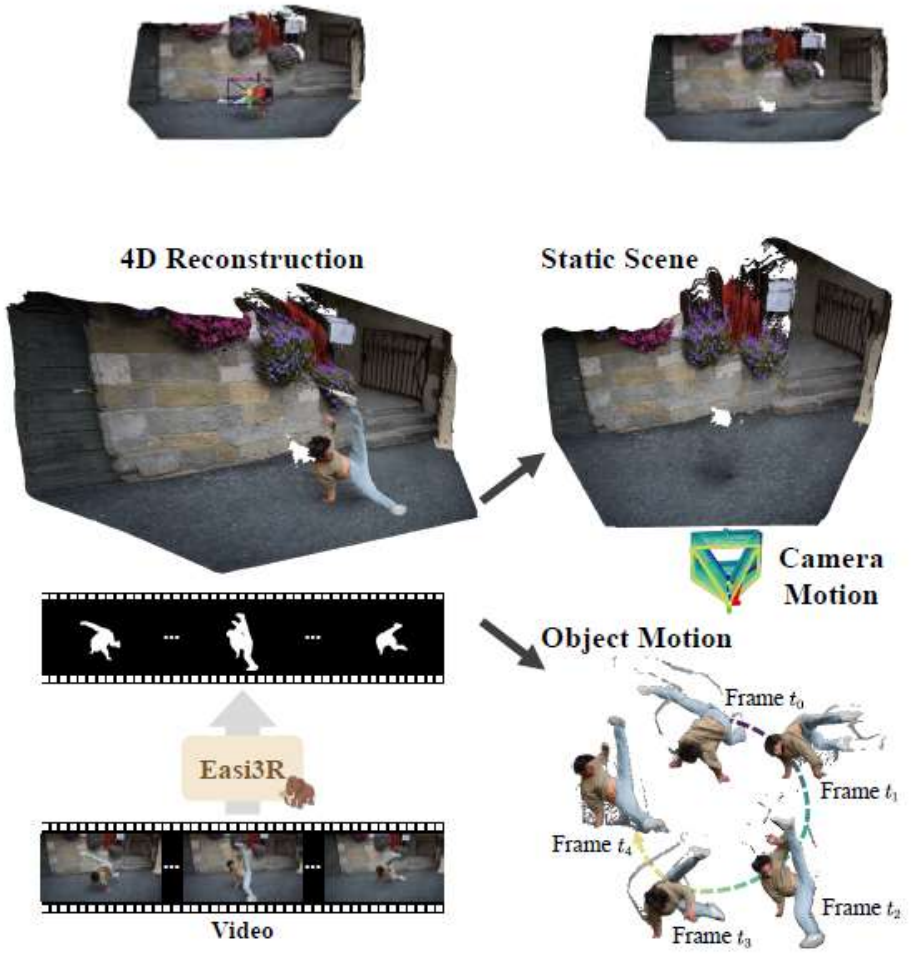


Figure 1. We present Easi3R, a training-free, plug-and-play approach that efficiently disentangles object and camera motion, enabling the adaptation of DUSt3R for 4D reconstruction.

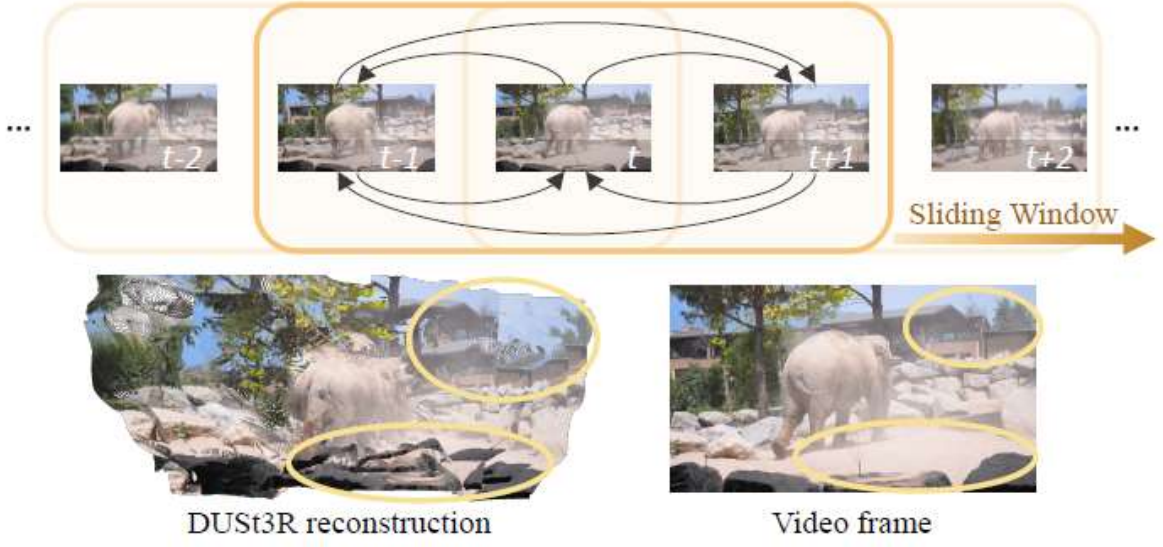


Figure 2. **DUSt3R with Dynamic Video.** We process videos us-

$$\mathbf{M}^{a\leftarrow b} = (1 - \mathbf{M}^{a}) \otimes \mathbf{M}^{b^T} \qquad \text{softmax}(\tilde{\mathbf{A}}_l^{a\leftarrow b}) = \begin{cases} 0 & \text{if } \mathbf{M}^{a\leftarrow b} \\ \text{softmax}(\mathbf{A}_l^{a\leftarrow b}) & \text{otherwise} \end{cases} \tag{10}$$

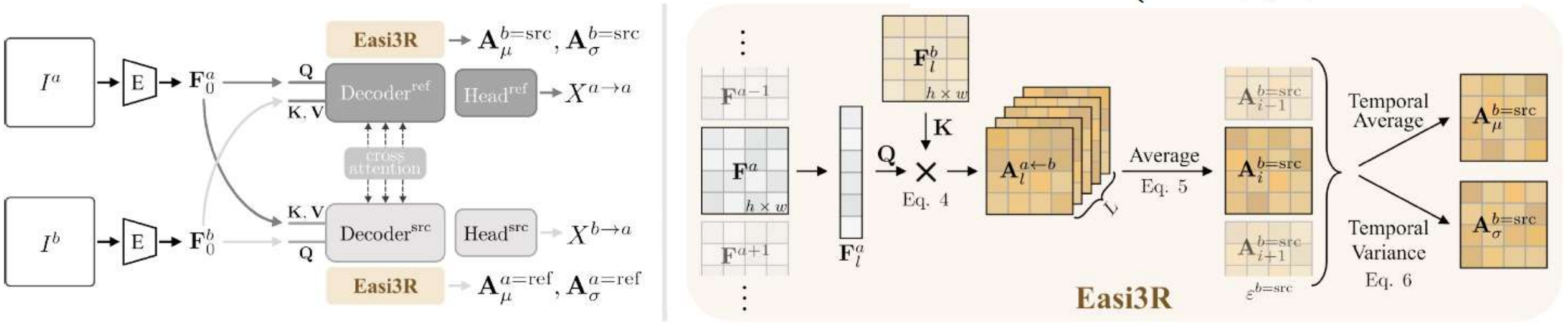


Figure 3. **DUSt3R** and our **Easi3R** adaptation. DUSt3R encodes two images I^a, I^b into feature tokens $\mathbf{F}_0^a, \mathbf{F}_0^b$, which are then decoded into point maps in the reference view coordinate space using two decoders. Our Easi3R aggregates the cross-attention maps from the decoders, producing four semantically meaningful maps: $\mathbf{A}_\mu^{b=\text{src}}, \mathbf{A}_\sigma^{b=\text{src}}, \mathbf{A}_\mu^{a=\text{ref}}, \mathbf{A}_\sigma^{b=\text{ref}}$. These maps are then used for a second inference pass to enhance reconstruction quality.

$$\mathbf{A}_l^{a\leftarrow b} = \mathbf{Q}_l^a \mathbf{K}_l^{b^T} / \sqrt{c}, \quad \mathbf{A}_l^{b\leftarrow a} = \mathbf{Q}_l^b \mathbf{K}_l^{a^T} / \sqrt{c} \tag{4}$$

$$\begin{aligned} \mathbf{A}^{b=\text{src}} &= \sum_l \sum_x \mathbf{A}_l^{a\leftarrow b}(x, y, z)/(L \times h \times w) \\ \mathbf{A}^{a=\text{ref}} &= \sum_l \sum_x \mathbf{A}_l^{b\leftarrow a}(x, y, z)/(L \times h \times w) \end{aligned} \tag{5}$$

$$\mathbf{A}_\mu^{b=\text{src}} = \text{Mean}(\mathbf{A}_i^{b=\text{src}}), \ \mathbf{A}_\sigma^{b=\text{src}} = \text{Std}(\mathbf{A}_i^{b=\text{src}}) \tag{6}$$

$$\begin{aligned} \mathcal{X}^* = \arg\min_{\mathcal{X},\mathbf{P},\mathbf{s}} \sum_{t\in T} \sum_{i\in\varepsilon^t} &\|\mathcal{X}^a - \mathbf{s}_i^t \mathbf{P}_i^t X^{a\to a}\|_1 \\ &+ \|\mathcal{X}^b - \mathbf{s}_i^t \mathbf{P}_i^t X^{b\to a}\|_1 \end{aligned} \tag{2}$$

$$\begin{aligned} \mathcal{L}_{\text{flow}} = \sum_{t\in T} \sum_{i\in\varepsilon^t} &(1 - \mathbf{M}^a) \cdot \|\hat{\mathcal{F}}_i^{a\to b} - \mathcal{F}_i^{a\to b}\|_1 \\ &+ (1 - \mathbf{M}^b) \cdot \|\hat{\mathcal{F}}_i^{b\to a} - \mathcal{F}_i^{b\to a}\|_1 \end{aligned} \tag{11}$$

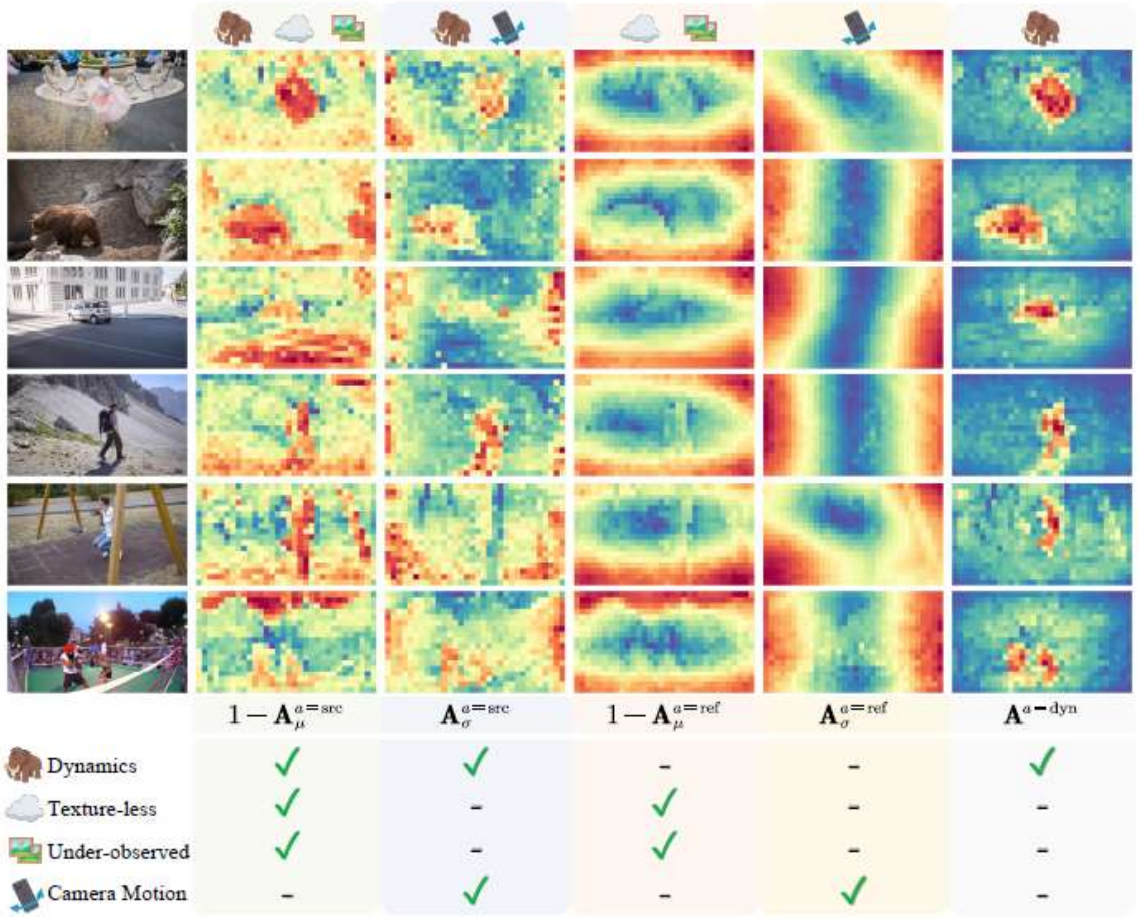


Figure 4. **Visualization for Cross-Attention Maps.** We color the *normalized* values of attention maps, ranging from one to zero. We highlight the patterns captured by each type of attention map using relatively high values. For a more detailed demonstration, we invite reviewers to visit our webpage under easi3r.github.io.

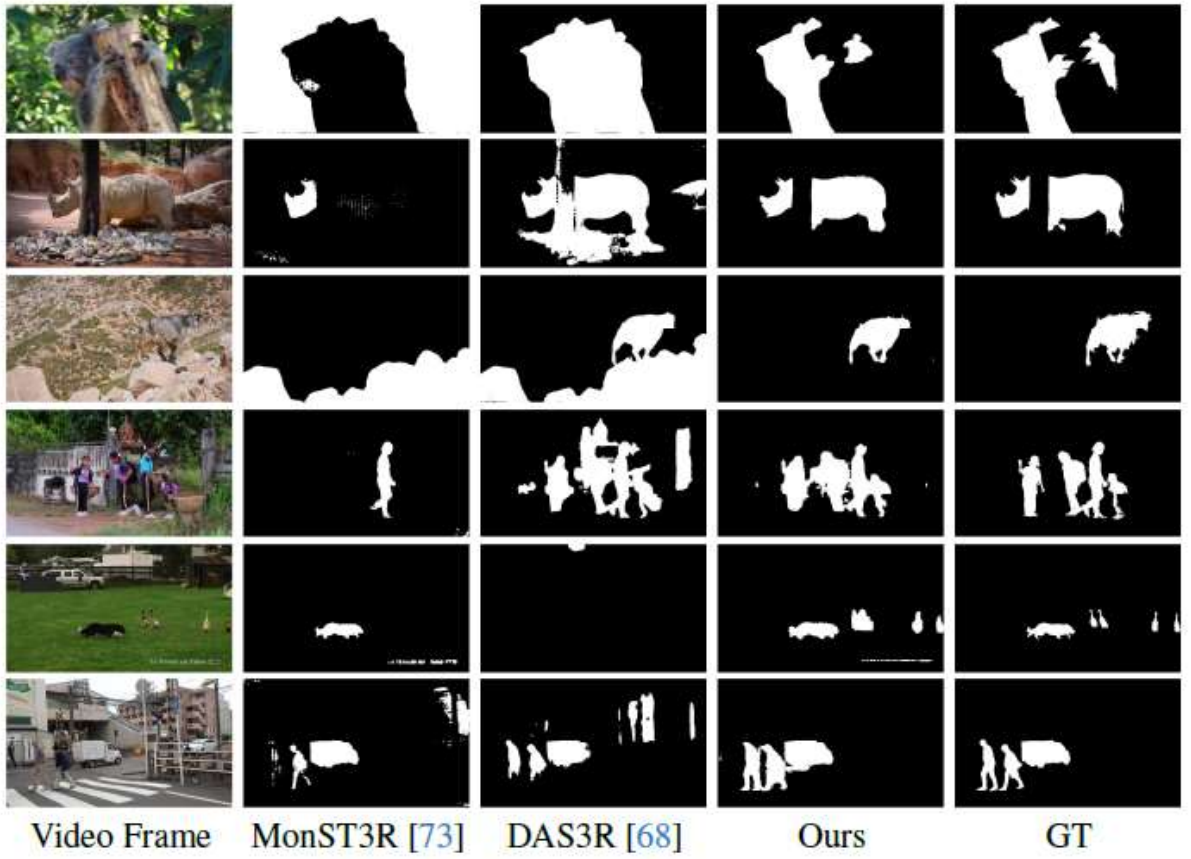
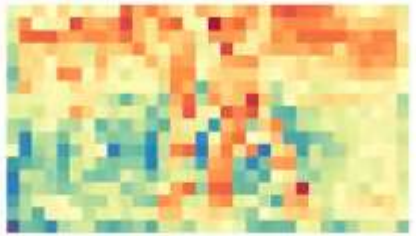
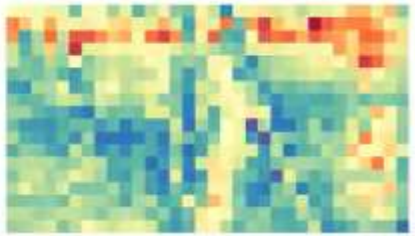


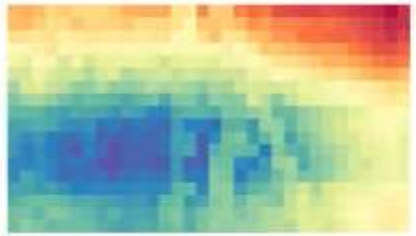
Figure 5. **Qualitative Results of Dynamic Object Segmentation.** "Ours" refers to the Easi3R$_{\text{monst3r}}$ setting. Here, we present the enhanced setting, where outputs from different methods serve as prompts and are used with SAM2 [46] for mask inference.

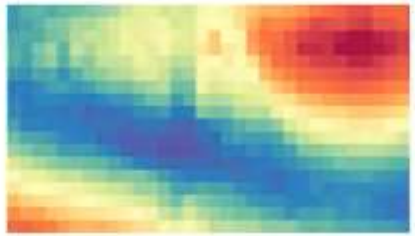
one to zero.

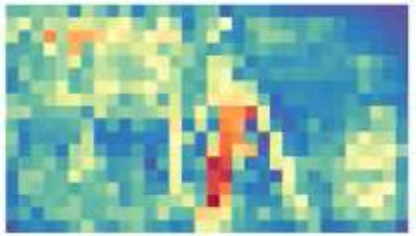
$$\mathbf{A}^{a=\text{dyn}} = (1 - \mathbf{A}_{\mu}^{a=\text{src}}) \cdot \mathbf{A}_{\sigma}^{a=\text{src}} \cdot \mathbf{A}_{\mu}^{a=\text{ref}} \cdot (1 - \mathbf{A}_{\sigma}^{a=\text{ref}}) \quad (9)$$











	$1 - \mathbf{A}_{\mu}^{a=\text{src}}$	$\mathbf{A}_{\sigma}^{a=\text{src}}$	$1 - \mathbf{A}_{\mu}^{a=\text{ref}}$	$\mathbf{A}_{\sigma}^{a=\text{ref}}$	$\mathbf{A}^{a=\text{dyn}}$
Dynamics	✓	✓	-	-	✓
Texture-less	✓	-	✓	-	-
Under-observed	✓	-	✓	-	-
Camera Motion	-	✓	-	✓	-

1.areas with less texture and under-observed areas but also highlights dynamic objects because they violate the rigid body transformation prior

2.highlights both camera and object motion, as the attention of these areas continuously changes over time

3.texture-less regions and under-observed areas less useful for registration

4.pixels perpendicular to the direction of motion generally share similar pixel flow speeds, resulting in consistent deviations that allow us to infer camera motion from the attention pattern.

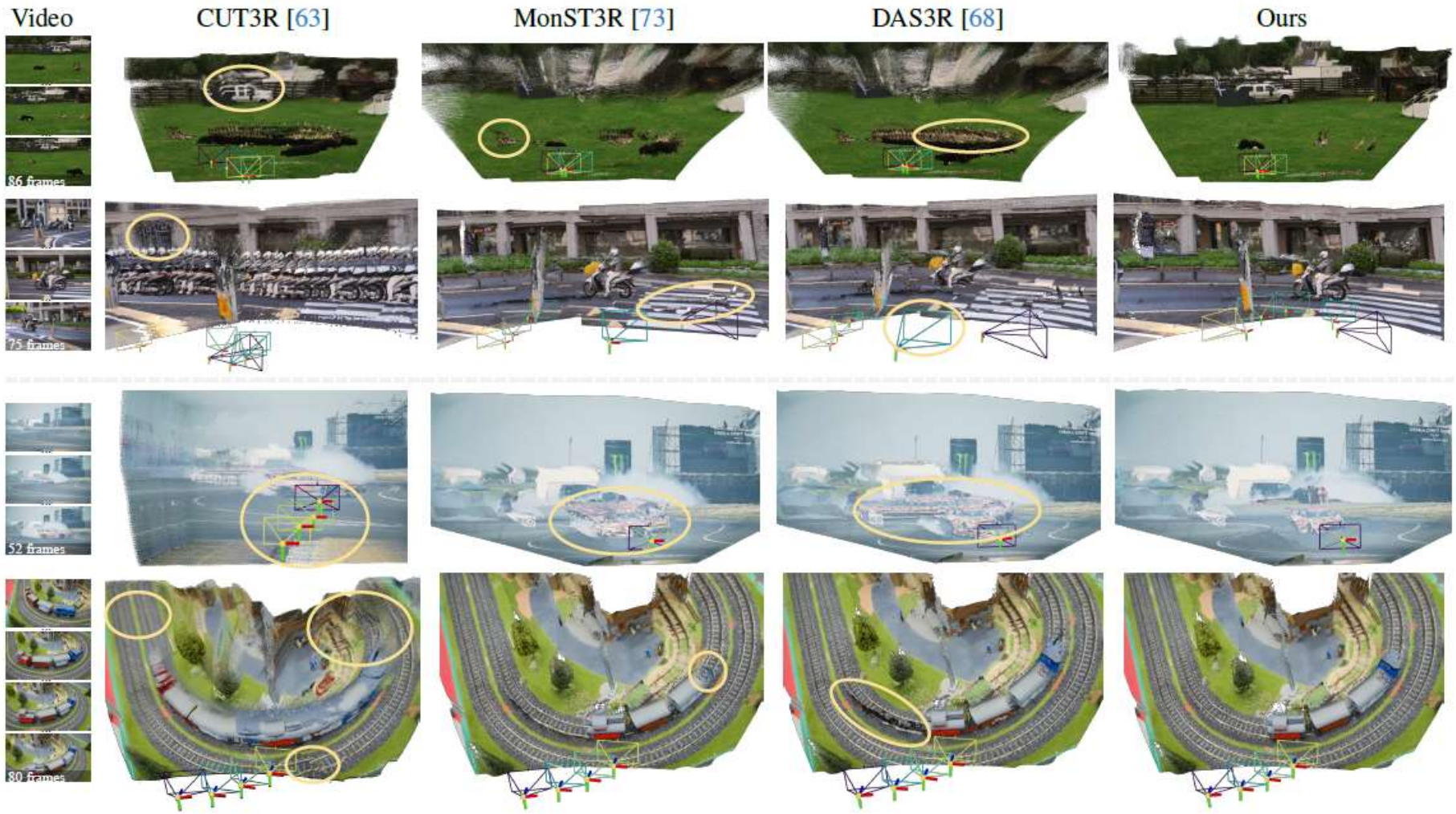


Figure 6. **Qualitative Comparison.** We visualize cross-frame globally aligned static scenes with dynamic point clouds at a selected timestamp. Notably, instead of using ground truth dynamic masks in previous work, we apply the estimated per-frame dynamic masks to filter out dynamic points at other timestamps for comparison. Our method (top two and bottom two rows as Easi3R $_{\text{dust3r/monst3r}}$, respectively) achieves temporally consistent reconstruction of both static scenes and moving objects, whereas baselines suffer from static structure misalignment and unstable camera pose estimation, and ghosting artifacts due to inaccuracy estimation of dynamic segmentation.

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

CVPR 2025 (Oral)

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.

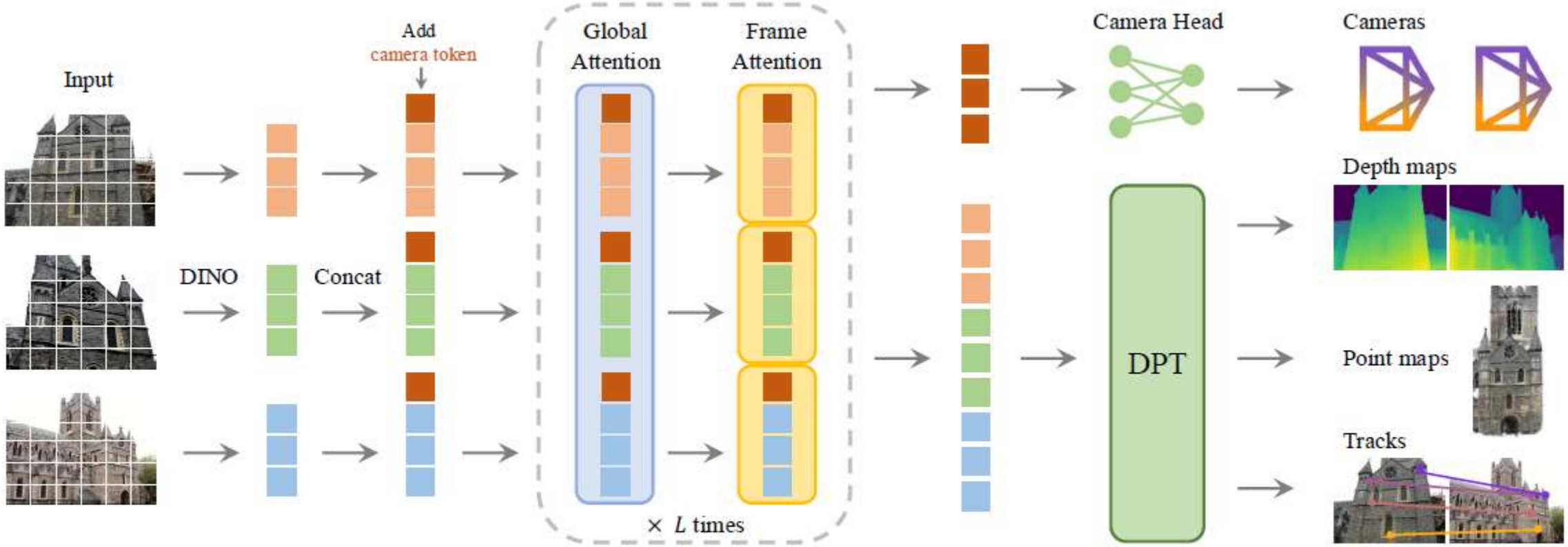


Figure 2. **Architecture Overview.** Our model first patchifies the input images into tokens by DINO, and appends camera tokens for camera prediction. It then alternates between frame-wise and global self attention layers. A camera head makes the final prediction for camera extrinsics and intrinsics,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

多任务联合训练的平衡　高效的Transformer架构设计

大规模数据的训练　　模型的泛化能力

$$\mathcal{L} = \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}} + \lambda \mathcal{L}_{\text{track}}.$$

$$\mathcal{L}_{\text{camera}} = \sum_{i=1}^{N} \|\hat{\mathbf{g}}_i - \mathbf{g}_i\|_\epsilon$$

$$\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} \|\Sigma_i^D \odot (\hat{D}_i - D_i)\| + \|\Sigma_i^D \odot (\nabla \hat{D}_i - \nabla D_i)\| - \alpha \log \Sigma_i^D$$

$$\mathcal{L}_{\text{pmap}} = \sum_{i=1}^{N} \|\Sigma_i^P \odot (\hat{P}_i - P_i)\| + \|\Sigma_i^P \odot (\nabla \hat{P}_i - \nabla P_i)\| - \alpha \log \Sigma_i^P$$

$$\mathcal{L}_{\text{track}} = \sum_{j=1}^{M} \sum_{i=1}^{N} \|\mathbf{y}_{j,i} - \hat{\mathbf{y}}_{j,i}\|.$$

Implementation Details. By default, we employ $L = 24$ layers of global and frame-wise attention, respectively. The model consists of approximately 1.2 billion parameters in total. We train the model by optimizing the training loss (2) with the AdamW optimizer for 160K iterations. We use a cosine learning rate scheduler with a peak learning rate of 0.0002 and a warmup of 8K iterations. For every batch, we randomly sample 2–24 frames from a random training scene. The input frames, depth maps, and point maps are resized to a maximum dimension of 518 pixels. The aspect ratio is randomized between 0.33 and 1.0. We also randomly apply color jittering, Gaussian blur, and grayscale augmentation to the frames. The training runs on 64 A100 GPUs over nine days. We employ gradient norm clipping with a threshold of 1.0 to ensure training stability. We leverage bfloat16 precision and gradient checkpointing to improve GPU memory and computational efficiency.

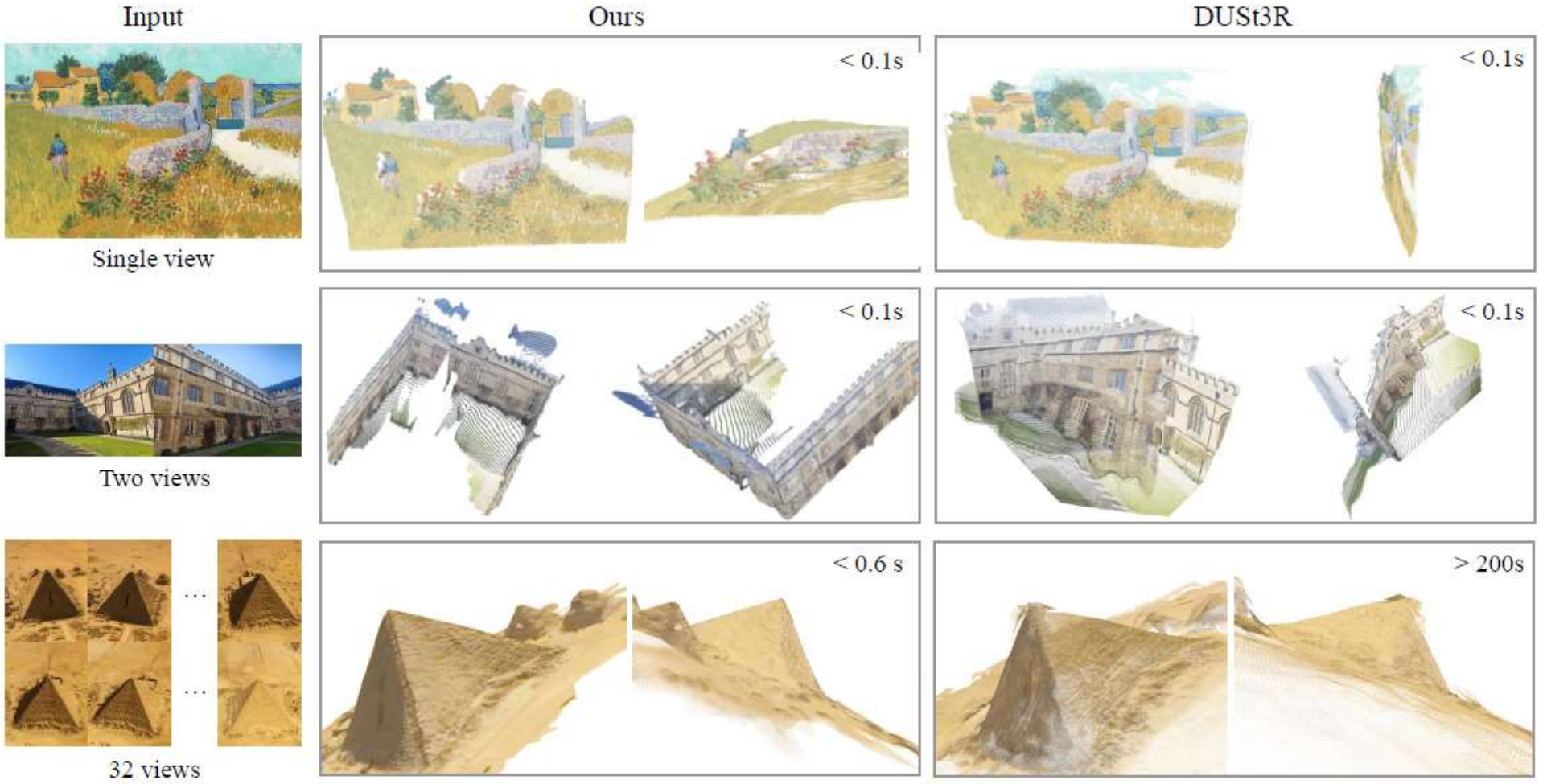


Figure 3. **Qualitative comparison of our predicted 3D points to DUSt3R on in-the-wild images.** As shown in the top row, our method successfully predicts the geometric structure of an oil painting, while DUSt3R predicts a slightly distorted plane. In the second row, our method correctly recovers a 3D 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.

局限性与未来工作

尽管VGGT在多个3D任务上表现出色，但仍存在一些局限性：

•**对极端条件的适应性**：在处理极端输入旋转或非刚性变形较大的场景时，VGGT的性能可能会下降。

•**对特定图像类型的限制**：当前模型不支持鱼眼或全景图像的处理。

•**内存和计算成本**：尽管VGGT的效率较高，但在处理大量输入图像时，内存和计算成本仍然是一个需要考虑的问题。

未来的工作可以集中在以下几个方向：

•**进一步优化架构**：探索更高效的Transformer架构，以降低内存和计算成本。

•**提升泛化能力**：通过在更多样化的数据集上进行训练，提升模型在极端条件下的泛化能力。

•**扩展应用场景**：将VGGT应用于更多的3D任务，如全景图像重建和动态场景重建等。