DAGE: Dual-Stream Architecture for Efficient and Fine-Grained Geometry Estimation

Tuan Duc Ngo[1,†] Jiahui Huang[2] Seoung Wug Oh[2] Kevin Blackburn-Matzen[2]
Evangelos Kalogerakis[1,3] Chuang Gan[1] Joon-Young Lee[2]
[1] Umass Amherst [2] Adobe Research [3] TU Crete
github.com/dage-site

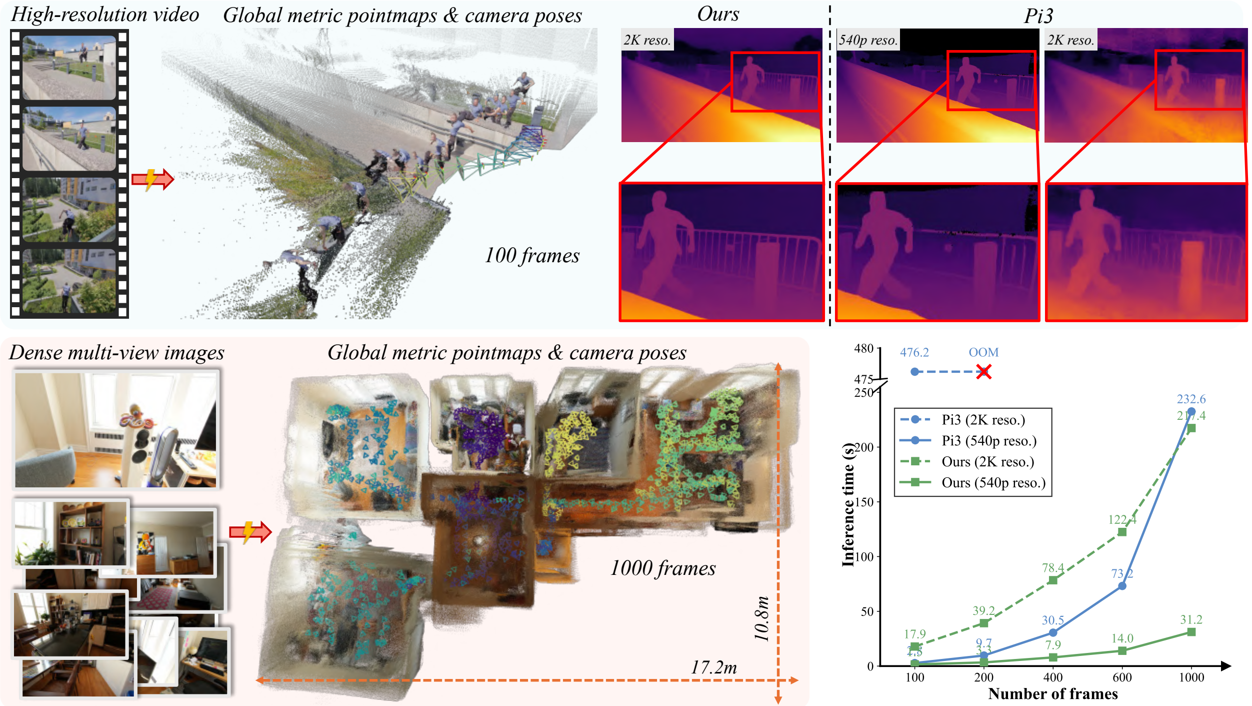


Figure 1. **DAGE** produces *high-resolution, fine-grained, metric-scale* and *cross-view consistent* 3D geometry together with accurate camera poses from visual inputs. It runs substantially faster than prior models [95, 102] and scales to long sequences (up to 1000 frames).

Abstract

*Estimating accurate, view-consistent geometry and camera poses from uncalibrated multi-view/video inputs remains challenging—especially at high spatial resolutions and over long sequences. We present **DAGE**, a dual-stream transformer whose main novelty is to disentangle global coherence from fine detail. A **low-resolution** stream operates on aggressively downsampled frames with alternating frame/global attention to build a view-consistent representation and estimate cameras efficiently, while a **high-resolution** stream processes the original images per-frame to preserve sharp boundaries and small structures. A lightweight **adapter** fuses these streams via cross-attention, injecting global context without disturbing the pretrained single-frame pathway. This design scales resolution and clip length* independently, *supports inputs up to 2K, and maintains practical inference cost. DAGE delivers sharp depth/pointmaps, strong cross-view consistency, and accurate poses, establishing new state-of-the-art results for video geometry estimation and multi-view reconstruction.*

1. Introduction

Estimating 3D geometry and camera poses from multi-view images is a fundamental problem in computer vision. We target the demanding regime of *uncalibrated, high-resolution* inputs with potentially *thousands of frames*. This task is particularly challenging, as the model must simultaneously (i) enforce global consistency across views, (ii)

[†] work done during internship at Adobe Research.

preserve fine-grained details at high resolution, and (iii) remain tractable in runtime and memory for long sequences.

On one hand, feed-forward visual geometry networks [48, 89, 95, 97, 102, 109, 119] have achieved remarkable progress in globally consistent multi-view geometry estimation, setting new state-of-the-art results on various benchmarks [18, 88], including video depth estimation, 3D reconstruction, and camera pose prediction. However, their typically heavy network architectures limit training and inference to modest image resolutions (e.g., long side $\leq$ 518px) and a small number of input views, which leads to blurred thin structures and poorly defined object boundaries. Several works have adopted post-training acceleration strategies [28, 80, 94] to reduce computational cost and support more views during inference, yet they do not address the loss of high-frequency details or the tendency toward oversmoothed surfaces near edges and small objects.

On the other hand, single-view geometry estimators [8, 98, 99, 110, 111] operate flexibly at high resolution and produce sharp, detail-rich depth/pointmaps from single images, yet they lack temporal and multi-view consistency by design. Attempts to adapt these models to handle videos [16, 46, 49, 53, 57, 108] introduce heavy pipelines, and typically do not recover accurate camera poses. As a result, they fail to assemble a globally consistent 3D scene geometry directly from the feed-forward predictions.

Based on this observation, we present DAGE, a **D**ual-stream **A**rchitecture for efficient and fine-grained **G**eometry **E**stimation that meets the above criteria. It comprises two parallel streams and a lightweight fusion adapter. The **Low-Resolution (LR) Stream** focuses on extracting globally consistent features and predicting camera poses. It is composed of a ViT backbone followed by a global transformer with alternating frame-global attention [95, 102], which processes the entire sequence at a lower spatial resolution. Although the global transformer is computationally intensive, operating at low resolution keeps it tractable while preserving global context. The **High-Resolution (HR) Stream** is designed to capture high-frequency details and fine-grained features. It employs a ViT [24] that processes each image independently at its native resolution. Finally, our proposed **Lightweight Adapter** synchronizes and fuses LR and HR tokens before the dense heads, yielding geometry that is both globally consistent and richly detailed.

This decoupled design grants two critical advantages. *First, it achieves global consistency and tractability.* By restricting the computationally-heavy global attention to the LR stream, we alleviate the quadratic scaling bottleneck of global transformers [95, 102]. This significantly reduces runtime, by 2× and 28× at 540p and 2K resolutions, respectively, enabling our model to process thousands of frames. *Second, it preserves high-fidelity detail.* The HR stream operates per-frame, allowing it to scale to any resolution (e.g., up to 2K) and leverage priors from state-of-the-art single-image models for sharp detail and strong real-world generalization. In contrast to standard pipelines [95, 102, 109, 119] that couple image resolution with sequence length, DAGE decouples the two, enabling independent control over spatial detail and multi-view coherence, with a tractable runtime (see Fig. 1).

We validate our method and design choices through extensive experiments. DAGE achieves state-of-the-art performance on video geometry and depth-sharpness benchmarks, and is competitive on 3D reconstruction and camera pose estimation—while offering higher throughput and a lower GPU memory footprint. In summary, our technical contributions are twofold:

- A *dual-stream transformer* that couples a per-frame, high-resolution detail path with a multi-view, low-resolution global-attention path.
- A lightweight *Adapter* that fuses the two streams to produce sharp yet globally consistent geometry.

2. Related Work

Single-view Geometry Estimation aims to recover 3D scene geometry from a single image. Early approaches relied on handcrafted features and probabilistic models [39, 45, 76, 77]. With deep networks, numerous architectures were proposed [4, 26, 29, 54, 93], yet their generalization remained limited. The introduction of relative-depth [72] enabled training on large, mixed datasets, leading to strong zero-shot performance [5, 8, 36, 40, 70, 110, 111, 117]. However, many such methods require camera intrinsics and metric scale to recover absolute 3D geometry. Recent work addresses this by jointly estimating depth and intrinsics [8, 70, 116]. A complementary line of research regresses dense *3D pointmaps* directly [98, 99], from which depths and intrinsics can be recovered. Despite impressive single-image results, these methods typically exhibit temporal jitter and inconsistent scale when applied to videos.

Fine-Grained Geometry Estimation targets predicting sharper depth/pointmaps with high-frequency detail. Patch-wise fusion methods [59, 65] enhance local detail by combining per-patch estimates, but often introduce stitching artifacts at patch boundaries. Another line of work leverages powerful generative priors [71, 75] to produce highly detailed depth [30, 31, 37, 47, 69]. Depth Anything V2 [111] improves detail via large-scale, high-quality synthetic data, while DepthPro [8] employs a multi-patch ViT design [24] to better capture fine structures. MoGe2 [99] combines synthetic and refined real-world annotations with a coarse-to-fine loss [98], achieving strong metric accuracy and sharp predictions. *Concurrently*, [107] integrates foundation-model geometry priors [95, 111] with a cascaded DiT [68], yielding *pixel-perfect depth*. Nonetheless,

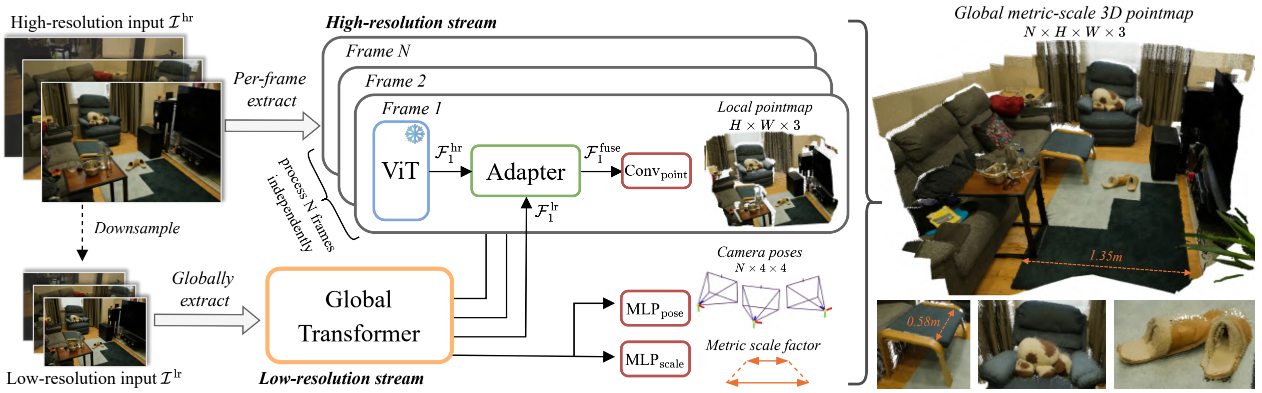

High-resolution input $\mathcal{I}^{\text{hr}}$
High-resolution stream
Frame N
Frame 2
Frame 1
Per-frame extract
ViT
$\mathcal{F}_1^{\text{hr}}$
Adapter
$\mathcal{F}_1^{\text{fuse}}$
$\text{Conv}_{\text{point}}$
Local pointmap
$H \times W \times 3$
process N frames independently
$\mathcal{F}_1^{\text{lr}}$
Downsample
Globally extract
Global Transformer
Camera poses
$N \times 4 \times 4$
MLP_{pose}
$\text{MLP}_{\text{scale}}$
Metric scale factor
Low-resolution input $\mathcal{I}^{\text{lr}}$
Low-resolution stream
Global metric-scale 3D pointmap
$N \times H \times W \times 3$
1.35m
0.58m

Figure 2. **Overview of DAGE.** Given a set of *unposed* RGB images, the model predicts per-frame pointmaps and camera poses, plus a scene-wise metric scale. The architecture has two parallel streams: (i) a low-resolution stream (lower part) that processes downsampled inputs to aggregate global context and regress poses/scene scale; and (ii) a high-resolution stream (upper part) that processes frames independently at native resolution to preserve fine detail. A lightweight Adapter fuses LR and HR tokens before the dense geometry head.

these methods are predominantly per-image and fail to ensure temporal consistency in video setting.

Video-based Geometry Estimation mitigates temporal jitter and scale inconsistency by either stabilize *per-frame* predictions with test-time procedures, or leverage video architectures. Several works regularize single-image depth across time using geometric consistency or online refinement [51, 52, 62, 90], or optimize scale/shift to co-align frames [46]. [13, 17, 53] add temporal heads or video transformers on top of pretrained single-view models, and diffusion-based pipelines [7, 38, 112] leverage strong video priors for temporally consistent depth. Despite impressive performance, diffusion-based methods are compute-intensive and typically do not recover camera poses.

Visual Geometry Estimation regresses both camera poses and 3D scene structure from uncalibrated images or video. Classical SfM/MVS pipelines [1, 78, 83, 104], are robust but require complex, multi-stage optimization. [14, 66, 113] inject motion priors (e.g., optical flow, point tracking) and then perform bundle adjustment, which reduces manual engineering but requires per-video optimization. Dust3R [100] introduced a learning-based alternative that predicts pointmaps for image pairs in a shared coordinate frame and stitches multi-view inputs via a global alignment step. Subsequent work improves metric-scale recovery [55], extends to dynamic scenes [15, 27, 61, 118], and scales multi-view processing [48, 89, 95, 97, 102, 105, 109, 119, 121]. Among these, VGGT [95] and Pi3 [102] demonstrate state-of-the-art performance with an alternating global-frame attention transformers. However, the quadratic cost of global attention imposes tight token budgets, limiting input resolution and the number of frames; consequently, predicted depth often appears blurred and fine structures are smoothed.

In contrast, our dual-stream approach performs feed-forward global aggregation on LR inputs for efficiency while preserving HR detail via a per-frame stream, with a lightweight adapter to fuse these two.

3. Method

3.1. Problem Definition

Given an *uncalibrated* set of N RGB images $\mathcal{I} = \{I_i\}_{i=1}^{N}$ of a scene, where each $I_i \in \mathbb{R}^{H \times W \times 3}$, our model aims to reconstruct the 3D scene geometry by predicting three components: (1) per-frame pointmaps $\mathcal{P} = \{P_i\}_{i=1}^{N}$, where $P_i \in \mathbb{R}^{H \times W \times 3}$ represents the 3D coordinates of each pixel in the local camera coordinate system; (2) camera poses $\mathcal{G} = \{G_i\}_{i=1}^{N}$, where $G_i \in \text{SE}(3)$ encodes each camera's rotation and translation; and (3) a single global metric scale factor $s \in \mathbb{R}^{+}$.

State-of-the-art feed-forward approaches [95, 102] are constrained by the high computational cost of global attention. This typically limits their inputs to modest resolutions (e.g., 518px on the long side) and short sequences (e.g., $N < 200$). We address this limitation with a dual-stream architecture designed to produce high-quality, fine-grained 3D geometry and accurate camera poses while supporting flexible spatial resolutions and long sequences.

Fig. 2 illustrates the overall architecture of our model. It consists of a *low-resolution (LR) stream* (Sec.3.2) and a *high-resolution (HR) stream* (Sec.3.3), which operate in parallel and are synchronized through a lightweight adapter (Sec.3.4), followed by dense prediction heads (Sec.3.5). The LR stream extracts globally consistent features and estimates camera poses, while the HR stream predicts per-frame pointmaps at the native input resolution. The global features produced by the LR stream are injected into the HR stream via the adapter to enhance geometric consistency across views. Training details are described in Sec.3.6.

3.2. Low-Resolution Stream

The low-resolution stream is responsible for enforcing global consistency and estimating camera poses. It pro-

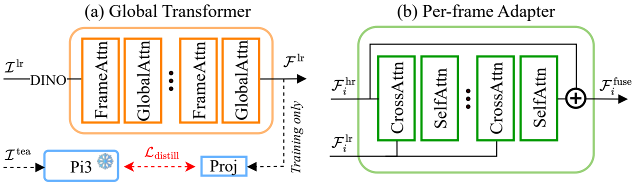

(a) Global Transformer
$\mathcal{I}^{\text{lr}}$
DINO
FrameAttn
GlobalAttn
FrameAttn
GlobalAttn
$\mathcal{F}^{\text{lr}}$
Training only
$\mathcal{I}^{\text{tea}}$
Pi3
$\mathcal{L}_{\text{distill}}$
Proj
(b) Per-frame Adapter
$\mathcal{F}_i^{\text{hr}}$
$\mathcal{F}_i^{\text{lr}}$
CrossAttn
SelfAttn
CrossAttn
SelfAttn
$\mathcal{F}_i^{\text{fuse}}$

Figure 3. The **Global transformer** (left) operates on low-resolution inputs with alternating global and frame-wise attention; during training, feature distillation compensates for aggressive downsampling. The **Adapter** (right) stacks cross and self-attention blocks to fuse multi-view–consistent LR tokens into the HR stream.

cesses the entire sequence $\{I_i\}$ at a fixed low resolution (long side $\leq$ 252px), $\mathcal{I}^{\text{lr}}$. These images are passed through a global transformer to output *LR feature tokens* $\mathcal{F}^{\text{lr}} \in \mathbb{R}^{N \times h_{lr} \times w_{lr} \times C}$. This global transformer (Fig. 3a) consists of a DINOv2 [67] tokenizer, and alternating blocks of frame-wise and global self-attention (`[FrameAttn` $\rightarrow$ `GlobalAttn]`) [95, 102], which is effective for capturing scene-level structure. We do not use dedicated camera tokens, preserving permutation equivariance [102].

While low-resolution processing ensures tractability, training the LR stream from scratch often leads to degraded camera pose accuracy. To address this, we leverage the rich global representations of a pre-trained teacher model, Pi3 [102], through knowledge distillation. Specifically, the teacher processes a higher-resolution input $\mathcal{I}^{\text{tea}}$ (capped at 518px to match [95, 102]) and produces features $\mathcal{F}^{\text{tea}} \in \mathbb{R}^{N \times h_{\text{tea}} \times w_{\text{tea}} \times C}$. These features are then used to supervise the LR stream via a feature distillation loss:

$$\mathcal{L}_{\text{dis}} = 1 - \text{sim}(p_\phi(\mathcal{F}^{\text{lr}}), \mathcal{F}^{\text{tea}}) \tag{1}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity function, and p_ϕ is a projection network mapping student features to the teacher's representation space and spatial dimension.

3.3. High-Resolution Stream

The high-resolution stream processes each frame of the input sequence $\{I_i\}$ independently at its original resolution. To preserve fine-grained detail and strong zero-shot generalization capabilities, we adopt MoGe2 [99] as the HR backbone. This model uses a 24-layer ViT encoder [24] to extract the *HR feature tokens* $\mathcal{F}^{\text{hr}} \in \mathbb{R}^{N \times h_{hr} \times w_{hr} \times C_{hr}}$.

3.4. Adapter

The adapter (Fig. 3b) is designed to inject global context from the LR stream into the per-frame HR stream. A naive solution—such as upsampling LR features via interpolation and concatenating them with HR features—often introduces artifacts and fails to capture meaningful cross-view relations. Alternative approaches, including pixel-shuffle upsampling [19, 107] or CNN-based upsampling [87], alleviate such artifacts but rely on a fixed scale factor, which is too restrictive for inputs that may vary up to 2K resolution.

To overcome these limitations, we adopt a more flexible *cross-attention* mechanism that accommodates arbitrary token counts from both streams. This fusion is followed by HR self-attention to restore per-frame spatial coherence. Concretely, for each i-th frame, the fused feature $\mathcal{F}_i^{\text{fuse}}$ is computed as:

$$\mathcal{F}_i^{\text{fuse}} = \text{CrossAttn}\big(Q = \mathcal{F}_i^{\text{hr}};\, K, V = \mathcal{F}_i^{\text{lr}}\big) \tag{2}$$

$$\mathcal{F}_i^{\text{fuse}} = \mathcal{F}_i^{\text{hr}} + \text{MLP}\big(\text{SelfAttn}(\mathcal{F}_i^{\text{fuse}})\big) \tag{3}$$

Positional encodings are applied before attention to align HR patch coordinates with their LR counterparts, stabilizing the cross-scale fusion. We stack five such `[CrossAttn` $\rightarrow$ `SelfAttn]` blocks.

We employ Rotary Positional Encodings (RoPE) for all attention layers, but with different strategies for self-attention and cross-attention to handle varying resolutions. **Self-Attention:** Standard RoPE does not extrapolate well to spatial dimensions larger than those seen during training, which can cause distortion on high-resolution inputs [95, 97, 102]. Thus, we adopt the *interpolated RoPE* [12] technique. We define a fixed maximum patch length, $l^{\max}$. At both training and inference, we rescale the angular frequencies of the positional encoding to this fixed context size, which keeps the positional spectrum stable even at very high resolutions. **Cross-Attention:** A challenge is the large spatial mismatch between the LR and HR streams (e.g., 252px vs. 2K). To align them, we "snap" each HR token to its nearest grid cell in the LR feature map. The HR token then uses the positional encoding from that corresponding LR cell. This simple strategy effectively matches patches across scales and avoids extrapolation, as the LR stream's spatial dimensions are always fixed and within the trained bounds. Concretely, let $R(\mathbf{f}, \mathbf{m})$ be the RoPE function applied to token $\mathbf{f}$ at 2D position $\mathbf{m}$. The modified RoPE functions are:

$$R_{\text{self}}\big(\mathbf{f}^{\text{hr}}, \mathbf{m}^{\text{hr}}\big) = R\big(\mathbf{f}^{\text{hr}},\, \mathbf{m}^{\text{hr}} \cdot l^{\max}/l^{\text{hr}}\big)\,, \tag{4}$$

$$R_{\text{cross}}\big(\mathbf{f}^{\text{hr}}, \mathbf{m}^{\text{hr}}\big) = R\Big(\mathbf{f}^{\text{hr}},\, \text{sampling}\Big(\mathbf{m}^{\text{hr}},\, \text{grid}^{\text{lr}}\Big)\Big) \tag{5}$$

where l^{hr} is the side length of the HR grid.

Adapter Design Discussion. We investigated various strategies for fusing the LR and HR tokens, focusing on *where* and *how* to inject the global information. For *where* to insert, one approach is to inject intermediate LR features into each ViT layer of the HR stream. This mitigates scale drift but fails to enforce cross-view global consistency (see Sec. 4.6). For *how* to fuse, we considered alternatives like concatenation and addition with learnable interpolation. We find that the best trade-off is a lightweight adapter after the HR ViT encoder, comprising cross-attention to inject global context and self-attention to re-calibrate intra-frame coherence (see Sec. 4.6). This strategy preserves the HR stream's original feature space at the start of training, allowing the model to gradually learn to incorporate the multi-view consistent constraints to refine the final geometry.

3.5. Prediction Heads

Dense Geometry. We employ a feature pyramid of convolutional layers [99] to gradually upsample the per-patch features $\mathcal{F}^{\text{fuse}}$ into dense feature maps at the original image resolution to regress the pointmaps $\mathcal{P}$. This convolutional-style head yields smoother predictions, avoiding the grid-like artifacts observed in [102] (see Fig. 5).

Camera Pose. We regress the per-frame camera parameters using the LR features $\mathcal{F}^{\text{lr}}$. This is done for efficiency, as camera poses do not require fine-grained features. Following [23, 102], we use average pooling and an MLP to regress the translation and rotation in a 9D representation [56].

Metric Scale. We add a *metric scale* token in the video transformer, followed by an MLP to predict a single metric scale factor for each scene.

3.6. Training Details

3.6.1. Training loss

We train the model with a combination of pointmap, camera, scale, normal, gradient, and distillation losses.

Pointmap loss. We predict a per-pixel 3D point $\hat{\mathbf{p}}_{i,j}$ up to a scene-wide scale. Let $\mathrm{norm}(\cdot)$ denote a scene normalization (distance-to-origin) applied to both prediction and ground truth. We compute a single alignment scale s^* using the ROE solver [98] and supervise with an ℓ_1 loss:

$$\mathcal{L}_{\text{pm}} = \frac{1}{NHW} \sum_{i=1}^{N} \sum_{j=1}^{H \times W} \left\| \frac{s^* \hat{\mathbf{p}}_{i,j}}{\mathrm{norm}(\hat{\mathcal{P}})} - \frac{\mathbf{p}_{i,j}}{\mathrm{norm}(\mathcal{P})} \right\|_1 . \quad (6)$$

Unlike uncertainty-weighted objectives used in [95, 100], we do *not* attenuate errors with confidences, as we found this can suppress hard structures and reduce sharpness.

Camera loss. Following [102], we supervise *relative* camera poses to avoid fixing a global coordinate frame. Let $\hat{\mathbf{g}}_{uv}$ and $\mathbf{g}_{uv}$ denote predicted and ground-truth pairwise poses between frames u-th and v-th, the camera loss is defined as:

$$\mathcal{L}_{\text{camera}} = \frac{1}{N(N-1)} \sum_{\substack{u,v=1 \\ u \neq v}}^{N} \mathcal{L}_{\text{cam}}\big(\hat{\mathbf{g}}_{uv}, \mathbf{g}_{uv}\big), \quad (7)$$

where $\mathcal{L}_{\text{cam}}$ comprises $\mathcal{L}_{\text{rot}}$ that minimizes the geodesic distance of the rotation part, and $\mathcal{L}_{\text{trans}}$ is the ℓ_1 loss of the translation part.

Scale loss. For datasets with metric supervision, we additionally supervise the predicted metric scale $\hat{s}$:

$$\mathcal{L}_{\text{scale}} = \left\| \log \hat{s} \; - \; \log \left(s^* \frac{\mathrm{norm}(\mathcal{P})}{\mathrm{norm}(\hat{\mathcal{P}})} \right) \right\|_2 . \quad (8)$$

Normal loss. To encourage locally smooth but faithful surfaces, we supervise normals computed *on-the-fly* from the pointmap via cross products [98]:

$$\mathcal{L}_{\text{normal}} = \frac{1}{NHW} \sum_{i=1}^{N} \sum_{j=1}^{H \times W} \angle\big(\hat{\mathbf{n}}_{i,j}, \mathbf{n}_{i,j}\big), \quad (9)$$

where $\angle(\cdot, \cdot)$ is the angular difference.

Gradient loss. To improve local geometry, MoGe [98] applies a *multi-scale affine-invariant pointmap loss* by sub-sampling local regions at several scales and aligning each region to ground truth independently. While this improves single-image sharpness, we found that per-region independent alignments introduce patch-wise degrees of freedom that break cross-view consistency, leading to seams and drift—undesirable in our multi-view setting (see Tab. 6b). Instead, we preserve a *single global alignment* and encourage detail by supervising *image gradients* of the canonical inverse depth [8] at multiple scales:

$$\mathcal{L}_{\text{gradient}} = \frac{1}{NHW} \sum_{i=1}^{N} \sum_{j=1}^{H \times W} \left\| \nabla \hat{d}_{i,j} - \nabla d_{i,j} \right\|_1 , \quad (10)$$

where $\hat{d}$ and d denote ground-truth and predicted canonical inverse depth, respectively, and ∇ is the Scharr and Laplace gradient filters.

Due to the sparsity of real-world depth annotations, we only apply the normal and gradient loss on synthetic data.

3.6.2. Implementation details

The HR stream uses a frozen 24-layer ViT from MoGe2 [99]. Since our training corpus is relatively small, we initialize the LR stream from Pi3 [102] instead of training from scratch. The adapter comprises five blocks, each containing a cross-attention and self-attention layer. We train DAGE on 18 publicly available datasets spanning indoor/outdoor, static/dynamic scenes. The complete list of datasets and implementation details are provided in the supplementary.

4. Experiments

This section compares our method to state-of-the-art approaches across four tasks to show its effectiveness.

4.1. Video Geometry Estimation

We evaluate on eight datasets spanning diverse conditions—GMU Kitchens [33], ScanNet [20] (indoor RGB-D), KITTI [32] (outdoor driving with LiDAR), Sintel [9] and Monkaa [63] (synthetic with precise depth and challenging dynamics), and the high-resolution UrbanSyn [34], Unreal4K [91], and Diode [92]-and resolutions from ∼640p to 2K. Following [98, 108], we report relative point error $\text{Rel}^p \downarrow$ and inlier ratio $\delta^p \uparrow$ at a 0.25 threshold, and evaluate *affine-invariant* pointmaps by aligning predictions to ground truth with a single, shared scale and shift per video. We compare against single-image methods [8, 98, 99], video diffusion-based model [108], and set-based visual-geometry models [95, 97, 102]. For methods that do not

Table 1. **Video pointmap evaluation**. Results are aligned with the ground truth by optimizing a shared scale and shift factor across the entire video. "MV/HR/PO" indicate multi-view support, high-resolution input support, and whether the method predicts camera poses. We mark best and second-best.

Method	MV	HR	PO	**GMU** [33]		**Monkaa** [63]		**Sintel** [9]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Unreal4K** [91]		**Diode** [92]		**Rank** ↓
				(960 × 512)		(960 × 512)		(896 × 448)		(640 × 512)		(768 × 384)		(2048 × 1024)		(1920 × 1080)		(1024 × 768)		
				Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	Rel^p ↓	δ^p ↑	
DepthPro [8]		✓		9.5	93.9	25.1	58.4	40.8	44.7	9.3	94.9	10.0	94.9	48.9	40.1	74.7	12.0	32.4	59.2	7.9
MoGe [98]		✓		20.3	71.2	22.9	61.3	29.4	59.8	13.4	88.0	8.0	95.8	14.9	87.0	38.3	51.5	31.8	52.9	7.4
MoGe2 [99]		✓		19.6	72.4	25.0	57.0	29.8	58.4	12.4	89.4	9.0	97.2	13.4	90.0	32.9	59.1	31.0	54.2	6.8
MoGe2 [99]†	△	✓		7.1	94.6	21.4	67.6	28.2	62.8	7.8	97.5	10.5	98.4	7.2	97.1	12.6	86.7	15.8	84.1	4.1
CUT3R [97]	✓		✓	8.0	93.7	31.8	47.5	35.8	47.5	5.9	97.9	14.5	87.5	21.6	68.6	16.8	79.6	17.9	78.3	6.8
VGGT [95]	✓		✓	5.4	93.8	13.6	84.4	23.7	73.1	2.9	99.0	7.5	97.4	14.5	87.3	8.6	96.1	13.3	85.9	3.4
Pi3 [102]	✓		✓	5.2	94.2	11.6	90.0	22.0	72.9	2.2	99.4	6.3	97.3	16.8	77.5	19.5	75.3	9.2	95.2	3.3
GeoCrafter [108]	✓	△		8.3	94.8	15.7	83.4	25.0	69.3	8.3	96.9	5.6	98.8	12.5	91.9	21.6	74.5	12.5	93.0	3.9
DAGE (ours)	✓	✓	✓	4.9	94.2	10.1	91.0	21.5	75.6	2.1	99.5	5.9	99.0	8.8	96.0	11.9	89.1	9.7	94.4	1.6

△: *partial support*. †: obtained by multiplying per-frame pointmap by predicted metric scale factor.

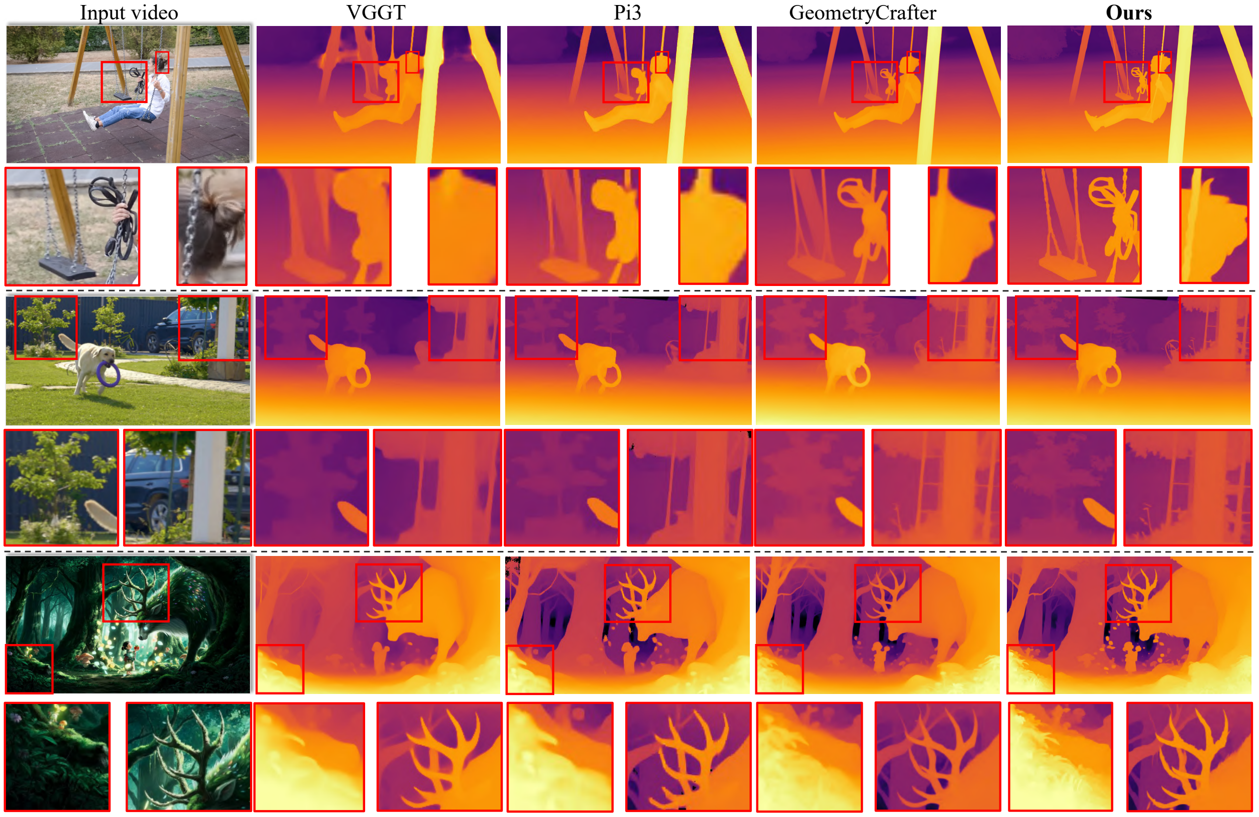


Figure 4. **Visual comparison of video depth on *in-the-wild* scenes.** We convert the depth map to a disparity map for better visualization, and zoom-in (red bounding boxes) to emphasize details. DAGE preserves sharp boundaries and fine-grained detail—especially for thin structures and small or distant objects, outperforming a diffusion-based baseline [108].

support high-resolution inference [95, 97, 102, 108], inputs are downsampled to the model's native resolution and outputs are upsampled to the original size to avoid degenerate predictions. In Tab. 1, DAGE delivers consistently strong performance across datasets—achieving state-of-the-art average rank on Rel^p and δ^p—with pronounced gains on high-resolution scenarios. We visualize the disparity map of our approach and other baselines in Fig. 4. Detailed evaluations (including *scale-invariant*, *affine-invariant* and video depth estimation) are provided in the supplementary.

4.2. Sharp Depth Estimation

We assess depth–boundary sharpness on four synthetic datasets with high-quality depth annotations —Monkaa [63], Sintel [9], UrbanSyn [34], and Unreal4K [91]. Following [8], we report the scale-invariant

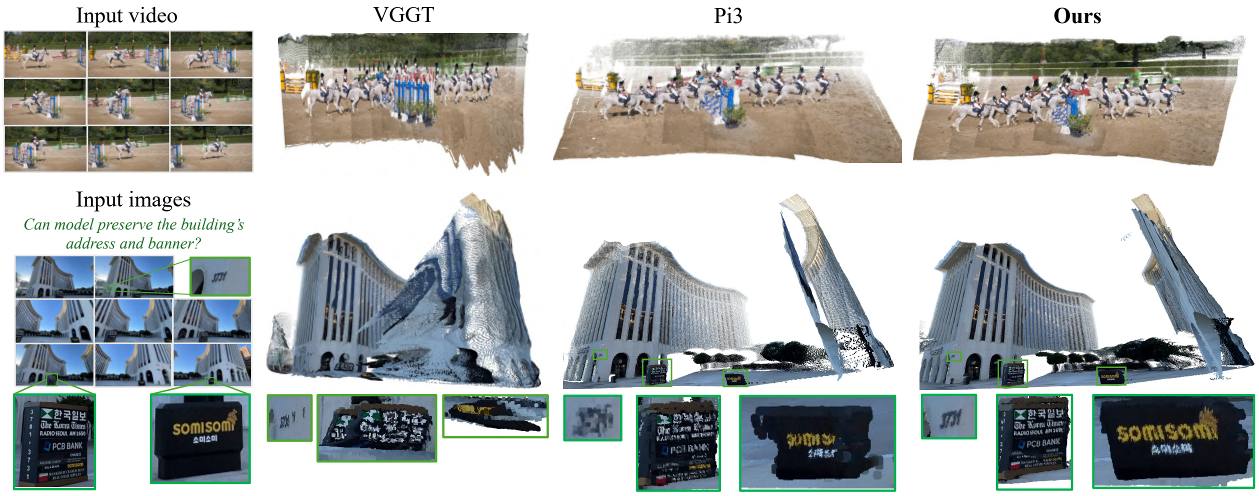


Figure 5. **Visual comparison of 3D reconstruction on *in-the-wild* scenes.** Compared to VGGT [95] and Pi3 [102], DAGE achieves comparable multi-view consistency while preserving markedly finer detail (green boxes).

Table 2. **Sharpness depth evaluation**.

Method	**Monkaa** [63]		**Sintel** [9]		**UrbanSyn** [34]		**Unreal4K** [91]	
	F1↑	C_{PDBE} ↓	F1↑	C_{PDBE} ↓	F1↑	C_{PDBE} ↓	F1↑	C_{PDBE} ↓
DepthPro [8]	0.19	21.3	0.41	17.0	0.14	12.5	0.07	116.4
MoGe2 [99]	0.27	11.6	0.27	10.1	0.09	19.1	0.10	35.2
GeoCrafter [108]	0.19	8.86	0.28	8.1	0.08*	33.2*	0.06*	41.4*
CUT3R [97]	0.08	20.3	0.11	16.5	0.01	44.0	0.01	63.1
VGGT [95]	0.14	11.1	0.20	9.6	0.02*	42.0*	0.03*	38.1*
Pi3 [102]	0.14	12.7	0.20	8.1	0.01	27.9	0.03	46.9
DAGE (ours)	0.29	10.1	0.34	7.8	0.09	17.8	0.14	33.1

*: *Input resolution downscaled to prevent out-of-memory (OOM).*

boundary F1↑, which compares occlusion contours induced by neighboring-pixel depth ratios in prediction versus ground truth. Because F1 does not reflect temporal stability, we also measure the pseudo depth boundary error (C_{PDBE} ↓) [69], defined as the Chamfer distance between prediction and ground-truth at canny-detected edges. For a fair comparison of sharpness detail, we evaluate methods at each dataset's native resolution; models that run out of memory (e.g., [95, 108]) are downsampled to the largest feasible resolution. Results in Tab. 2 show that, among methods producing temporally consistent video depth [95, 97, 102, 108], DAGE achieves the highest F1 and the lowest PDBE. While DepthPro [8] attains a higher F1 on some datasets, DAGE yields lower C_{PDBE}—indicating more temporally consistent boundaries in the video setting.

4.3. Multi-view Reconstruction

Following [97, 102], we evaluate reconstructed multi-view *pointmaps* on 7-Scenes [81] and NRGBD [2] under *sparse* and *dense* settings. Predictions are first aligned to ground truth via Umeyama $\mathrm{Sim}(3)$, then refined with ICP. We report accuracy Acc.↓, completeness Comp.↓, and normal consistency NC↑ in Tab. 3. Comparisons include recent feed-forward visual-geometry methods [48, 95, 97, 102, 109, 119]. We also assess metric-scale reconstruction by aligning with rigid transformation SE(3), comparing against metric-pointmap methods [48, 97]. Across sparse and dense settings, DAGE matches state-of-the-art performance [95, 102] while recovering metric-accurate geometry. Fig. 5 shows that our model produces globally consistent pointmaps while preserving fine-grained details.

4.4. Camera Pose Estimation

We evaluate on the synthetic Sintel [9] and two real-world datasets, TUM-Dynamics [84] and ScanNet [20]. We report *Absolute Trajectory Error* (ATE) and *Relative Pose Error* for translation/rotation (RPE$_T$/RPE$_R$). Predicted camera trajectories are registered to ground truth with a $\mathrm{Sim}(3)$ alignment. We summarize the performance in Tab. 4. Notably, we run the LR stream at 252px (long side) to estimate poses efficiently. Competing methods [95, 97, 102] typically require 518px to achieve accurate predictions. Despite using lower resolution inputs, DAGE matches their performance at their high-res settings—and outperform them when evaluated at the same low-res setting.

4.5. Runtime Comparison

Tab. 5 reports the FPS and GPU memory consumption averaged over ten 100-frame videos on a single A100 GPU. DAGE sustains 65.4 FPS at 540p, which is 2× faster than Pi3, and remains tractable at 5.6 FPS on 2K, where global-attention baselines [95, 102] often run out of memory (*OOM*). It is consistently faster than multi-view methods [95, 97, 108] and adds only marginal overhead over the single-view MoGe2 [99]—thanks to the decoupled LR/HR design that confines heavy global attention to the LR path, keeping runtime largely insensitive to HR input size.

Table 3. **Multi-view reconstruction evaluation.** We report the *median* values on 3 settings, including *sparse*, *dense*, and *metric*.

Method	Setting	7-Scenes [81]			NRGBD [2]		
		Acc.↓	Comp.↓	NC↑	Acc.↓	Comp.↓	NC↑
Fast3R [109]	*sparse*	0.065	0.089	0.759	0.091	0.104	0.877
CUT3R [97]		0.049	0.051	0.805	0.041	0.031	0.968
FLARE [119]		0.057	0.107	0.780	0.024	0.025	0.988
VGGT [95]		0.025	0.033	0.845	0.029	0.038	0.981
Pi3 [102]		0.029	0.049	0.841	0.015	0.014	0.992
MapAny [48]		0.053	0.064	0.83	0.064	0.058	0.946
DAGE (ours)		0.027	0.042	0.846	0.018	0.016	0.992
Fast3R [109]	*dense*	0.017	0.018	0.725	0.030	0.016	0.934
CUT3R [97]		0.010	0.008	0.764	0.037	0.017	0.953
FLARE [119]		0.007	0.013	0.785	0.011	0.008	0.986
VGGT [95]		0.008	0.012	0.760	0.010	0.005	0.988
Pi3 [102]		0.007	0.011	0.792	0.008	0.005	0.987
MapAny [48]		0.008	0.008	0.780	0.018	0.010	0.970
DAGE (ours)		0.007	0.009	0.793	0.009	0.006	0.985
CUT3R [97]	*metric*	0.189	0.186	0.582	0.307	0.253	0.606
MapAny [48]		0.339	0.109	0.639	0.156	0.108	0.910
DAGE (ours)		0.034	0.041	0.847	0.085	0.101	0.923

Table 4. **Camera pose evaluation**

Method	Sintel [9]			TUM-dynamics [84]			ScanNet [20]		
	ATE↓	RPE$_T$↓	RPE$_R$↓	ATE↓	RPE$_T$↓	RPE$_R$↓	ATE↓	RPE$_T$↓	RPE$_R$↓
Fast3R [109]	0.371	0.298	13.75	0.090	0.101	1.425	0.155	0.123	3.491
CUT3R [97]	0.217	0.070	0.636	0.047	0.015	0.451	0.094	0.022	0.629
FLARE [119]	0.207	0.090	3.015	0.026	0.013	0.475	0.064	0.023	0.971
VGGT [95]	0.167	0.062	0.491	0.012	0.010	0.311	0.035	0.015	0.382
Pi3 [102]	0.074	0.040	0.282	0.014	0.009	0.312	0.031	0.013	0.347
VGGT (252px)	0.228	0.095	1.03	0.053	0.028	0.652	0.109	0.039	1.357
Pi3 (252px)	0.153	0.088	0.684	0.025	0.019	0.370	0.045	0.017	0.438
DAGE (ours)	0.132	0.051	0.406	0.014	0.010	0.323	0.031	0.014	0.389

Table 5. **Throughput comparison.** FPS↑ / GPU memory↓ (GB) measured on 100-frame clips per resolution.

Resolution	MoGe2†		GeoCrafter		CUT3R		VGGT		Pi3		**DAGE**	
	FPS	Mem.	FPS	Mem.	FPS	Mem.	FPS	Mem.	FPS	Mem.	FPS	Mem.
540×360	79.4	8.1	3.1	17.3	27.2	16.5	30.1	17.3	36.3	17.2	65.4	10.1
960×512	30.0	15.3	1.7	24.1	20.3	19.0	2.1	26.9	3.1	23.1	28.9	18.3
2048×1024	6.1	22.1	*OOM*		4.5	33.2	*OOM*		0.2	66.7	5.6	27.9

† : Methods that do *not* produce temporally consistent geometry.

4.6. Ablation Study

Ablation on Adapter. We investigate strategies to fuse *multi-view–consistent* LR tokens into *high-resolution, fine-grained* HR tokens (Tab. 6a). **Setting A (post-align):** per-frame MoGe2 [99] with *post hoc* alignment to a multi-view–consistent pointmap from a visual-geometry model (e.g., [102]), termed aligned MoGe2; this improves detail but leaves layering/stitching artifacts (see Supp.). **Setting B (interp+SA):** LR tokens interpolated to the HR grid, concatenated with HR tokens, then fused via several HR self-attention layers. **Setting C (all-CA):** adapter blocks use cross-attention only, with HR queries attending to LR keys/values at every block. **Setting D (Alter-Adapter):** a cross/self-attention module inserted after each of the last 5 ViT blocks in the HR stream. Overall, the proposed CrossAttn – SelfAttn adapter, inserted after the HR-stream ViT, consistently outperforms these variants, reducing artifacts and improving cross-view coherence.

Table 6. **Ablation studies** (a) different adapter design, (b) effect of each component on the depth sharpness

Variant	Acc.↓	Comp.↓
A: Aligned MoGe2	0.031	0.028
B: Interp+SA	0.030	0.024
C: All-CA	0.021	0.018
D: Alter-Adapter	0.023	0.018
Ours	**0.018**	**0.016**

(a) Adapter design (NRGBD [2])

Variant	F1↑	Rel^p↓	δ^p↑
Pi3 [102]+AnyUp [103]	0.09	24.5	67.8
Ours	**0.34**	21.5	**75.6**
– W/o mono. prior	0.27	22.6	73.5
– W/o gradient loss	0.31	21.4	75.5
+ With local loss	0.30	**20.9**	75.1

(b) Depth sharpness (Sintel [9])

Table 7. **Ablation study** on the LR stream resolution

Reso.	Sintel [9] (pose)			Sintel [9] (depth)			NRGBD [2]		FPS↑
	ATE↓	RPE$_T$↓	RPE$_R$↓	Rel^p↓	δ^p↑	F1↑	Acc.↓	Comp.↓	
252px (default)	0.132	0.051	0.406	21.5	75.5	0.34	0.018	0.016	65.4
252px (no distill)	0.111	0.057	0.584	22.9	73.0	0.35	0.019	0.017	65.4
336px	0.130	0.042	0.307	20.4	76.9	0.35	0.018	0.016	46.6
518px	0.117	0.039	0.258	19.7	77.7	0.36	0.016	0.016	22.5

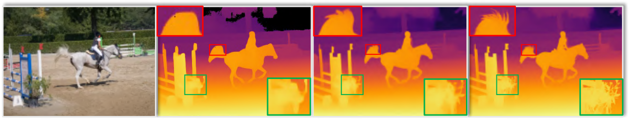


Figure 6. Comparison on disparity quality on Pi3, our variant without monocular prior, and our full model.

Ablation on sharpness depth. Tab. 6b ablates the contribution of the monocular prior [99] and the gradient loss; Fig. 6 shows qualitative results with and without the prior.

Ablation on LR-stream resolution. We vary the LR stream resolution in Tab. 7. In general, increasing LR resolution slightly improves performance but significantly reduces the FPS.

5. Conclusion

We introduced DAGE, a dual-stream visual geometry transformer. A low-resolution stream efficiently estimates cameras and enforces cross-view consistency, while a high-resolution stream preserves sharp details; a lightweight adapter fuses them. This decouples resolution from sequence length, supporting 2K inputs and long videos at practical costs. Empirically, DAGE yields sharper pointmaps and outperforms prior video geometry methods. It matches the 3D reconstruction and pose accuracy of state-of-the-art models [95, 102] while running significantly faster. **Limitations.** Performance can drop under extremely low overlap or rapid non-rigid motion; the HR path is memory-intensive at very high resolutions; and the current method does not recover dynamic motion.

Acknowledgements Evangelos Kalogerakis has received funding from the European Research Council (ERC) under the Horizon research and innovation programme (Grant agreement No. 101124742).

DAGE: Dual-Stream Architecture for Efficient and Fine-Grained Geometry Estimation

Supplementary Material

Table 8. **Training datasets**.

Dataset Name	Scene type	Metric	Real	Dynamic
ARKitScenes [3]	Indoor	Yes	Real	Static
ScanNet [20]	Indoor	Yes	Real	Static
ScanNet++ [115]	Indoor	Yes	Real	Static
TartanAir [101]	Mixed	Yes	Synthetic	Dynamic
Waymo [86]	Outdoor	Yes	Real	Dynamic
BlendedMVS [114]	Mixed	No	Synthetic	Static
HyperSim [74]	Indoor	Yes	Synthetic	Static
MVS Synth [41]	Outdoor	No	Synthetic	Static
GTA-Sfm [96]	Outdoor	No	Synthetic	Static
MegaDepth [58]	Outdoor	No	Real	Static
CO3Dv2 [73]†	Object-centric	No	Real	Static
WildRGBD [106]†	Object-centric	Yes	Real	Static
VirtualKITTI2 [10]	Outdoor	Yes	Synthetic	Dynamic
Matterport3D [11]	Indoor	Yes	Real	Static
BEDLAM [6]†	Mixed	Yes	Synthetic	Dynamic
Dynamic Replica [44]	Indoor	Yes	Synthetic	Dynamic
PointOdyssey [120]†	Mixed	Yes	Synthetic	Dynamic
Spring [64]	Mixed	Yes	Synthetic	Dynamic

† *Only a subset of each dataset is used.*

6. More Training Details

6.1. Training datasets

We train on 18 datasets spanning indoor, outdoor, and object-centric scenes, covering both static and dynamic settings. The full list appears in Tab. 8. Following [97], we filter scenes with ambiguous annotations in PointOdyssey [120], and remove scenes with panorama backgrounds and zoom-in/out effect in BEDLAM [6]. For object-centric datasets [73, 106], we only subsample 40 scenes for each object category.

6.2. Implementation Details

Architecture. For the high-resolution (HR) stream, we initialize the model with the 24-layer ViT from MoGe2 [99] and keep these weights frozen throughout training. For the low-resolution (LR) stream, our training corpus (Sec. 6.1) is considerably smaller than those used by recent feed-forward visual-geometry models [48, 95, 102]. Consequently, rather than training a global video transformer from scratch, we start from the Pi3 [102] checkpoint, which comprises 36 attention layers with alternating frame-wise and global attention. The adapter contains five blocks; each block consists of one cross-attention layer, one self-attention layer, and an MLP. We train the adapter from scratch and *zero-initialize* its final projection to avoid destabilizing the frozen HR features at the start of training. For dense geometry, the pointmap head is implemented as a stack of residual convolutional blocks with transposed convolutions that progressively upsample from patch resolution $(h_{hr} \times w_{hr} = H/14 \times W/14)$ to the original image resolution $(H \times W)$. Camera poses and the scene-wise metric scale factor are predicted with a two-layer MLP. For distillation during training, we use a 2-layer MLP and *pixel shuffle* to project $\mathcal{F}^{\mathrm{lr}}$ to the teacher spatial resolution.

Training loss. We set the weight for each loss as follow: $\lambda_{\mathrm{pm}} = 1.0, \lambda_{\mathrm{cam}} = 0.1, \lambda_{\mathrm{trans}} = 100.0, \lambda_{\mathrm{rot}} = 1.0, \lambda_{\mathrm{scale}} = 1.0, \lambda_{\mathrm{normal}} = 1.0, \lambda_{\mathrm{gradient}} = 0.1, \lambda_{\mathrm{distill}} = 0.5$.

Optimization. We train our model in two stages. *Stage 1* targets low–medium resolutions with longer clips; *Stage 2* fine-tunes on high-resolution inputs with short clips. Both stages use AdamW [60] optimizer with a OneCycleLR schedule. In Stage 1 (30,000 steps), we set a base LR of 1×10^{-4} for the adapter and dense heads, and 1×10^{-5} (10× lower) for the global transformer initialized from Pi3. In Stage 2 (10,000 steps), we freeze the global transformer and fine-tune only the adapter and heads at 1×10^{-5}. To keep training efficient, we use FlashAttention [21, 22], `bfloat16` mixed precision, gradient checkpointing, and gradient accumulation. With this setup, training takes roughly five days on 16×A100-80GB GPUs.

Augmentation and sampling. We extend the MoGe [98] augmentation pipeline to the multi-view setting and adopt stage-specific regimes. *Stage 1 (long sequence):* we sample 2–24 frames per clip and constrain the total pixels to $[1.0 \times 10^5,\ 2.55 \times 10^5]$, thereby enabling a large per-GPU batch of 48 images; we apply distillation only in this stage. *Stage 2 (high resolution):* we sample just 2–4 frames per clip, set the total pixels to $[2.7 \times 10^5,\ 9.0 \times 10^5]$ (roughly 518×518–952×952 for 14-px patches), vary the aspect ratio within $[0.5,\ 2.0]$, and use 24 images per GPU.

7. More Evaluation Details

This section details the datasets and metrics used in our experiments.

7.1. Video geometry estimation.

Datasets. Following GeometryCrafter [108], we configure each test dataset as follows:

- **GMU Kitchens** [33]: We use all scenarios, extract 110 frames per sequence with a stride of 2, and downsample the 1920p videos and depth maps to 960×512.
- **Monkaa** [63]: We select 9 scenes and truncate each sequence to 110 frames at the native resolution of 960×512.
- **Sintel** [9]: We use all training sequences (21–50 frames) and crop from 1024×436 to 896×448.
- **ScanNet** [20]: We evaluate 100 test scenes with 90 frames per video (stride 3), and center-crop each frame to 640×512.
- **KITTI** [32]: We use all sequences from the depth-annotated validation split; for longer videos we keep the first 110 frames (yielding 13 videos with 67–110 frames), and center-crop to 768×384.
- **Diode** [92]: We use all 771 validation images at the default resolution of 1024×768.

In addition, we prepare two high-resolution evaluation sets:

- **UrbanSyn** [34]: We sample ten clips of 100 frames each from the original 7000-frame sequences and keep the resolution at 2048×1024.
- **Unreal4K** [91]: We use all nine scenes, keep the first 100 frames per scene, and downsample to 1920×1080.

Metrics. For the pointmap estimation, we report the mean relative point error $\mathrm{Rel}^p \downarrow = \|\hat{\mathbf{p}} - \mathbf{p}\|_2 / \|\mathbf{p}\|_2$ and the inlier ratio $\delta^p \uparrow$, where a point is an inlier if $\|\hat{\mathbf{p}} - \mathbf{p}\|_2 / \min(\|\mathbf{p}\|_2, \|\hat{\mathbf{p}}\|_2) < \tau$ (with τ=0.25), averaged over valid pixels. Similarly, we leverage $\mathrm{Rel}^d \downarrow$ and $\delta^d \uparrow$ for depth estimation.

7.2. Video sharpness depth.

Datasets. We evaluate depth–boundary sharpness on four synthetic datasets—Monkaa [63], Sintel [9], UrbanSyn [34], and Unreal4K [91].

Metrics. We use the F1↑ edge metric from DepthPro [8]. For each pair of neighboring pixels, we mark an occluding contour when the depth ratio exceeds a *predefined threshold*. Applying this to both prediction and ground truth yields two contour maps. *Precision* is the fraction of predicted contour pairs that are also contours in the ground truth, and *recall* is the fraction of ground-truth contour pairs recovered by the prediction. The F1 score is the harmonic mean of precision and recall. We report the F1 averaged over multiple thresholds. This metric requires no ground-truth edge maps and is easily computed wherever dense depth annotations are available (e.g., synthetic data).

To further assess boundary sharpness, we adopt the Depth Boundary Error (DBE) from iBims [50] and use its pseudo variant (PDBE) for datasets without depth–edge annotations (following [69]). Concretely, we run Canny edge detection on both predicted and ground-truth depth maps to obtain edge sets, then compute the iBims accuracy and completeness terms. The accuracy term penalizes predicted edges that are far from any ground-truth edge, while the completeness term penalizes ground-truth edges not recovered by the prediction. Finally, we report the *chamfer distance* $\mathcal{C}_{\mathrm{PDBE}} \downarrow$, which is the average of accuracy and completeness.

7.3. Multi-view reconstruction.

Datasets. We evaluate 3D pointmap reconstruction on 7-Scenes [81] and NRGBD [2] under both sparse and dense view protocols. For sparse views, we sample keyframes every 200 frames on 7-Scenes and every 500 on NRGBD; for dense views, the strides are 40 and 100, respectively.

Metrics. We employ the *Accuracy* (Acc↓): mean nearest-neighbor distance from each predicted point to the ground truth, *Completion* (Comp↓): mean nearest-neighbor distance from each ground-truth point to the reconstruction, and *Normal Consistency* (NC↑): mean absolute dot product of ground truth and predicted normals (computed on the fly using `Open3D` library).

7.4. Camera pose estimation.

Datasets. We evaluate on Sintel [9], TUM-Dynamics [84], and ScanNet [20]. For Sintel, we follow [14, 118], excluding static scenes and those with perfectly straight camera motion, leaving 14 sequences. For TUM-Dynamics and ScanNet, we use the first 90 frames with a temporal stride of 3.

Metrics. Following [97, 102, 118], we report Absolute Trajectory Error (ATE↓) and Relative Pose Error for translation and rotation ($\mathrm{RPE}_T \downarrow$ / $\mathrm{RPE}_R \downarrow$). Predicted trajectories are first aligned to ground truth with a single $\mathrm{Sim}(3)$ transform (global scale, rotation, translation). ATE is the root-mean-square discrepancy between aligned and ground-truth camera positions over the entire sequence. RPE_T is the translation error over a certain distance, and RPE_R is the rotation error over a certain degree; both are averaged over all pose pairs.

8. More Results

8.1. Video geometry estimation

We evaluate video geometry estimation under four other settings. First, for *scale-invariant* video pointmaps, we align predictions to ground truth with a *single* per-video scale and report results in Tab. 9. Second, for video *depth*, we follow standard practice and report both *affine-invariant* results—per-frame scale + shift alignment—in Tab. 10, and *scale-invariant* results—single per-video scale—in Tab. 11. Finally, we assess *metric-scale* video pointmaps with **no** alignment (direct comparison in the dataset's metric units); see Tab. 12. For the metric setting, we compare against

Table 9. **Scale-invariant video pointmap evaluation**. Results are aligned with the ground truth by optimizing a shared scale factor across the entire video. We mark best and second-best.

Method	**GMU** [33]		**Monkaa** [63]		**Sintel** [9]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Unreal4K** [91]		**Diode** [92]		**Rank** ↓
	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	
DepthPro [8]	10.5	92.7	27.9	51.2	55.0	37.5	9.3	95.0	11.7	93.6	22.5	61.1	96.1	1.2	30.3	58.1	7.6
MoGe [98]	21.4	69.0	27.7	58.3	29.5	59.8	13.4	88.2	8.6	95.6	13.4	89.9	34.0	55.7	30.3	53.4	6.9
MoGe2 [99]	19.7	72.1	30.8	51.1	34.3	47.7	12.7	89.4	11.7	96.9	12.3	91.7	30.1	62.3	29.5	55.4	7.0
MoGe2 [99]†	7.1	94.6	25.8	60.2	33.1	52.1	7.8	97.5	10.5	98.4	6.5	97.2	8.9	92.1	15.8	84.1	3.9
CUT3R [97]	8.2	93.6	34.9	45.9	42.9	35.8	6.5	98.0	16.0	88.1	57.9	14.0	17.5	78.3	17.2	81.6	6.9
VGGT [95]	5.6	93.8	16.0	80.4	26.7	65.8	3.1	99.0	8.4	97.3	18.5	75.0	8.7	96.5	13.6	80.2	3.6
Pi3 [102]	5.4	94.2	12.6	90.2	29.6	62.5	2.4	99.4	9.2	90.8	10.7	93.8	17.2	75.4	9.0	96.1	3.1
GeoCrafter [108]	8.4	94.5	20.7	73.9	30.2	57.8	8.9	96.4	6.4	98.8	11.3	95.3	21.0	73.5	13.0	92.8	4.1
DAGE (ours)	5.0	94.2	11.3	88.1	26.6	66.2	2.4	99.5	7.3	99.0	7.9	96.6	9.2	92.9	10.0	94.4	**1.7**

Table 10. **Affine-invariant video depthmap evaluation**. Results are aligned with the ground truth by optimizing a shared scale and shift factor across the entire video. We mark best and second-best.

Method	**GMU** [33]		**Monkaa** [63]		**Sintel** [9]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Unreal4K** [91]		**Diode** [92]		**Rank** ↓
	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	
DepthPro [8]	8.8	93.0	23.3	55.8	36.1	49.5	8.1	94.6	11.7	93.6	51.3	38.3	105.0	20.0	31.0	58.8	8.0
MoGe [98]	19.9	66.5	19.9	63.6	26.5	60.0	12.5	85.9	7.6	94.2	15.2	82.1	38.7	46.2	31.3	48.2	7.6
MoGe2 [99]	19.0	68.9	20.8	60.8	26.4	59.9	12.1	86.9	6.7	96.7	13.8	85.4	33.6	52.8	29.5	50.4	6.8
MoGe2 [99]†	6.6	93.8	18.0	68.1	25.0	63.8	6.4	96.8	5.2	98.3	7.7	96.4	12.0	88.3	14.7	80.7	3.8
CUT3R [97]	7.3	92.9	28.0	49.9	31.9	50.5	5.4	97.4	10.2	89.1	48.6	36.7	15.4	79.7	16.0	78.4	6.8
VGGT [95]	5.2	93.2	12.3	80.5	22.2	70.4	2.7	98.7	4.7	97.2	13.9	84.5	8.2	94.3	12.4	85.2	3.5
Pi3 [102]	4.9	93.5	8.2	91.4	20.2	71.7	2.0	99.3	3.0	99.1	16.0	78.8	18.3	78.6	8.6	92.9	2.7
GeoCrafter [108]	7.7	94.1	13.4	79.3	21.4	70.6	7.3	96.1	5.0	98.5	12.2	90.3	20.7	72.2	9.1	93.4	3.9
DAGE (ours)	4.8	93.5	9.5	87.2	19.5	74.4	2.1	99.4	3.2	98.8	7.7	95.8	12.1	88.1	8.7	92.5	**1.9**

methods capable of predicting metric geometry, including CUT3R [97] and MapAnything [48].

We additionally evaluate feed-forward visual-geometry approaches at each dataset's native resolution (540p–2K). As reported in Tab. 13, performance degrades steadily with increasing resolution; at the highest, far beyond training scales (e.g. Urbansyn and Unreak4k datasets), most methods collapse except ours.

8.2. Single-image geometry estimation

Following [98, 99], we evaluate the single-image geometry estimation on eight different datasets, including NYUv2 [82], KITTI [32], ETH3D [79], iBims-1 [50], GSO [25], Sintel [9], DDAD [35], DIODE [92], HAMMER [43]. The results are summaried in Tab. 14, validating that our dual-stream design preserves single-image quality compared to single-image based methods like DepthPro [8], MoGE [98, 99].

8.3. Camera pose estimation

We additionally report the predicted camera poses on RealEstate10K and CO3Dv2 datasets. We report the Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA) at a given threshold, and the Area Under the Curve (AUC) of the min(RRA,RTA) threshold curve. Tab. 15 shows that DAGE remains competitive with Pi3 [102] and VGGT [95], even while operating at a lower resolution.

8.4. More ablation studies

Low-resolution stream architecture. We perform an ablation study of the global module in our LR stream. Specifically, in addition to the global transformer with alternative frame/global attention, we ablate with two other design: (1) transformer-based recurrent network [97] and (2) temporal Mamba network [17]. Results in Tab. 16a show that the alternating global-attention transformer consistently outper-

Table 11. **Scale-invariant video depthmap evaluation**. Results are aligned with the ground truth by optimizing a shared scale factor across the entire video. We mark best and second-best.

Method	**GMU** [33]		**Monkaa** [63]		**Sintel** [9]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Unreal4K** [91]		**Diode** [92]		**Rank** ↓
	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	
DepthPro [8]	9.4	92.1	26.7	45.9	53.6	35.3	8.8	92.9	8.2	92.5	22.2	40.2	96.0	1.1	29.3	56.4	7.9
MoGe [98]	20.7	64.7	25.5	54.8	31.4	48.9	13.3	85.0	7.7	94.1	13.1	86.3	34.7	49.8	29.8	48.2	7.6
MoGe2 [99]	19.5	67.1	27.1	51.6	31.2	47.7	12.0	86.5	7.2	96.7	12.0	88.8	30.7	56.6	28.5	50.7	6.9
MoGe2 [99]†	6.7	93.8	21.9	60.4	30.1	51.9	7.1	95.9	5.6	98.3	6.0	96.8	8.7	91.0	14.7	80.7	3.7
CUT3R [97]	7.9	92.6	33.0	38.3	37.3	42.4	5.8	97.0	11.3	86.8	22.2	63.1	16.8	79.6	15.6	80.0	6.7
VGGT [95]	5.2	93.0	14.4	77.3	25.3	62.1	2.8	98.6	5.3	96.7	18.3	73.3	8.2	96.1	13.4	79.2	3.6
Pi3 [102]	4.9	93.4	10.8	88.9	28.4	60.6	2.1	99.3	3.1	99.1	9.5	92.5	16.6	75.0	8.7	95.5	2.3
GeoCrafter [108]	8.1	93.8	18.1	71.1	27.1	58.7	7.9	95.5	5.1	98.4	11.0	92.4	21.1	70.9	10.0	92.4	4.1
DAGE (ours)	4.7	93.4	11.5	85.5	25.6	64.8	2.2	99.4	3.3	98.8	7.9	95.9	8.7	90.3	9.9	94.0	**1.9**

Table 12. **Metric video pointmap evaluation**. Predicted pointmaps are directly compared with ground truth.

Method	**GMU** [33]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Diode** [92]	
	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑	Rel^p↓	δ^p↑
CUT3R [97]	13.5	90.7	9.1	95.2	34.2	14.6	15.4	84.4	31.6	47.2
MapAny [48]	22.6	63.4	35.2	28.5	29.1	26.3	28.5	37.3	33.8	31.8
DAGE (ours)	7.5	95.3	2.5	99.5	12.0	98.3	8.3	96.5	12.9	87.5

forms both variants, reflecting stronger multi-view aggregation and more reliable cross-view consistency.

RoPE design in the adapter. We ablate rotary positional encodings (RoPE) in the adapter in Tabs. 16b and 16c. For self-attention (Tab. 16b), standard RoPE [85] is ineffective at high resolutions (e.g., UrbanSyn dataset), whereas interpolated RoPE improves performance. For cross-attention (Tab. 16c), adding RoPE alongside our alignment ("snapping") further boosts results.

8.5. More qualitative results

Interactive viewer (highly recommended). The supplementary contains an HTML page (`webpage/index.html`) with side-by-side videos of predicted depth and reconstructed 3D pointmaps.

Fig. 7 shows qualitative 3D pointmap reconstructions on in-the-wild scenes spanning static/dynamic motion, indoor/outdoor settings, and object-centric versus scene-level compositions.

Figs. 8, 9, 10 compare our video depth to recent state-of-the-art methods [95, 102, 108], highlighting sharper boundaries and stronger temporal stability.

Fig. 11 visualizes depth-edge maps—the contours obtained by thresholding neighboring-pixel depth changes. Compared to baselines [95, 102, 108], our results capture thin structures and small or distant objects more reliably.

Fig. 12 compares 3D pointmaps from DAGE to an *aligned-MoGe2* baseline. In Tab. 6 (Sec. 4.6), we define **Setting A**: run MoGe2 [99] per frame and *post hoc* align each predicted pointmap to a globally consistent pointmap from Pi3 [102]. This simple alignment recovers fine detail and enforces a shared scale, but—as the figure shows—still produces layering/stitching artifacts because depth is estimated independently per frame without strong cross-view coupling.

Fig. 13 visualizes 3D pointmaps reconstructed from 2K inputs. DAGE runs substantially faster—especially on longer clips—while producing more plausible, multi-view–consistent reconstructions. In contrast, global-attention baselines [95, 102] either run out of memory or degrade at this resolution.

9. High-resolution inference analysis of visual-geometry models

We analyze how pretrained feed-forward visual-geometry models [95, 102] behave when evaluated well beyond their training resolution (up to 2K on the long side).

Single-image stress test. We resize single-image inputs to several resolutions (e.g., 540p, 1080p, and 2K) and run the public checkpoints of VGGT [95] and Pi3 [102] without any architectural changes. We visualize depth maps and corresponding 3D pointmaps (VGGT in Fig. 14a, Pi3 in

Table 13. **Affine-invariant video pointmap evaluation at native resolution.** Predictions are aligned to ground truth by optimizing a single scale and shift across the entire video.

Method	**GMU** [33]		**Monkaa** [63]		**Sintel** [9]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Unreal4K** [91]		**Diode** [92]	
	(960 × 512)		(960 × 512)		(896 × 448)		(640 × 512)		(768 × 384)		(2048 × 1024)		(1920 × 1080)		(1024 × 768)	
	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑
CUT3R [97]	22.0	67.9	30.9	51.0	38.9	40.9	7.0	97.9	13.2	88.7	56.7	13.9	71.6	5.6	31.1	52.6
VGGT [95]	15.9	91.4	17.7	81.6	28.7	63.8	4.5	99.1	7.8	97.5	OOM	OOM	OOM	OOM	20.5	76.3
Pi3 [102]	6.2	92.2	12.6	88.9	21.7	72.9	2.2	99.5	5.9	97.5	55.9	14.7	54.2	17.1	13.9	87.2
DAGE (ours)	4.9	94.2	10.1	91.0	21.5	75.6	2.1	99.5	5.9	99.0	8.8	96.0	11.9	89.1	9.7	94.4

Table 14. **Single-image geometry evaluation**. Results are aligned with the ground truth by optimizing a scale and shift factor for each image. We mark best and second-best.

Method	**NYUv2** [82]		**KITTI** [32]		**ETH3D** [79]		**iBims-1** [50]		**GSO** [25]		**Sintel** [9]		**DDAD** [35]		**DIODE** [92]		**HAMMER** [43]		**Rank**↓
	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	Rel^d↓	δ^d↑	
DepthPro [8]	4.36	97.9	9.15	90.7	7.73	94.0	4.34	97.4	3.16	99.7	19.6	74.5	14.4	81.2	6.28	93.7	5.31	98.8	3.8
MoGe [98]	3.68	98.3	4.86	97.2	3.57	99.0	3.61	97.3	1.14	100	16.8	77.8	10.5	91.4	4.37	96.4	3.88	98.1	1.8
MoGe2 [99]	3.33	98.4	6.47	96.4	3.89	98.7	3.65	98.5	1.16	100	17.4	77.0	10.1	90.3	5.13	94.9	4.19	99.1	1.9
DAGE (ours)	3.34	98.4	7.52	94.7	3.49	98.0	3.70	97.8	1.26	99.9	18.9	74.8	10.7	89.2	4.97	94.6	3.43	98.6	2.5

Fig. 14b).

At ∼ 540p, both methods produce plausible geometry. When the resolution is increased to ∼ 1080p, predictions exhibit shape distortions; at ∼ 2K, outputs often collapse into fragmented or globally inconsistent pointmaps. These failures are consistent across scenes.

Global-attention behavior (3-view input). To probe failure modes under high resolution, we evaluate VGGT with *triplets* of views (3 frames), not single images. We fix the number of views to three and vary only the spatial resolution. Following prior observations that global layers perform exhaustive correspondence search [94], we visualize post-softmax maps for a few query tokens in view 1 and overlay their responses in the other views (Figs. 15–17). At ∼ 540p, the maps are compact and centered on true correspondences. As resolution increases, attention becomes diffuse and multi-modal, drifting toward semantically similar yet geometrically incorrect regions; by 2K it degenerates into high-entropy responses with no clear matches.

Likely causes. (i) *Positional extrapolation:* standard rotary/absolute positional parameterizations learned at ∼540 px do not extrapolate reliably to much larger token grids, skewing query–key phases and degrading similarity scores [12]. (ii) *Entropy growth:* increasing resolution raises token count without increasing the effective receptive field, making correspondence sparser per token and increasing attention entropy [42]. (iii) *Distribution shift:* training rarely exposes models to high-frequency, high-resolution statistics; the learned global matcher thus overfits to lower-res aliasing patterns.

From our experiments, we find that naively scaling input resolution is unreliable for current global-attention pipelines: at 1K–2K, pretrained models often exhibit correspondence collapse—diffuse attention and distorted depth/pointmaps—likely due to positional-encoding extrapolation and distribution shifts. Therefore, in our proposed DAGE, we amortize global aggregation at low resolution and fuse it into a per-frame high-resolution path; this preserves detail at 2K while keeping memory and runtime practical. Furthermore, to stabilize high-res inference, we adopt resolution-aware positional encodings (interpolated RoPE), explicit cross-scale alignment (snapping HR token coordinates to the LR grid for cross-attention), and multi-scale training that includes high-res regimes.

Table 15. **Pose Estimation** on RealEstate10K and Co3Dv2

Method	**RealEstate10K**			**Co3Dv2**		
	RRA@30 ↑	RTA@30 ↑	AUC@30 ↑	RRA@30 ↑	RTA@30 ↑	AUC@30 ↑
VGGT (518px)	99.97	93.13	77.62	98.64	97.62	91.28
Pi3 (518px)	99.99	95.62	85.90	98.49	97.53	91.39
DAGE (252px)	99.98	95.22	83.12	98.74	97.71	90.71

Table 16. **Ablations.** (a) LR-stream architectures. (b,c) Positional encodings.

(a) **Ablation on different architectures of the LR stream**.

	GMU [33]		**Monkaa** [63]		**Sintel** [9]		**ScanNet** [20]		**KITTI** [32]		**UrbanSyn** [34]		**Unreal4K** [91]		**Diode** [92]	
Method	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$
MoGe2	19.6	72.4	25.0	57.0	29.8	58.4	12.4	89.4	9.0	97.2	13.4	90.0	32.9	59.1	31.0	54.2
Mamba	8.4	93.5	17.8	77.1	27.7	63.5	7.0	97.1	5.9	98.1	10.1	91.0	24.0	60.0	25.1	66.5
Trans. RNN	6.7	94.4	22.2	68.8	27.9	64.5	4.9	98.8	7.6	98.2	9.3	93.9	15.7	80.0	17.3	81.6
Global Trans.	4.9	94.2	10.1	91.0	21.5	75.6	2.1	99.5	5.9	99.0	8.8	96.0	11.9	89.1	9.7	94.4

(b) **Effect of RoPE in the self-attention**.

Positional Embedding	**Monkaa** [63]		**UrbanSyn** [34]	
	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$
None	11.0	89.9	10.1	93.9
RoPE	**9.7**	**92.1**	10.3	93.5
Interp. RoPE (ours)	10.1	91.0	**8.8**	**96.0**

(c) **Effect of RoPE in the cross-attention**.

Positional Embedding	**Monkaa** [63]		**UrbanSyn** [34]	
	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$	$\text{Rel}^p \downarrow$	$\delta^p \uparrow$
None	10.7	**91.1**	9.6	95.1
“Snap” RoPE (ours)	**10.1**	91.0	**8.8**	**96.0**

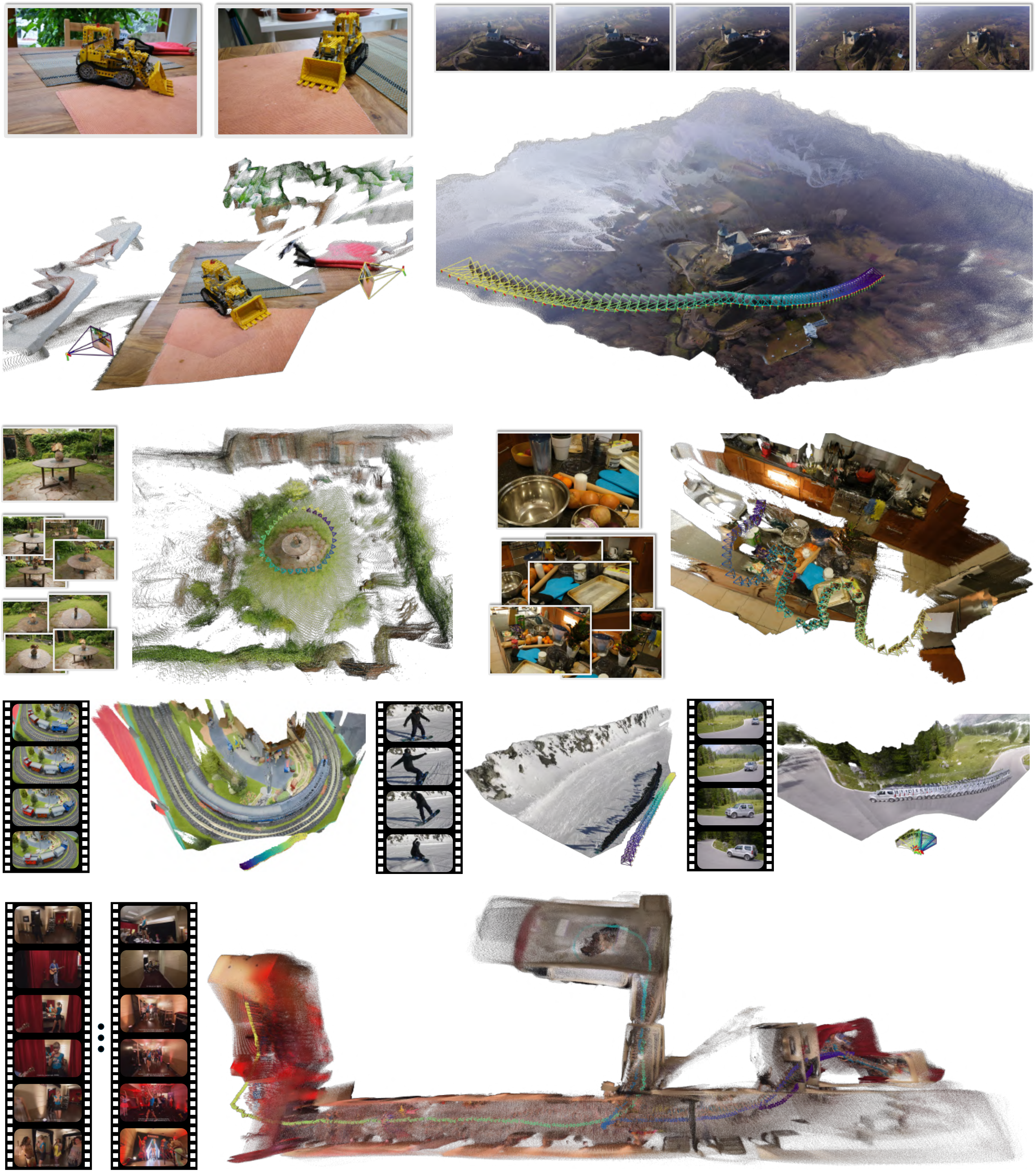


Figure 7. **Visualization of 3D pointmap reconstruction** on *in-the-wild* scenarios.

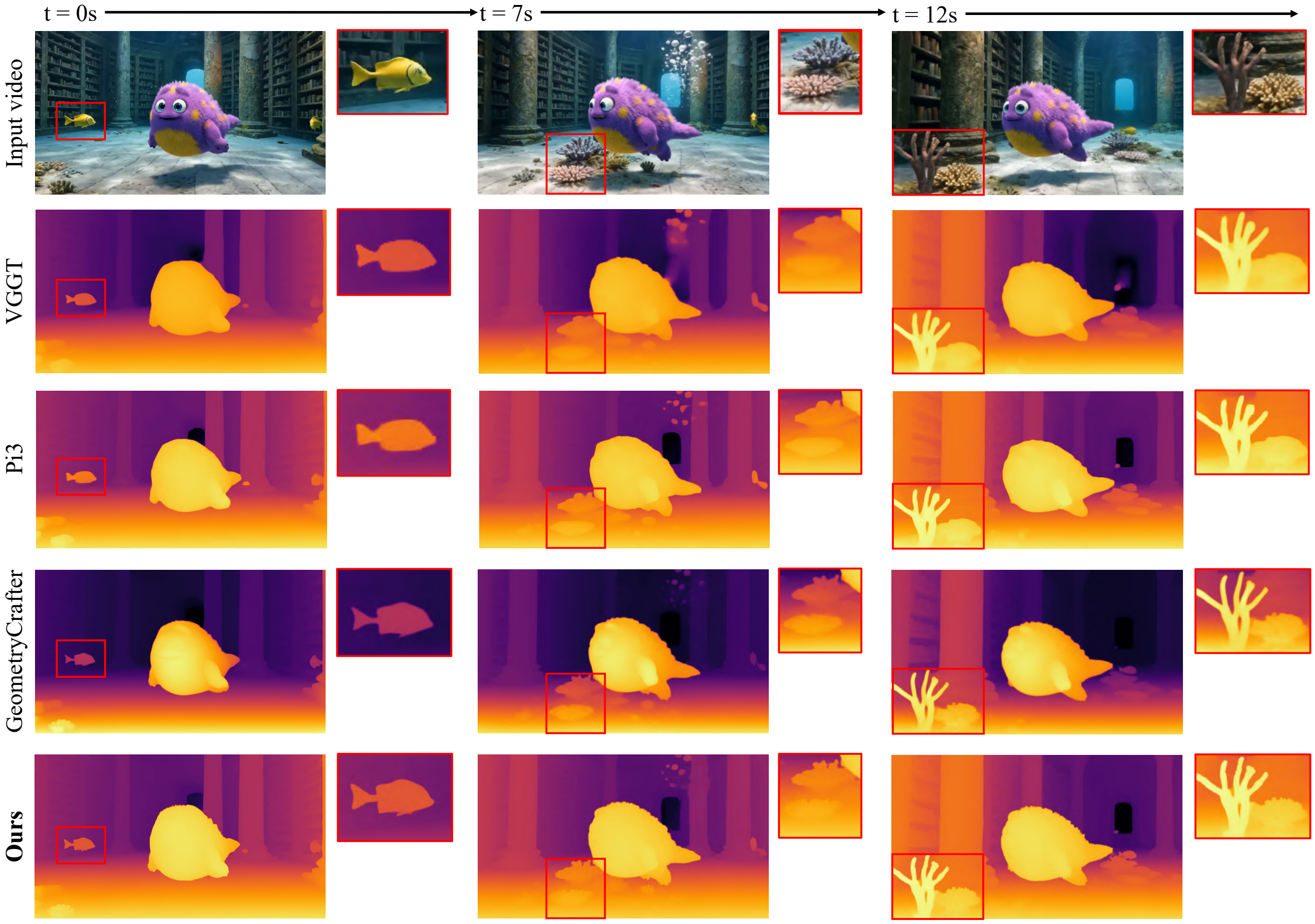


Figure 8. **Visualization of video depth estimation**. We compare our video depth prediction with VGGT [95], Pi3 [102], and GeoemtryCrafter [108]. DAGE demonstrates more sharp and fine-grained predictions.

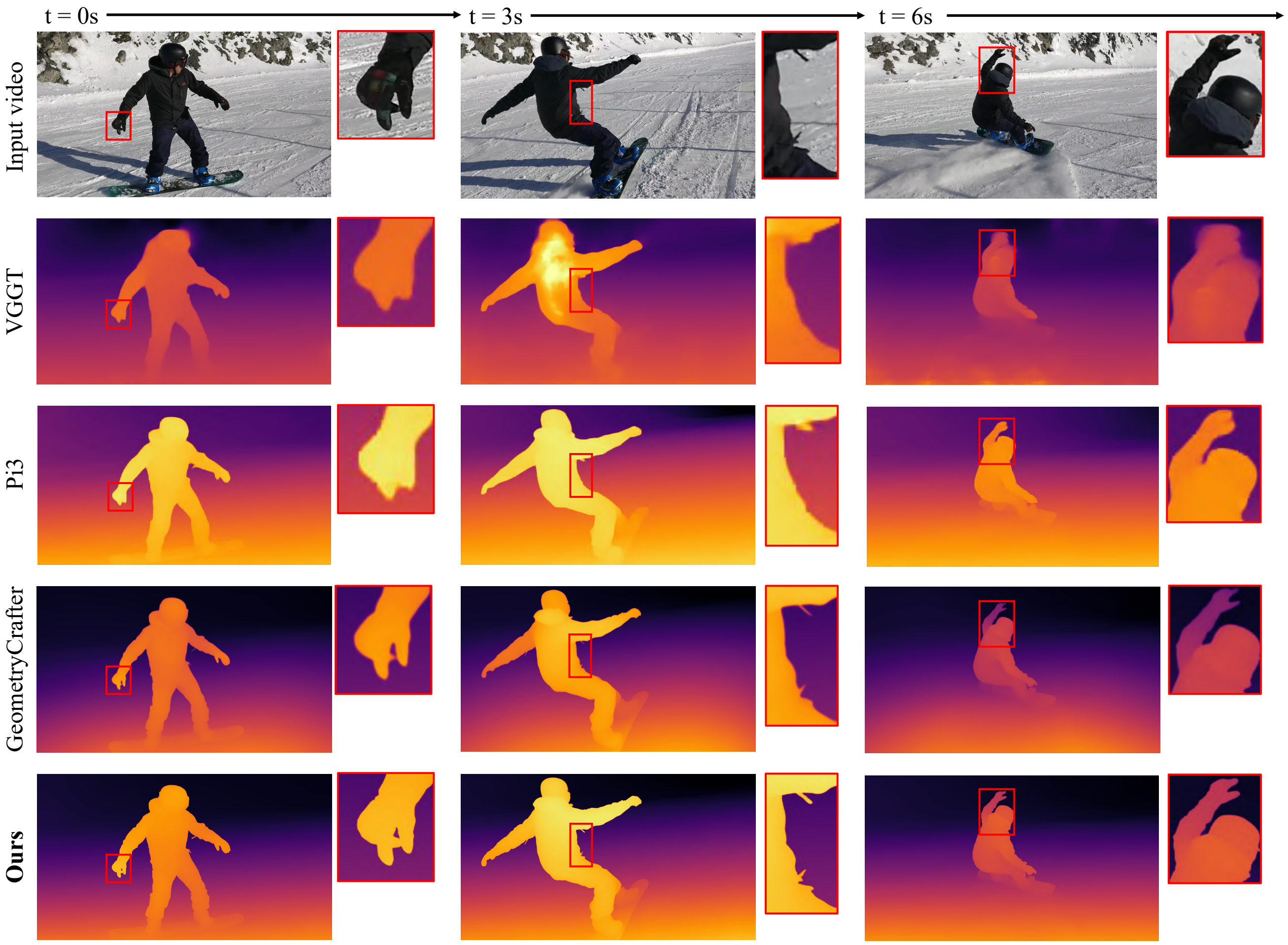


Figure 9. **Visualization of video depth estimation**. We compare our video depth prediction with VGGT [95], Pi3 [102], and GeoemtryCrafter [108]. DAGE demonstrates more sharp and fine-grained predictions.

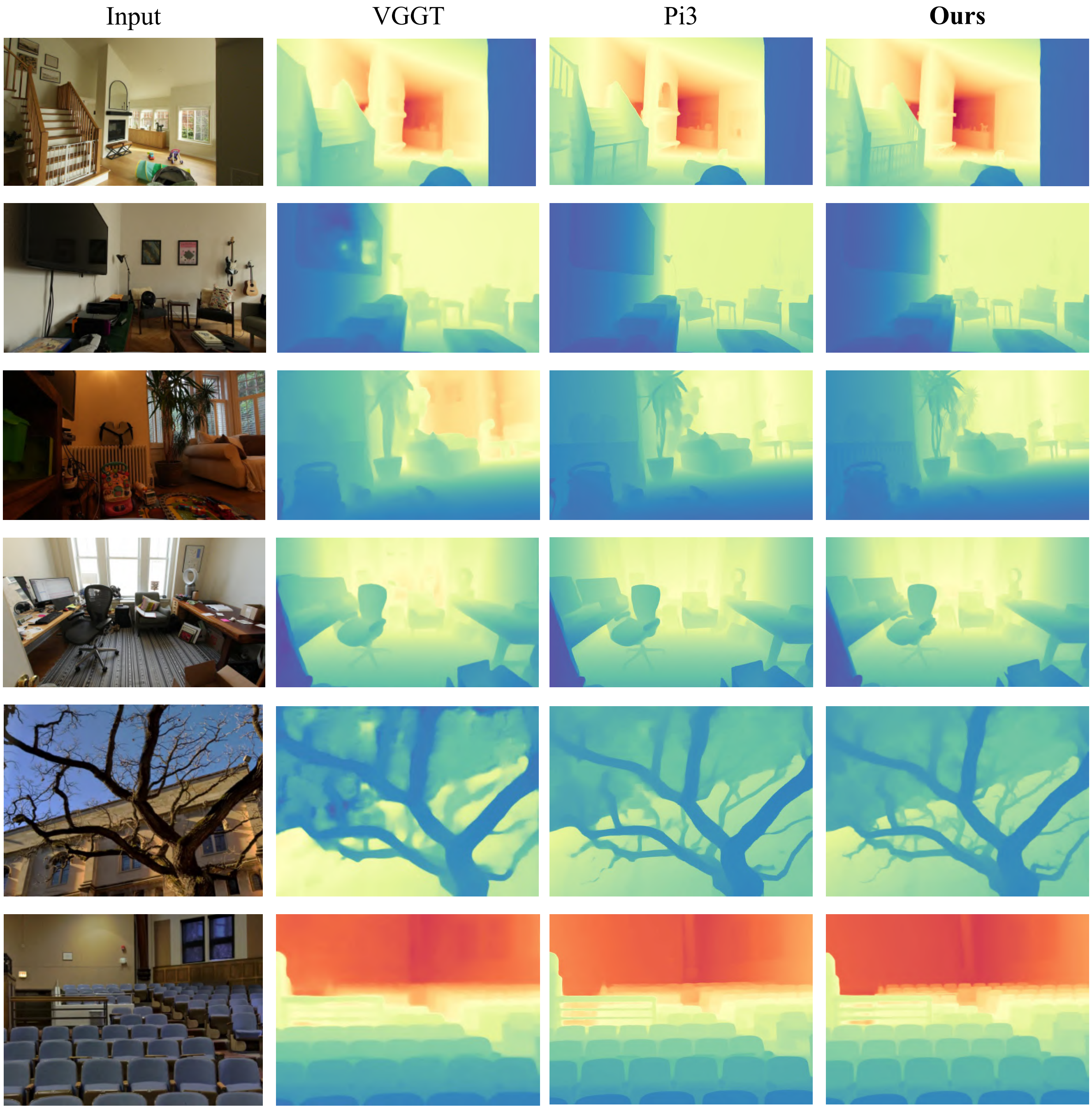


Figure 10. **Visualization of depth estimation** on static scenes.

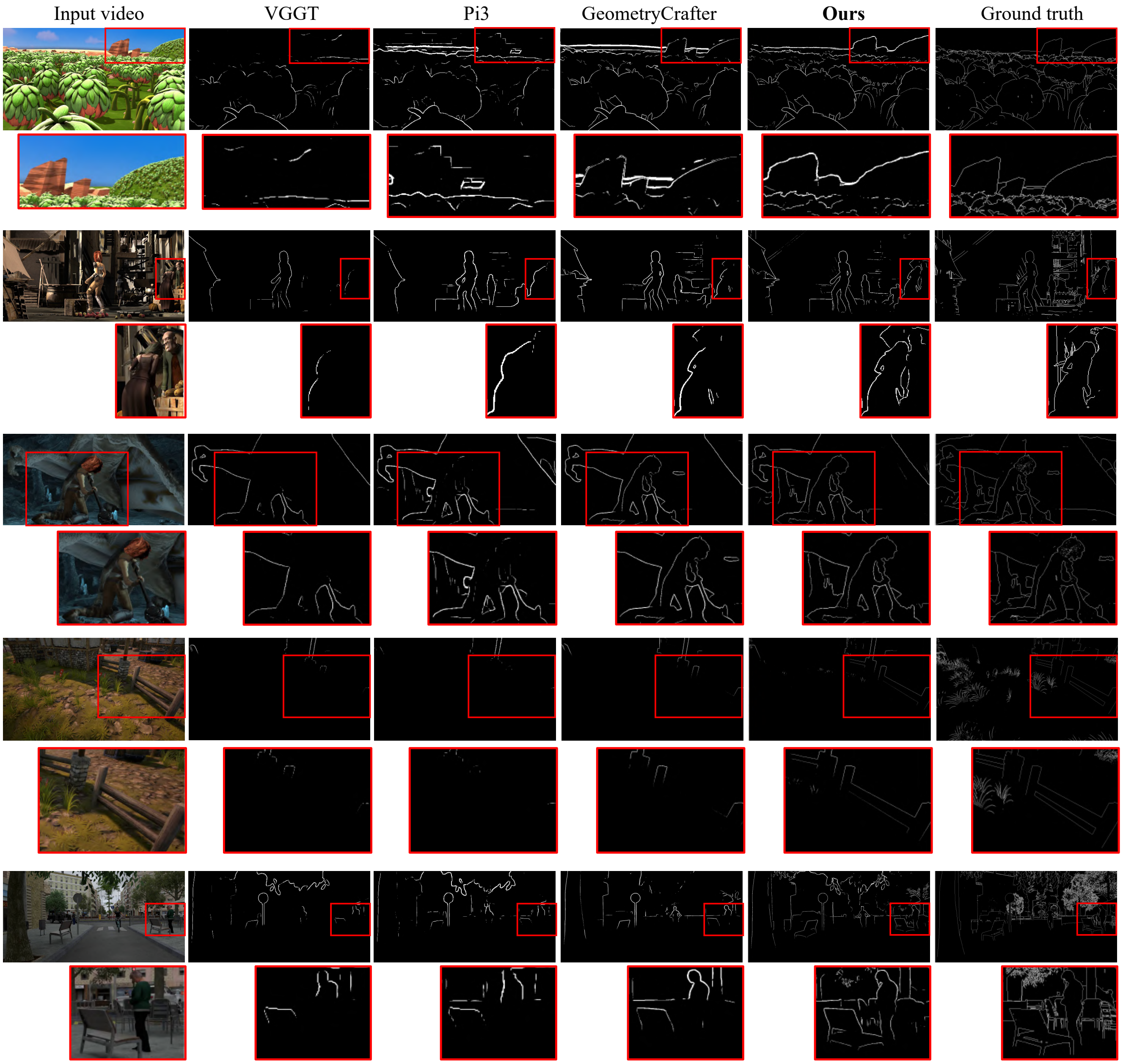


Figure 11. **Visualization of predicted depth edge maps**, which are defined by a depth ratio between neighboring pixels above a threshold. We zoom-in the edge map details in the red bounding boxes.

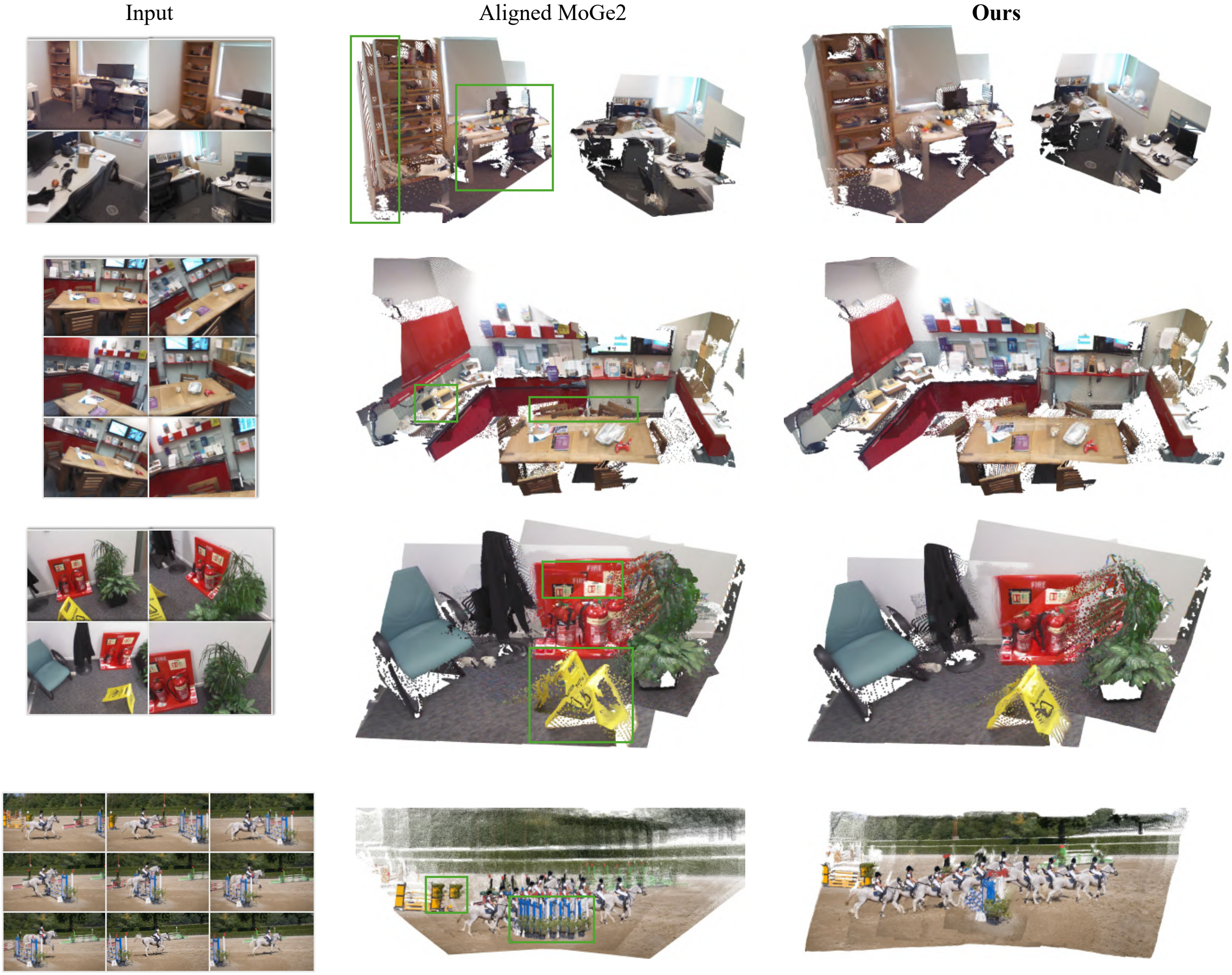


Figure 12. **Predicted 3D pointmaps** of the aligned MoGe2 baseline and our method. The aligned MoGe2 baseline exhibits layering artifacts (green boxes) due to the lack of strong multi-view binding.

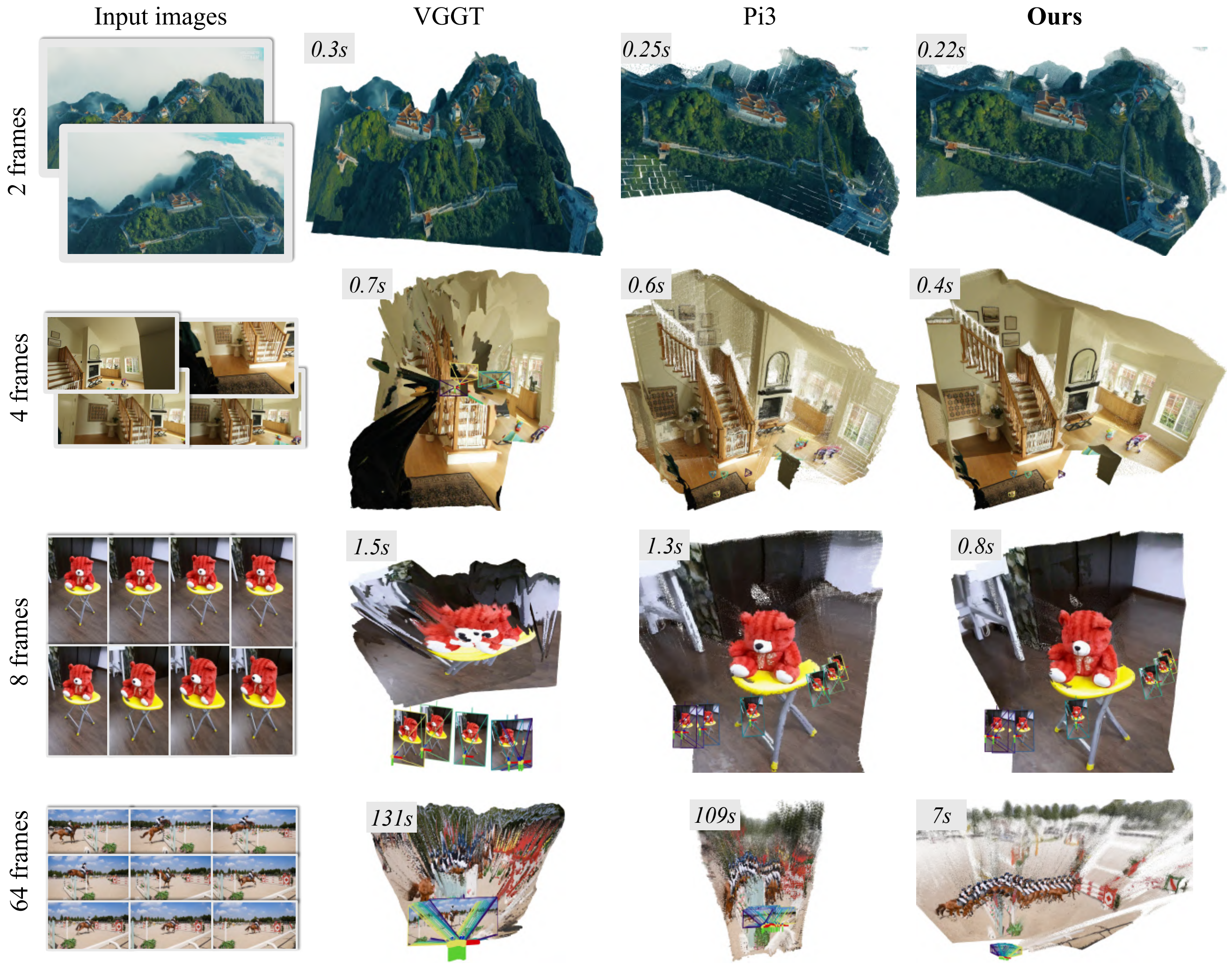
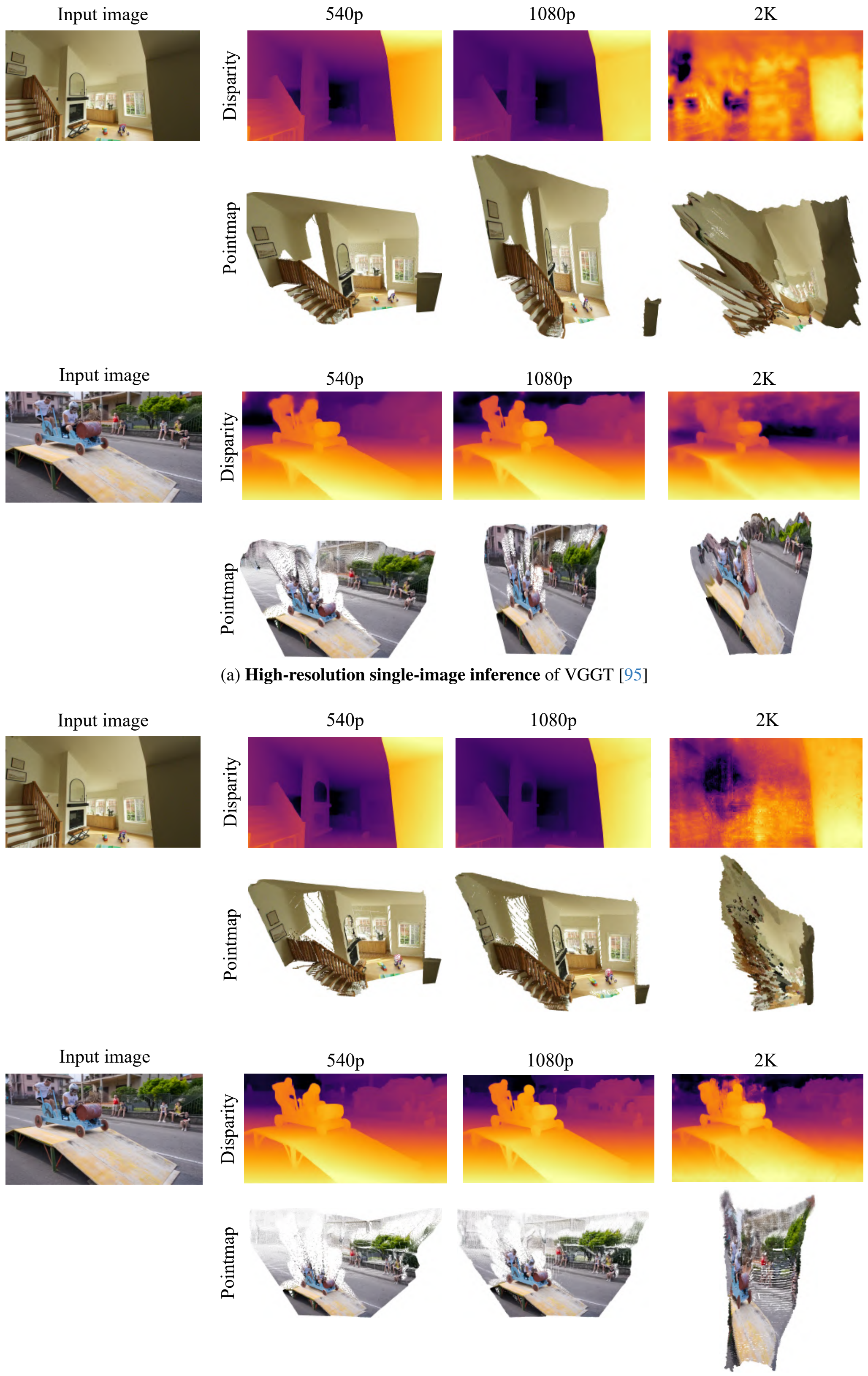


Figure 13. **Visualization of 3D reconstruction** with high-resolution inputs.



(a) **High-resolution single-image inference** of VGGT [95]

(b) **High-resolution single-image inference** of Pi3 [102]

Figure 14. Qualitative results for high-resolution single-image inference: (a) VGGT [95] and (b) Pi3 [102].

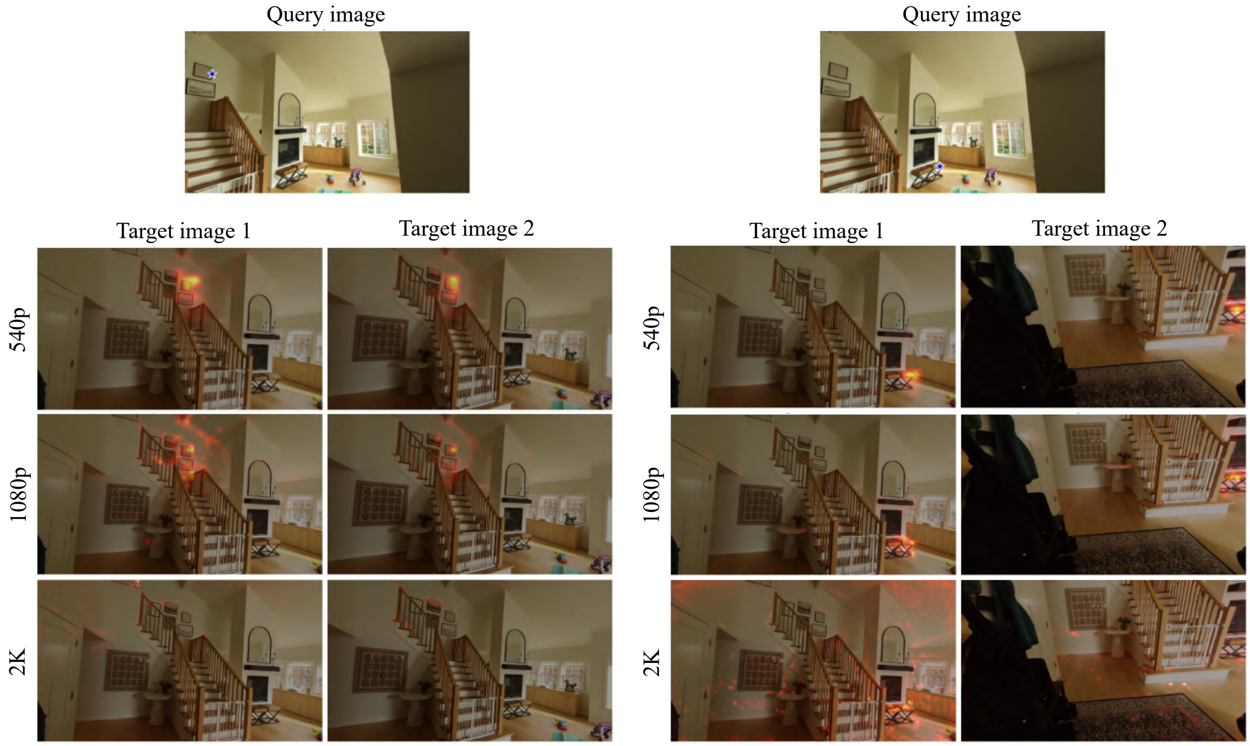


Figure 15. **Attention map** of the 15th global-attention layer of VGGT [95] at different input resolutions. The query token in the first image is marked with a blue star.

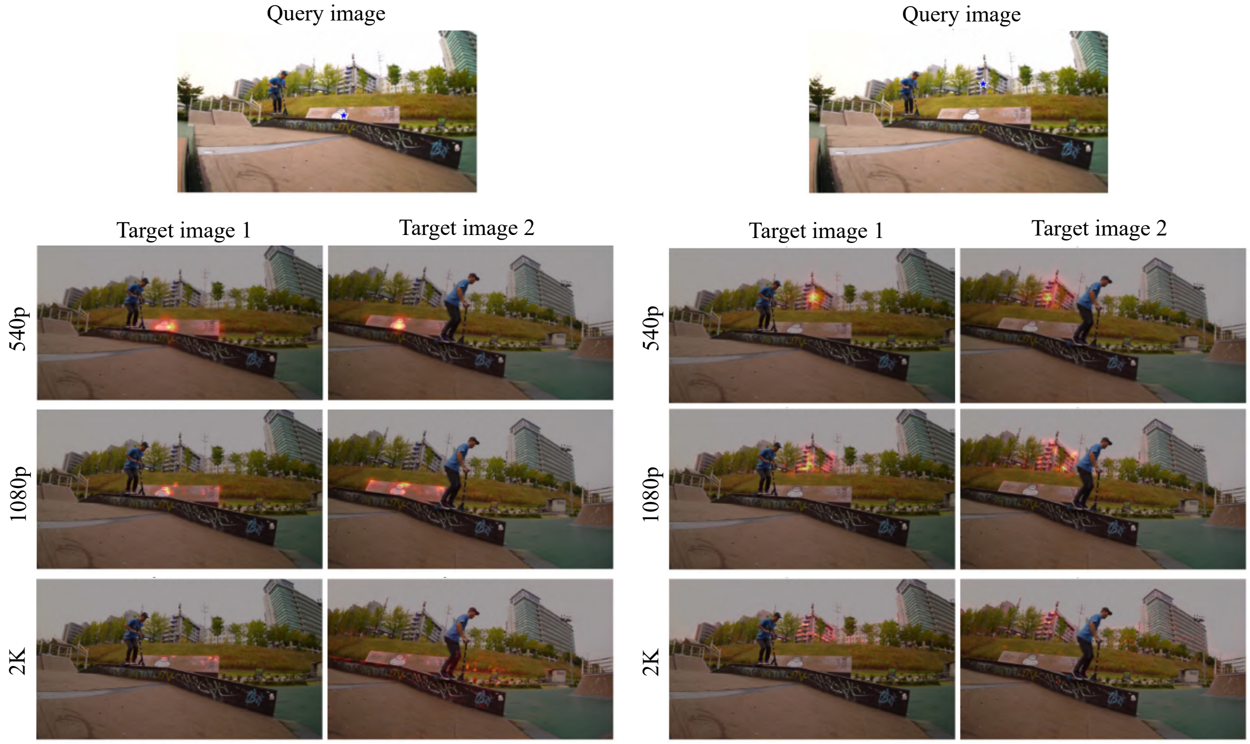


Figure 16. **Attention map** of the 15th global-attention layer of VGGT [95] at different input resolutions. The query token in the first image is marked with a blue star.

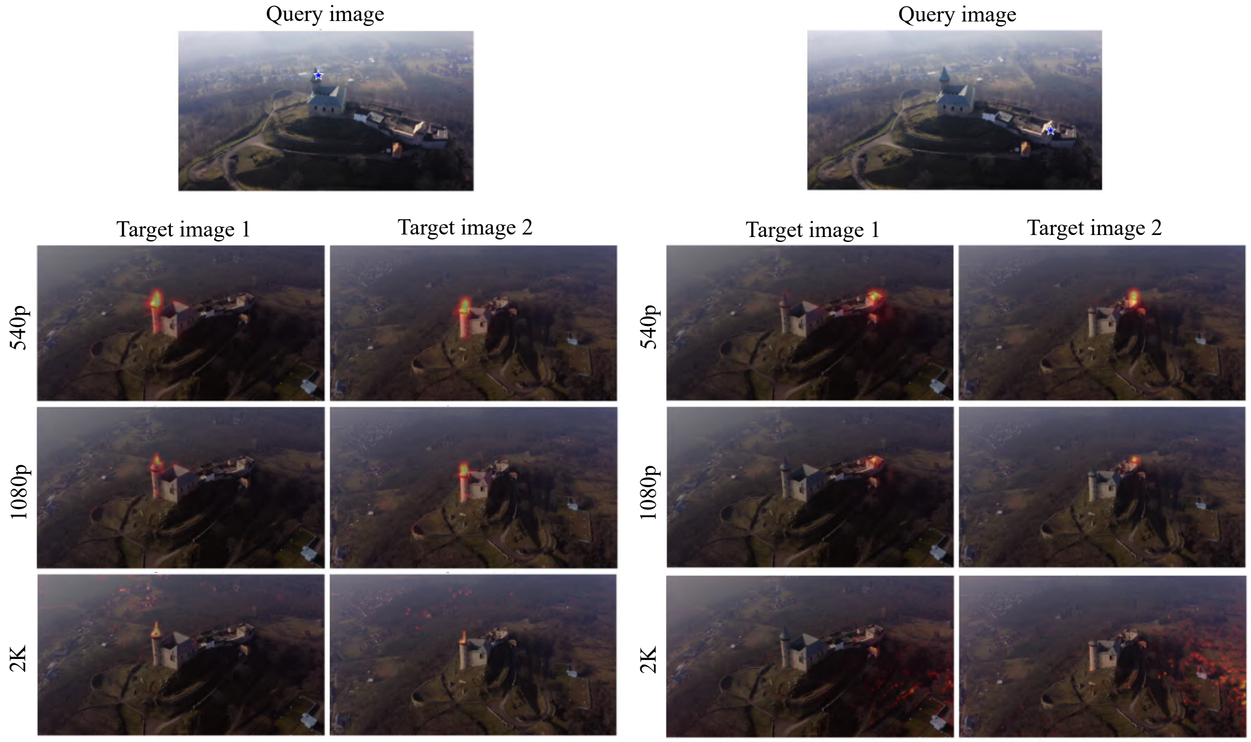


Figure 17. **Attention map** of the 15th global-attention layer of VGGT [95] at different input resolutions. The query token in the first image is marked with a blue star.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 3

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 7, 8, 2

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 1

[4] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4009–4018, 2021. 2

[5] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Muller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *ArXiv*, abs/2302.12288, 2023. 2

[6] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 1

[7] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3

[8] Aleksei Bochkovskii, AmaÃ**c**Gl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 2, 5, 6, 7, 3, 4

[9] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, 2012. 5, 6, 7, 8, 2, 3, 4

[10] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 1

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 1

[12] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595, 2023. 4, 5

[13] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22831–22840, 2025. 3

[14] Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. Leap-vo: Long-term effective any point tracking for visual odometry. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19844–19853, 2024. 3, 2

[15] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. *arXiv preprint arXiv:2503.24391*, 2025. 3

[16] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Seurat: From moving points to depth. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7211–7221, 2025. 2

[17] Gene Chou, Wenqi Xian, Guandao Yang, Mohamed Abdelfattah, Bharath Hariharan, Noah Snavely, Ning Yu, and Paul Debevec. Flashdepth: Real-time streaming video depth estimation at 2k resolution. *arXiv preprint arXiv:2504.07093*, 2025. 3

[18] Wenyan Cong, Yiqing Liang, Yancheng Zhang, Ziyi Yang, Yan Wang, Boris Ivanovic, Marco Pavone, Chen Chen, Zhangyang Wang, and Zhiwen Fan. E3d-bench: A benchmark for end-to-end 3d geometric foundation models. *arXiv preprint arXiv:2506.01933*, 2025. 2

[19] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Forty-first International Conference on Machine Learning*, 2024. 4

[20] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 5, 6, 7, 8, 1, 2, 3, 4

[21] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 1

[22] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 1

[23] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16739–16752, 2024. 5

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 2, 4

[25] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B

McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. Ieee, 2022. 3, 5

[26] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 2

[27] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arXiv:2504.13152*, 2025. 3

[28] Weilun Feng, Haotong Qin, Mingqiang Wu, Chuanguang Yang, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Yulun Zhang, Michele Magno, et al. Quantized visual geometry grounded transformer. *arXiv preprint arXiv:2509.21302*, 2025. 2

[29] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 2

[30] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *European Conference on Computer Vision*, pages 241–258. Springer, 2024. 2

[31] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan De Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 753–762. IEEE, 2025. 2

[32] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231 – 1237, 2013. 5, 6, 2, 3, 4

[33] Georgios Georgakis, Md. Alimoor Reza, Arsalan Mousavian, Phi Hung Le, and Jana Kosecka. Multiview rgb-d dataset for object instance detection. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 426–434, 2016. 5, 6, 2, 3, 4

[34] Jos'e L. G'omez, Manuel Silva, Antonio Seoane, Agnes Borr'as, Mario Noriega, Germ'an Ros, Jose A. Iglesias-Guitian, and Antonio M. L'opez. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *ArXiv*, abs/2312.12176, 2023. 5, 6, 7, 2, 3, 4

[35] Vitor Campanholo Guizilini, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2482–2491, 2019. 3, 5

[36] Vitor Campanholo Guizilini, Igor Vasiljevic, Di Chen, Rares Ambrus, and Adrien Gaidon. Towards zero-shot scale-aware monocular depth estimation. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9199–9209, 2023. 2

[37] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 2

[38] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3

[39] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007. 2

[40] Mu Hu, Wei Yin, China. Xiaoyan Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:10579–10596, 2024. 2

[41] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018. 1

[42] Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. *Advances in Neural Information Processing Systems*, 36:70847–70860, 2023. 5

[43] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, et al. On the importance of accurate geometry data for dense 3d vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–791, 2023. 3, 5

[44] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. 1

[45] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth Transfer: Depth Extraction from Video Using Non-Parametric Sampling . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 36(11):2144–2158, 2014. 2

[46] Bingxin Ke, Dominik Narnhofer, Shengyu Huang, Lei Ke, Torben Peters, Katerina Fragkiadaki, Anton Obukhov, and Konrad Schindler. Video depth without video models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7233–7243, 2024. 2, 3

[47] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9492–9502, 2024. 2

[48] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel López-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. Mapanything: Universal feed-forward metric 3d reconstruction. *ArXiv*, abs/2509.13414, 2025. 2, 3, 7, 8, 1, 4

[49] Numair Khan, Eric Penner, Douglas Lanman, and Lei Xiao. Temporally consistent online depth estimation using point-based fusion. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9119–9129, 2023. 2

[50] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2, 3, 5

[51] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 3

[52] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 3

[53] Zihang Lai and Andrea Vedaldi. Tracktention: Leveraging point tracking to attend videos faster and better. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22809–22819, 2025. 2, 3

[54] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 2

[55] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, 2024. 3

[56] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *ArXiv*, abs/2006.14616, 2020. 5

[57] Haodong Li, Chen Wang, Jiahui Lei, Kostas Daniilidis, and Lingjie Liu. Stereodiff: Stereo-diffusion synergy for video depth estimation. *arXiv preprint arXiv:2506.20756*, 2025. 2

[58] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 1

[59] Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10016–10025, 2024. 2

[60] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017. 1

[61] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22820–22830, 2025. 3

[62] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 3

[63] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2015. 5, 6, 7, 2, 3, 4

[64] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023. 1

[65] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021. 2

[66] T.D. Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. *ArXiv*, abs/2410.24211, 2024. 3

[67] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023. 4

[68] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2

[69] Duc-Hai Pham, Tung Do, Phong Nguyen, Binh-Son Hua, Khoi Nguyen, and Rang Nguyen. Sharpdepth: Sharpening metric depth predictions using diffusion distillation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17060–17069, 2025. 2, 7

[70] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10106–10116, 2024. 2

[71] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and

Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[72] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 2

[73] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021. 1

[74] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 1

[75] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[76] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. *Advances in neural information processing systems*, 18, 2005. 2

[77] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008. 2

[78] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3

[79] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017. 3, 5

[80] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025. 2

[81] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 7, 8, 2

[82] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 3, 5

[83] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 3

[84] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012. 7, 8, 2

[85] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021. 4

[86] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[87] Saksham Suri, Matthew Walmer, Kamal Gupta, and Abhinav Shrivastava. Lift: A surprisingly simple lightweight feature transform for dense vit descriptors. In *European Conference on Computer Vision*, 2024. 4

[88] Jeff Tan, Nikhil Varma Keetha, Yifei Liu, Shubham Tulsiani, and Deva Ramanan. Benchmarking stereo geometry estimation in the wild. 2025. 2

[89] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5283–5293, 2025. 2, 3

[90] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 3

[91] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8938–8948, 2021. 5, 6, 7, 2, 3, 4

[92] Igor Vasiljevic, Nicholas I. Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. Diode: A dense indoor and outdoor depth dataset. *ArXiv*, abs/1908.00463, 2019. 5, 6, 2, 3, 4

[93] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2022–2030, 2018. 2

[94] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster vggt with block-sparse global attention. *arXiv preprint arXiv:2509.07120*, 2025. 2, 5

[95] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 16

[96] Kaixuan Wang and Shaojie Shen. Flow-motion and depth

network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020. 1

[97] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 2, 3, 4, 5, 6, 7, 8, 1

[98] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5261–5271, 2025. 2, 5, 6, 1, 3, 4

[99] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 2, 4, 5, 6, 7, 8, 1, 3

[100] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3, 5

[101] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 1

[102] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 1, 2, 3, 4, 5, 6, 7, 8, 9, 14

[103] Thomas Wimmer, Prune Truong, Marie-Julie Rakotosaona, Michael Oechsle, Federico Tombari, Bernt Schiele, and Jan Eric Lenssen. Anyup: Universal feature upsampling. 2025. 8

[104] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 3

[105] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025. 3

[106] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22389, 2024. 1

[107] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *arXiv preprint arXiv:2510.07316*, 2025. 2, 4

[108] Tian-Xing Xu, Xiangjun Gao, Wenbo Hu, Xiaoyu Li, Song-Hai Zhang, and Ying Shan. Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors. *arXiv preprint arXiv:2504.01016*, 2025. 2, 5, 6, 7, 1, 3, 4, 8, 9

[109] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025. 2, 3, 7, 8

[110] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024. 2

[111] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 2

[112] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3

[113] David Yifan Yao, Albert J. Zhai, and Shenlong Wang. Uni4d: Unifying visual foundation models for 4d modeling from a single video. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1116–1126, 2025. 3

[114] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[115] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 1

[116] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 204–213, 2020. 2

[117] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9009–9019, 2023. 2

[118] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 3, 2

[119] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21936–21947, 2025. 2, 3, 7, 8

[120] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 1

[121] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *ArXiv*, abs/2507.11539, 2025. 3