Unlocking the Power of Critical Factors for 3D Visual Geometry Estimation

Guangkai Xu[1]*, Hua Geng[1]*, Yanlong Sun[2], Huanyi Zheng[1], Songyi Yin[1], Hao Chen[1], Chunhua Shen[1,3]†

[1] Zhejiang University, State Key Lab of CAD & CG [2] Tsinghua University [3] Ant Group

Abstract

*Feed-forward visual geometry estimation has recently made rapid progress. However, an important gap remains: multi-frame models usually produce better cross-frame consistency, yet they often underperform strong per-frame methods on single-frame accuracy. This observation motivates our systematic investigation into the critical factors driving model performance through rigorous ablation studies, which reveals several key insights: 1) Scaling up data diversity and quality unlocks further performance gains even in state-of-the-art visual geometry estimation methods; 2) Commonly adopted confidence-aware loss and gradient-based loss mechanisms may unintentionally hinder performance; 3) Joint supervision through both per-sequence and per-frame alignment improves results, while local region alignment surprisingly degrades performance. Furthermore, we introduce two enhancements to integrate the advantages of optimization-based methods and high-resolution inputs: a consistency loss function that enforces alignment between depth maps, camera parameters, and point maps, and an efficient architectural design that leverages high-resolution information. We integrate these designs into **CARVE**, a resolution-enhanced model for feed-forward visual geometry estimation. Experiments on point cloud reconstruction, video depth estimation, and camera pose/intrinsic estimation show that CARVE achieves strong and robust performance across diverse benchmarks.*

1. Introduction

Recovering accurate and consistent 3D attributes from monocular video, including 3D point clouds, camera parameters, and depth maps, remains a long-standing challenge with broad applications, such as autonomous driving [5, 76], virtual and augmented reality [34, 42], robotic navigation [32, 63], and medical imaging [7, 50].

Existing approaches can be roughly categorized into two groups: optimization-based approaches and learning-based approaches. Optimization-based techniques [9, 36, 40, 45] generally rely on robust feature matching to estimate 3D attributes by minimizing reprojection errors. These methods often produce sparse or semi-dense reconstructions due to the heavy dependence on reliable correspondences.

In contrast, learning-based approaches directly regress 3D attributes through end-to-end neural networks trained on large-scale labeled datasets, and can be broadly categorized into per-frame methods [2, 23, 59, 69] and multi-frame methods [28, 30, 56–58, 61, 64, 67]. Despite having access to cross-frame information, multi-frame methods do not consistently outperform per-frame methods. In practice, multi-frame methods mainly improve temporal consistency, while per-frame approaches often achieve higher accuracy on individual frames. This advantage is often attributed to carefully designed training objectives, high-resolution inputs, and well-structured training curriculum.

Motivated by these observations, we systematically investigate these key factors on a representative method VGGT [57] through extensive ablation studies, resulting in several critical insights: 1) Despite extensive pre-training on large datasets, scaling up data diversity and quality unlocks further performance gain. 2) The commonly adopted spatial gradient loss and confidence-aware weighting strategy can unexpectedly degrade the model performance. Conversely, employing fixed weighting inversely proportional to ground-truth depth consistently improves performance; 3) The sequence-level and frame-level alignment strategy of training objectives improves overall performance, whereas the local region alignment unexpectedly brings a performance reduction.

Alongside these insights, we explore two complementary enhancements to integrate the advantage of optimization-based methods and leverage high-resolution inputs. First, motivated by the geometric constraints of optimization-based techniques, we introduce a consistency loss enforcing strict consistency among estimated camera parameters, depth maps, and 3D point clouds. Second, rather than directly feeding high-resolution inputs, we propose to extract high-resolution and low-resolution ViT features, and fuse them via cross-attention equipped with zero-initialized gating parameters, which can preserve the pre-training knowledge.

*Both authors contributed equally. † Corresponding author.

Combining these insights and improvements, we scale up the training and propose **CARVE**, an accurate and resolution-enhanced visual geometry estimation model. The rigorous ablation study paves the way for better performance of video depth estimation, camera pose and intrinsics estimation, and 3D point cloud estimation on evaluation datasets including KITTI [18], 7-Scenes [48], TUM [51], HO3D [20], ETH3D [47], HAMMER [26], and Bonn [39]. Our main contributions are summarized as follows:

- Rigorous ablation experiments are conducted on a representative visual geometry method to explore how training objectives and data influence the performance.
- We propose a consistency loss to enforce geometric coherence among predicted camera parameters, depth maps, and 3D point clouds, thereby integrating intrinsic perspective projection constraints during training.
- We develop an efficient and effective feature fusion mechanism that integrates high-resolution features into low-resolution features via cross-attention, enabling accurate estimation with less computational burden.
- Integrating these insights and improvements, we propose the **CARVE** framework, achieving strong overall performance across diverse benchmark datasets, including point cloud reconstruction, video depth estimation, and camera pose/intrinsic estimation.

2. Related Work

2.1. Optimization-based Reconstruction

Traditional Structure-from-Motion (SfM) [9, 36, 40, 45], Multi-View Stereo (MVS) [17, 19, 46], and visual SLAM methods [13, 14, 37], rely heavily on a multi-stage pipeline of feature extraction, matching, and optimization. They jointly estimate camera poses and per-pixel 3D geometry by minimizing the reprojection error. However, the geometric fidelity achieved by these optimization techniques is critically dependent on the accuracy and robustness of the initial feature correspondences. To enhance robustness, subsequent works [12, 24, 29, 53, 70] integrate learned features or correspondences into the optimization framework to assist or replace traditional modules.

2.2. Per-frame Reconstruction

One approach to 3D visual geometry estimation from monocular video is to first estimate per-frame geometry and then enforce multi-view consistency. Numerous studies have investigated monocular depth estimation, demonstrating notable progress and improved accuracy [2, 16, 22, 23, 27, 66, 68, 69, 73, 74]. Several of them [2, 74] can produce 3D reconstruction from a single image. The MoGe series [59, 60] directly estimates dense affine-invariant point clouds. To ensure consistency between frames, these methods still rely on alignment with matching information [44, 52] or consistency optimization [65].

2.3. Multi-frame Reconstruction

Moving beyond explicit matching cost volume, feed-forward geometry methods have emerged after matching-based stereo approaches [4, 8, 31]. DUSt3R [61] and its follow-ups [25, 28, 75] directly regress point clouds from input images using neural networks. Extending this paradigm to longer temporal contexts, Spann3R [56] and CUT3R [58] model multi-frame geometry with implicit scene representations. More recently, Fast3R [67], VGGT [57], and Pi3 [64] adopt feed-forward architectures to estimate multi-view geometry while reducing reliance on explicit correspondence matching and iterative optimization. In particular, Pi3 removes the need for a fixed reference view through a permutation-equivariant design, while Depth Anything 3 [30] further generalizes visual geometry estimation with a simple transformer backbone and a unified depth-ray prediction target.

3. Method

In this section, we first introduce the multi-frame baseline and then systematically investigate the gap between representative multi-frame and per-frame methods.

3.1. Preliminaries

Problem Definition. The representative visual geometry estimation method [57] takes a set of images $\mathbf{I} \in \mathbb{R}^{T\times H\times W\times 3}$ as input, and produces depth maps $\hat{\mathbf{D}} \in \mathbb{R}^{T\times H\times W}$, world coordinates point maps $\hat{\mathbf{P}} \in \mathbb{R}^{T\times H\times W\times 3}$, and camera parameters $\hat{\mathbf{g}} \in \mathbb{R}^{T\times 9}$, which are composed of quaternion $\hat{\mathbf{r}} \in \mathbb{R}^{T\times 4}$, translation vectors $\hat{\mathbf{t}} \in \mathbb{R}^{T\times 3}$, and field of view angles $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{T\times 2}$. The point tracking task is not discussed here.

Network Architecture. The network patchifies the input images $\mathbf{I}$ into tokens $\hat{\mathbf{f}}_{\text{img}} \in \mathbb{R}^{T\times P\times C}$ with DINOv2 [38] encoder and then passes them along with learnable camera tokens $\mathbf{f}_{\text{cam\_init}}$ into transformer blocks and decoders.

$$\begin{gathered}\hat{\mathbf{f}}_{\text{img}} = \text{Encoder}(\mathbf{I}),\ (\hat{\mathbf{f}}_{\text{geo}}, \hat{\mathbf{f}}_{\text{cam}}) = \text{Transformer}(\hat{\mathbf{f}}_{\text{img}}, \mathbf{f}_{\text{cam\_init}}),\\ \hat{\mathbf{D}} = \text{Head}_{\text{depth}}(\hat{\mathbf{f}}_{\text{geo}}),\ \ \hat{\mathbf{P}} = \text{Head}_{\text{point}}(\hat{\mathbf{f}}_{\text{geo}})\\ \hat{\mathbf{g}} = [\hat{\mathbf{t}}, \hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}] = \text{Head}_{\text{cam}}(\hat{\mathbf{f}}_{\text{cam}}).\end{gathered} \tag{1}$$

Training Objectives. The training losses contain three types of functions. For regression loss, they filter out invalid regions and simply supervise the valid regions:

$$\mathcal{L}_{\text{reg}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\xi}, \mathbf{W}) = \mathbb{E}_{p\in\mathcal{M}} \left\| \mathbf{W}_p \cdot (\hat{\boldsymbol{\xi}}_p - \boldsymbol{\xi}_p) \right\|, \tag{2}$$

where $\hat{\boldsymbol{\xi}}$ and $\boldsymbol{\xi}$ are the prediction and the ground truth of either depth maps or point maps. $\mathcal{M}$ represents the valid

region, and $\mathbf{W}$ means the weight map. For spatial gradient loss, it supervises the difference between nearby pixels:

$$\mathcal{L}_{\text{sg}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\xi}, \mathbf{W}) = \mathbb{E}_{p\in\mathcal{M}} \left\| \mathbf{W}_p \cdot (\nabla_p \hat{\boldsymbol{\xi}}_p - \nabla_p \boldsymbol{\xi}_p) \right\|, \quad (3)$$

where ∇_p represents the difference between nearby pixels of the spatial x and y axes. Another confidence loss is adopted to supervise the learnable confidence map:

$$\mathcal{L}_{\text{conf}}(\mathbf{W}) = \mathbb{E}_{p\in\mathcal{M}} \left| -\alpha \log \mathbf{W}_p \right|. \quad (4)$$

The overall training losses consist of three components:

$$\begin{gathered}\mathcal{L} = \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{point}}, \mathcal{L}_{\text{cam}} = \mathbb{E}_t \left\| \hat{\mathbf{g}}_t - \mathbf{g}_t \right\|, \\ \mathcal{L}_{\text{depth}} = \mathcal{L}_{\text{reg}}(\hat{\mathbf{D}}, \mathbf{D}, \boldsymbol{\Sigma}^{\text{d}}) + \mathcal{L}_{\text{sg}}(\hat{\mathbf{D}}, \mathbf{D}, \boldsymbol{\Sigma}^{\text{d}}) + \mathcal{L}_{\text{conf}}(\boldsymbol{\Sigma}^{\text{d}}), \\ \mathcal{L}_{\text{point}} = \mathcal{L}_{\text{reg}}(\hat{\mathbf{P}}, \mathbf{P}, \boldsymbol{\Sigma}^{\text{p}}) + \mathcal{L}_{\text{sg}}(\hat{\mathbf{P}}, \mathbf{P}, \boldsymbol{\Sigma}^{\text{p}}) + \mathcal{L}_{\text{conf}}(\boldsymbol{\Sigma}^{\text{p}}) \end{gathered} \quad (5)$$

where the confidence maps $\boldsymbol{\Sigma}^{\text{d}}$ and $\boldsymbol{\Sigma}^{\text{p}}$ are learned automatically, serving as adaptive weights for the loss functions.

3.2. Effectiveness of Training Components

Compared with per-frame methods [2, 23, 59, 69], multi-frame visual geometry estimation methods [28, 56–58, 61, 67] achieve better multi-frame consistency but lower per-frame accuracy. We conduct extensive ablations to investigate the reasons.

Experimental Details. By default, we initialize the model with VGGT [57] pretrained weights, freeze the ViT feature extractor, and train the remaining components. The predicted point cloud, depth map, and camera translation are aligned to ground truth via a per-sequence scale factor before loss computation. Training uses a dynamic batch size (up to 24 frames) for 30K iterations, and evaluation is conducted on uniformly sampled keyframes with up to 200 frames per video (see supplementary for details). For data ablation, compared to the original VGGT, only the training data is varied; for loss ablation, we fix "Data3" and evaluate different loss terms; and for resolution ablation, we use "Data3" with "Our Loss".

Training Data. We progressively expand the training data from "Data1" to "Data3," with the composition summarized in Table 2. Specifically, "Data1" consists solely of high-quality datasets, "Data2" introduces greater data diversity while maintaining quality, and "Data3" further incorporates noisy datasets. As shown in Table 1, performance improves consistently with data scaling, suggesting that current visual geometry estimation models can still benefit from larger and more diverse training data.

> ***Insight* 1.** Scaling up data diversity and quality unlocks further performance gains in SOTA visual geometry estimation.

Training Objective. Using "Data3", we ablate the original loss by removing $\mathcal{L}_{\text{sg}}$ and $\mathcal{L}_{\text{conf}}$. As shown in Table 1, removing either term improves performance. $\mathcal{L}_{\text{sg}}$ appears to focus excessively on local region variance, resulting in lower overall accuracy. Regarding ($\mathcal{L}_{\text{conf}}$), the model can find a shortcut: instead of learning difficult regions, it can reduce the overall loss by decreasing the learnable loss weights of these areas. In contrast, using the inverse of depth values as a fixed weight map [59] provides a natural alternative. This strategy focuses on the relatively close areas, and achieves further performance improvement. ("$\mathcal{L}_{\text{reg}}$ + $\mathcal{L}_{\text{conf}}$" vs. "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}})$"). Besides $\mathcal{L}_{\text{sg}}$, we observe that the temporal gradient loss $\mathcal{L}_{\text{tg}}$ [6] also negatively impacts performance. ("$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}})$" vs. "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}})$ + $\mathcal{L}_{\text{tg}}$ ").

$$\mathcal{L}_{\text{tg}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\xi}, \mathbf{W}) = \mathbb{E}_{t\in\mathcal{T}, p\in\mathcal{M}} \left\| \mathbf{W}_{t,p} \cdot (\nabla_t \hat{\boldsymbol{\xi}}_{t,p} - \nabla_t \boldsymbol{\xi}_{t,p}) \right\|, \quad (6)$$

where ∇_t represents the temporal difference operation.

> ***Insight* 2.** For the training objectives, gradient-based loss functions and learnable confidence weight maps unexpectedly lead to performance degradation. Instead, adopting a simple inverse depth weight map demonstrates superior effectiveness.

To further explore the gap between multi-frame and single-frame methods, we conduct an additional ablation study on key components of the single-frame method. For the alignment strategy in training loss, rather than applying a global scale to align the entire sequence with the ground truth, MoGe [59] applies separate scale-shift alignment for each frame (L_{F}) and each sampled local 3D spherical region of the point cloud (L_{S}).

$$\begin{gathered}\mathcal{L}_{\text{F}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\xi}, \mathbf{W}) = \mathbb{E}_{t\in\mathcal{T}, p\in\mathcal{M}} \left\| \mathbf{W}_{t,p} \cdot (\mathbf{a}_t \cdot \hat{\boldsymbol{\xi}}_{t,p} + \mathbf{B}_t - \boldsymbol{\xi}_{t,p}) \right\|, \\ \mathcal{L}_{\text{S}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\xi}, \mathbf{W}) = \mathbb{E}_{\mathcal{S}_j\in\mathcal{S}, p\in\mathcal{S}_j} \left\| \mathbf{W}_{j,p} \cdot (\mathbf{a}_j \cdot \hat{\boldsymbol{\xi}}_{j,p} + \mathbf{B}_j - \boldsymbol{\xi}_{j,p}) \right\|, \\ \mathcal{S}_j = \{p \mid \|\mathbf{P}_p - \mathbf{P}_j\| \le r_j, p \in \mathcal{M}\}, \end{gathered} \quad (7)$$

where $\mathbf{a}_t$ and $\mathbf{B}_t$ are the scale and shift alignment parameters for each frame t, and $\mathbf{a}_j$ and $\mathbf{B}_j$ are the parameters for each sampled local 3D spherical region $\mathcal{S}_j$, with 3D region radius r_j. All these scale-shift parameters are computed with ROE alignment [59]. As presented in Table 1, we observe that supervising with both per-sequence and per-frame alignment improves performance ("$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}})$" vs. "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}}$"), while the local region alignment unexpectedly results in a decrease in performance ("$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}}$" vs. "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}} + \mathcal{L}_{\text{S}}$").

Moreover, we observe that the predicted depth map, camera parameters, and point cloud do not consistently align with the geometry projection constraint from 2D to 3D. One potential approach is to filter out the inaccurate

Table 1. Ablation study for training data, training loss, and high-resolution input. For training loss, we explore the effectiveness of the original VGGT losses ($\mathcal{L}_{\text{sg}}$, $\mathcal{L}_{\text{conf}}$), several losses adopted in state-of-the-art methods ($\mathcal{L}_{\text{tg}}$, $\mathcal{L}_{\text{F}}$, $\mathcal{L}_{\text{S}}$), and our proposed consistency loss ($\mathcal{L}_{\text{consis}}$). "Rank" represents the average rank value across all metrics. Rows in gray denote training with up to 12 frames (vs. the usual 24) and evaluation with up to 100 frames (vs. the usual 200) due to GPU memory constraints.

Method	7-Scenes			Bonn			KITTI			TUM			Rank↓
	Recon C-L1↓	Pose ATE↓	Depth Rel↓	Recon C-L1↓	Pose ATE↓	Depth Rel↓	Recon C-L1↓	Pose ATE↓	Depth Rel↓	Recon C-L1↓	Pose ATE↓	Depth Rel↓	
VGGT baseline	0.049	0.073	0.069	0.057	0.075	0.054	0.296	1.113	0.094	0.051	0.047	0.062	-
Data1	0.056	0.079	0.070	0.051	0.064	0.049	0.281	1.411	0.085	0.040	0.090	0.049	2.50
Data2	0.052	0.078	0.069	0.051	0.071	0.052	0.277	1.267	0.083	0.040	0.090	0.052	2.25
(Our Data) Data3	0.049	0.065	0.065	0.048	0.055	0.046	0.263	0.937	0.082	0.038	0.050	0.042	1.00
(VGGT Loss) $\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{sg}}$	0.049	0.065	0.065	0.048	0.055	0.046	0.263	0.937	0.082	0.038	0.050	0.042	2.08
$\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{conf}}$	0.050	0.064	0.065	0.043	0.046	0.046	0.270	1.059	0.082	0.038	0.050	0.043	2.00
$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}})$	0.043	0.066	0.062	0.044	0.050	0.046	0.254	0.866	0.079	0.036	0.039	0.039	1.33
$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}})$	0.043	0.066	0.062	0.044	0.050	0.046	0.254	0.866	0.079	0.036	0.039	0.039	2.42
$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{sg}}$	0.045	0.066	0.063	0.045	0.048	0.048	0.270	0.949	0.082	0.038	0.039	0.041	4.17
$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{tg}}$	0.045	0.067	0.063	0.046	0.050	0.048	0.263	1.270	0.081	0.039	0.079	0.042	5.17
$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}}$	0.042	0.065	0.061	0.044	0.050	0.045	0.245	1.042	0.078	0.037	0.050	0.041	2.33
$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}} + \mathcal{L}_{\text{S}}$	0.043	0.068	0.061	0.047	0.055	0.044	0.255	0.901	0.080	0.037	0.036	0.040	3.08
(Our Loss) $\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}} + \mathcal{L}_{\text{consis}}$	0.043	0.065	0.061	0.042	0.045	0.045	0.249	0.919	0.077	0.037	0.041	0.041	1.92
w/o VGGT High Resolution	0.043	0.065	0.061	0.042	0.045	0.045	0.249	0.919	0.077	0.037	0.041	0.041	1.42
(Ours) w/ Efficient High Resolution	0.043	0.068	0.061	0.038	0.042	0.046	0.238	0.964	0.080	0.031	0.035	0.041	1.33
w/ VGGT High Resolution	0.056	0.081	0.067	0.057	0.036	0.045	0.237	0.355	0.064	0.039	0.060	0.050	1.75
(Ours) w/ Efficient High Resolution	0.058	0.059	0.061	0.037	0.029	0.046	0.235	0.289	0.071	0.034	0.024	0.042	1.25

Table 2. The training data components for ablation study. From "Data1" to "Data3", the data volume increases progressively. While "Data1" and "Data2" contain only high-quality data, "Data3" incorporates additional noisy data.

Name	Training Data Components
Data1	Hypersim [43], ScanNet++ [72], Virtual KITTI2 [3], MVS-Synth [24], Spring [35], UnrealStereo4K [55]
Data2	"Data1", Tartanair [62], Parallel Domain [54], TartanGround [41]
Data3	"Data2", ScanNet [10], ARKitScenes [1], GraspNet [15], BlendedMVS [71]

Table 3. The parameter count (Params.) and frames per second (FPS) of VGGT and CARVE tested on a single NVIDIA H200. The number of parameters is reported in millions, and the FPS is measured with sequence length 32, averaged over 100 runs after warm-up.

Method	Params. (M)	Image Resolution	FPS
VGGT [57]	1189.01	518 × 518	24.85
		1036 × 1036	2.54
CARVE (Ours)	1214.21	1036 × 1036	15.26

areas by assessing the inconsistencies. In contrast, we propose integrating this inherent geometry constraint directly into the training framework, rather than treating it as a post-processing step. Specifically, we introduce a consistency loss $\mathcal{L}_{\text{consis}}$ to enforce the alignment between the estimated point cloud and its unprojected counterpart.

$$\begin{aligned}
\mathcal{L}_{\text{consis}}(\hat{\mathbf{P}}, \hat{\mathbf{D}}, \hat{\mathbf{r}}, \hat{\mathbf{t}}, \hat{\boldsymbol{\theta}}) &= \mathbb{E}_{p\in\mathcal{M}} \left| \hat{\mathbf{P}}_{\text{unproj}}(p) - \hat{\mathbf{P}}(p) \right|, \\
\hat{\mathbf{P}}_{\text{unproj}}(p) &= \hat{\mathbf{R}}(\hat{\mathbf{D}}(p)\hat{\mathbf{K}}^{-1}p) + \hat{\mathbf{t}},\ \hat{\mathbf{R}} = \mathcal{H}(\hat{\mathbf{r}}), \\
\hat{\mathbf{K}} &= \text{Intrinsics}(\hat{f}_x, \hat{f}_y, \hat{c}_x, \hat{c}_y),\ \hat{c}_x = W/2, \hat{c}_y = H/2 \\
\hat{f}_x &= \frac{W}{2\tan\left(\hat{\boldsymbol{\theta}}_x/2\right)},\ \hat{f}_y = \frac{H}{2\tan\left(\hat{\boldsymbol{\theta}}_y/2\right)},
\end{aligned} \tag{8}$$

where $\text{Intrinsics}(\cdot,\cdot,\cdot,\cdot)$ computes 3×3 camera intrinsic matrix from the focal length and the optical center, and $\mathcal{H}(\cdot)$ transforms a rotation quaternion to a 3×3 camera rotation matrix. As shown in Table 1, the comparison between "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}}$" and "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}} + \mathcal{L}_{\text{consis}}$" demonstrates that enforcing consistency can lead to improved robustness and accuracy.

***Insight* 3.** 1) Supervision based on both per-sequence and per-frame alignment enhances the results, while local region alignment unexpectedly leads to a decrease in performance. 2) Enforcing consistency between the estimated point cloud and the unprojected one yields improvement.

Efficient High-Resolution Adaptation. It is well recognized that higher-resolution inputs typically enhance the performance of computer vision tasks. However, for the attention module of the transformer block, directly upsampling the input image by a factor of 2 theoretically results in 4× tokens and 16× computational complexity. In practice, we report TFLOPs, GPU memory usage, and FPS of VGGT under both low- and high-resolution input settings in Table 4 and Table 3. Despite the adoption of several engineering optimizations (see the supplementary material for details), high-resolution input still results in 4× TFLOPs, 3× to 4× GPU memory usage, and 0.1× FPS.

In contrast, we propose an efficient high-resolution adaptation network as illustrated in Figure 1. We extract the

Table 4. Efficiency metrics for different input frames and resolutions on a single NVIDIA H200. The number of floating point operations is measured in teraFLOPs (TFLOPs), and memory is measured in gibibytes (GiB). "(H× W)" represents the input image resolution.

	VGGT [57] (518 × 518)		VGGT [57] (1036 × 1036)		CARVE (Ours, 1036 × 1036)	
# Frames	TFLOPs	Peak GPU Mem (GiB)	TFLOPs	Peak GPU Mem (GiB)	TFLOPs	Peak GPU Mem (GiB)
8	25.57	8.81	101.99	21.80	52.97	9.08
16	51.14	10.99	203.98	30.49	105.93	11.40
32	102.28	15.36	407.97	47.89	211.87	16.05
64	204.56	25.61	815.93	88.81	423.73	25.71
128	409.13	46.75	OOM	OOM	847.47	46.86
256	818.25	89.02	OOM	OOM	1694.94	89.14

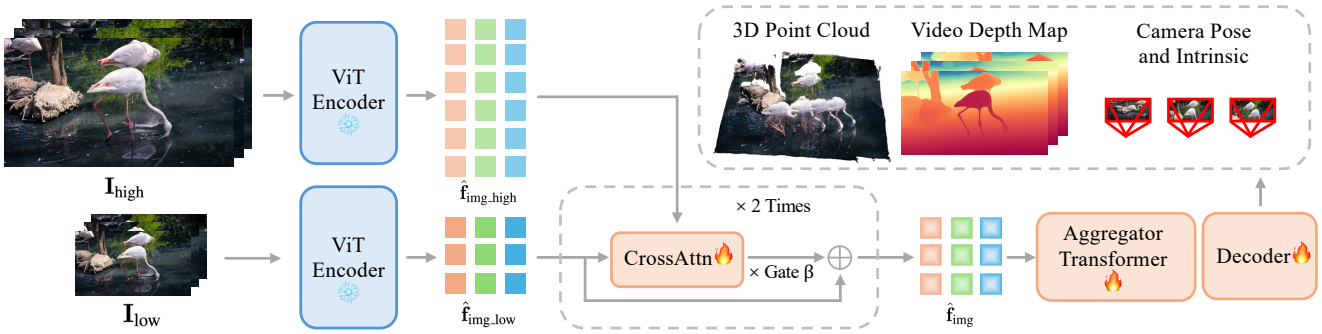

Figure 1. Network architecture of our proposed CARVE model. We extract the high-resolution feature and fuse it into the low-resolution main branch with frame-wise cross attention modules and zero-initialized residual gate parameters β.

high-resolution feature and fuse it to the low-resolution main branch before sending it to the transformer block with frame-wise cross attention modules. The low-resolution image serves as the query, while the high-resolution image serves as the key and value. Similar to frame-wise attention, the cross-attention is computed between low- and high-resolution image pairs of the same frame. To prevent the pretrained parameters from being degraded, inspired by ResNet [21], we treat the cross-attention outputs as a residual branch, which is added to the main branch for each cross-attention block after being scaled by a learnable gating parameter. These gating parameters are initialized to zero. For the depth head and point head, we simply upsample the feature prior to the last few convolution layers. The formulation is as follows.

$$\begin{aligned} \hat{\mathbf{f}}_{\text{img\_low}} &= \text{Encoder}(\mathbf{I}_{\text{low}}),\ \hat{\mathbf{f}}_{\text{img\_high}} = \text{Encoder}(\mathbf{I}_{\text{high}}), \\ \hat{\mathbf{f}}_{\text{img}} &= \hat{\mathbf{f}}_{\text{img\_low}} + \beta \cdot \text{CrossAttn}(\hat{\mathbf{f}}_{\text{img\_low}}, \hat{\mathbf{f}}_{\text{img\_high}}), \\ (\hat{\mathbf{f}}_{\text{geo}}, \hat{\mathbf{f}}_{\text{cam}}) &= \text{Transformer}(\hat{\mathbf{f}}_{\text{img}}, \mathbf{f}_{\text{cam\_init}}), \end{aligned} \tag{9}$$

where the feature $\hat{\mathbf{f}}_{\text{img\_low}}$ and $\hat{\mathbf{f}}_{\text{img\_high}}$ are extracted separately from the low-resolution image $\mathbf{I}_{\text{low}}$ and high-resolution image $\mathbf{I}_{\text{high}}$, and β is the learnable gate parameter with zero initialization. The cross-attention block $\text{CrossAttn}(\cdot, \cdot)$ takes $\hat{\mathbf{f}}_{\text{img\_low}}$ as query and $\hat{\mathbf{f}}_{\text{img\_high}}$ as key and value. The fused feature $\hat{\mathbf{f}}_{\text{img}}$ shares the same dimensionality as $\hat{\mathbf{f}}_{\text{img\_low}}$, allowing it to seamlessly replace the original low-resolution feature in subsequent modules. As demonstrated in Table 1, our architecture enhances the overall performance ("w/o High Resolution" vs. "w/ Efficient High Resolution"). Furthermore, our proposed efficient high-resolution architecture even outperforms the direct input upsampling strategy ("w/ Efficient High Resolution" vs. "w/ VGGT High Resolution" in gray color, the evaluations are conducted with a maximum of 100 frames due to GPU memory constraints.). We hypothesize that this improvement arises from two factors: 1) Our efficient architecture processes both high- and low-resolution images, where the integration of multi-resolution features proves beneficial. 2) High-resolution inputs may conflict with the original pretrained weights, which were learned from low-resolution data. For efficiency metrics, our proposed high-resolution architecture achieves substantial computational efficiency, requiring only 0.3× to 0.4× GPU memory, 0.5× TFLOPs, and delivering up to 6× higher FPS during inference, as reported in Table 3.

***Insight* 4.** Leveraging an efficient high-resolution architecture, the network demonstrates a superior balance between performance and efficiency.

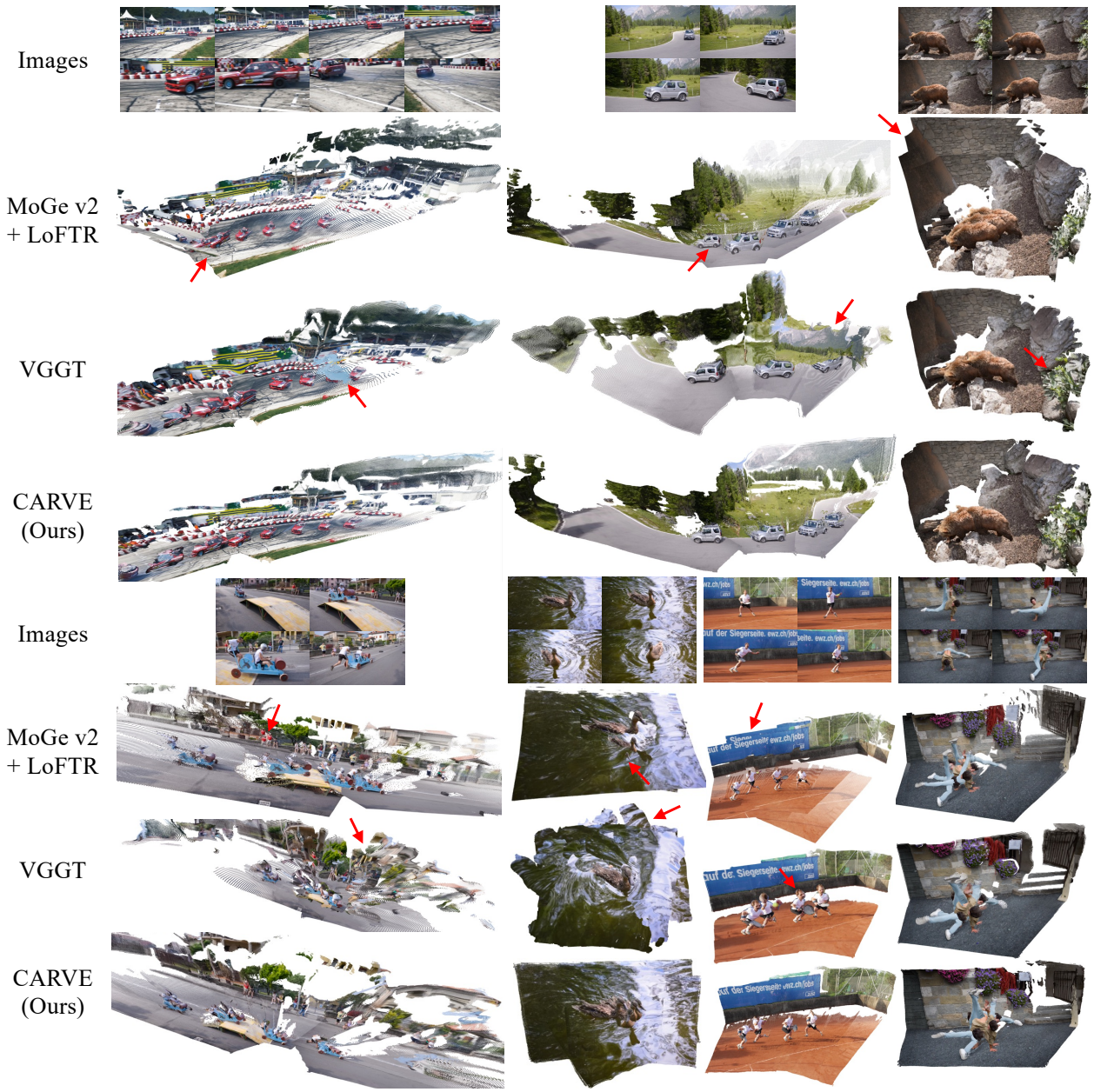

Figure 2. Qualitative results of point cloud estimation on in-the-wild images. The red arrows highlight instances of failed estimations, including incorrect camera pose estimation, abnormal geometry scaling, and inconsistencies between frames.

Table 5. Quantitative results of point cloud estimation on KITTI, 7-Scenes, and TUM.

	KITTI				7-Scenes				TUM				Rank↓
Method	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	
MoGe v2 + LoFTR	0.726	0.142	0.562	0.750	0.161	0.242	0.777	0.950	0.221	0.199	0.696	0.844	4.67
Spann3R	2.359	0.044	0.296	0.452	0.101	0.375	0.922	0.987	0.122	0.498	0.860	0.949	4.58
Fast3R	4.974	0.088	0.357	0.501	0.655	0.045	0.226	0.422	0.936	0.028	0.153	0.261	5.75
VGGT	0.296	0.220	0.688	0.842	0.049	0.660	0.988	0.997	0.051	0.712	0.980	0.993	2.75
Pi3	0.273	0.273	0.749	0.879	0.049	0.662	0.991	0.997	0.032	0.834	0.993	0.998	1.67
CARVE (Ours)	0.238	0.257	0.767	0.892	0.043	0.720	0.986	0.998	0.029	0.861	0.991	0.997	1.42

Table 6. Quantitative results of point cloud estimation on HAMMER, Bonn, and ETH3D.

Method	HAMMER				Bonn				ETH3D				Rank↓
	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	
MoGe v2 + LoFTR	0.030	0.872	1.000	1.000	0.174	0.226	0.765	0.941	1.889	0.002	0.026	0.066	3.92
Spann3R	0.041	0.727	1.000	1.000	0.114	0.354	0.894	0.978	2.479	0.067	0.216	0.366	3.75
Fast3R	0.062	0.488	1.000	1.000	0.983	0.019	0.134	0.285	3.901	0.000	0.000	0.000	5.17
VGGT	0.035	0.828	0.999	1.000	0.057	0.645	0.972	0.987	0.202	0.410	0.787	0.915	3.00
Pi3	0.013	0.997	1.000	1.000	0.031	0.796	0.998	1.000	0.106	0.433	0.896	0.971	1.17
CARVE (Ours)	0.012	0.999	1.000	1.000	0.043	0.720	0.986	0.998	0.236	0.423	0.765	0.867	1.92

Table 7. Quantitative results of video depth estimation on seven datasets.

Method	KITTI		7-Scenes		TUM		HO3D		HAMMER		Bonn		ETH3D		Rank↓
	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	
MoGe v2 + LoFTR	0.453	0.430	0.217	0.675	0.225	0.593	0.278	0.811	0.036	0.997	0.171	0.797	0.242	0.690	3.79
Fast3R	0.254	0.645	0.312	0.476	0.377	0.446	0.524	0.587	0.135	0.838	0.339	0.551	0.568	0.335	4.86
VGGT	0.094	0.917	0.069	0.930	0.062	0.954	0.270	0.755	0.046	0.968	0.054	0.953	0.043	0.978	3.21
Pi3	0.078	0.939	0.064	0.938	0.043	0.977	0.248	0.846	0.033	0.984	0.026	0.987	0.023	0.998	1.57
CARVE (Ours)	0.082	0.933	0.062	0.940	0.040	0.976	0.220	0.869	0.020	0.996	0.041	0.959	0.023	0.997	1.50

Table 8. Quantitative evaluation of camera pose and intrinsics on KITTI, 7-Scenes, TUM, and HO3D.

Method	KITTI				7-Scenes				TUM				HO3D	Rank↓
	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	
MoGe v2	0.162	–	–	–	0.192	–	–	–	0.124	–	–	–	0.067	4.50
Fast3R	0.079	106.082	0.161	125.443	0.075	1.696	1.056	2.576	0.028	1.189	1.257	1.897	0.012	3.38
VGGT	0.084	1.113	0.015	2.177	0.076	0.073	0.062	0.117	0.020	0.047	0.038	0.063	0.109	2.69
Pi3	0.094	0.572	0.016	2.270	0.036	0.058	0.059	0.103	0.045	0.046	0.034	0.071	0.082	2.31
CARVE (Ours)	0.078	0.664	0.016	1.740	0.024	0.052	0.064	0.104	0.049	0.041	0.032	0.060	0.039	1.69

Table 9. Quantitative evaluation of camera pose and intrinsics on HAMMER, Bonn, and ETH3D.

Method	HAMMER				Bonn				ETH3D				Rank↓
	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	
MoGe v2	0.084	–	–	–	0.136	–	–	–	0.058	–	–	–	4.67
Fast3R	0.062	0.119	0.166	0.187	0.025	0.669	0.722	1.089	0.075	13.074	1.491	16.466	3.83
VGGT	0.040	0.001	0.003	0.002	0.040	0.075	0.042	0.091	0.020	1.804	0.021	2.143	2.25
Pi3	0.082	0.003	0.005	0.006	0.022	0.039	0.024	0.055	0.031	0.140	0.021	0.193	1.92
CARVE (Ours)	0.035	0.001	0.004	0.003	0.028	0.044	0.029	0.056	0.018	0.184	0.022	0.223	1.92

Table 10. Quantitative results of monocular depth estimation on seven datasets.

Method	KITTI		7-Scenes		TUM		HO3D		HAMMER		Bonn		ETH3D		Rank↓
	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	Rel↓	δ↑	
MoGe	0.094	0.904	0.070	0.938	0.055	0.966	0.282	0.788	0.028	0.988	0.034	0.986	0.035	0.988	2.43
MoGe v2	0.098	0.908	0.077	0.932	0.057	0.964	0.256	0.837	0.023	0.996	0.037	0.984	0.036	0.986	2.79
Fast3R	0.274	0.594	0.247	0.591	0.285	0.597	0.550	0.556	0.143	0.813	0.230	0.646	0.404	0.527	6.00
VGGT	0.125	0.855	0.070	0.934	0.062	0.948	0.269	0.775	0.054	0.972	0.042	0.975	0.043	0.974	4.64
Pi3	0.112	0.878	0.068	0.941	0.055	0.963	0.271	0.819	0.040	0.986	0.033	0.983	0.034	0.986	2.93
CARVE (Ours)	0.106	0.885	0.066	0.941	0.049	0.969	0.236	0.851	0.028	0.994	0.035	0.985	0.033	0.985	1.86

4. Experiments

We scale up the training process with the losses of "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}} + \mathcal{L}_{\text{consis}}$", training data of "Data3", and our proposed efficient high-resolution architecture. The model is initialized with the VGGT [57] pretrained weights for common parameters. More training and evaluation details are provided in the supplementary material. We evaluate visual geometry estimation across multiple datasets, including KITTI [18], 7-Scenes [48], HO3D [20], TUM [51], ETH3D [47], HAMMER [26], and Bonn [39].

4.1. Point Cloud Estimation

For point cloud evaluation, we align the stacked point cloud with the corresponding stacked ground-truth one using a similarity transformation comprising a scale factor, a rotation matrix, and a translation vector. We report the Chamfer L1 distance (C-L1) and the F-score at thresholds of 5cm (F@5), 25cm (F@25), and 50cm (F@50). The estimated and ground-truth point clouds are downsampled using a voxel size of 2cm for fast evaluation.

We compare with the monocular reconstruction model MoGe v2 [60] ("MoGe v2 + LoFTR"), multi-view reconstruction model Spann3R [56], Fast3R[67], VGGT [57], and Pi3 [64]. We use LoFTR [52] for feature extraction and matching, and compute the similarity transformation between frames using the matched points to achieve alignment of the results from MoGe v2 [60].

Quantitative comparisons are shown in Table 5 and Table 6. As observed, "MoGe v2 + LoFTR" performs well on datasets with limited viewpoint variation, such as HAMMER. However, its reliance on accurate matching information limits its effectiveness in more complex scenarios. VGGT outperforms other multi-view methods, including Spann3R, and Fast3R, which can be attributed to its temporally scalable network framework that improves its ability to model long-range dependencies across views. Our CARVE, benefiting from our comprehensive analysis and improvements, achieves strong robustness and high accuracy across six evaluation datasets.

Qualitative comparisons are presented in Figure 2. MoGe v2 + LoFTR demonstrates detailed visualizations but suffers from poor temporal consistency across frames. In contrast, VGGT produces consistent results, but its accuracy is relatively suboptimal. Our proposed method effectively balances both spatial accuracy and temporal consistency, achieving superior overall performance.

4.2. Video Depth Estimation

We evaluate video depth using sequence-level scale alignment. Specifically, the predicted depth sequence is aligned to the ground-truth depth sequence with a single global scale factor, and we report the absolute relative error Rel$=\mathbb{E}_{p\in\mathcal{M}}\ |(\hat{\mathbf{D}}_p - \mathbf{D}_p)|/\mathbf{D}_p$ and the percentage of pixels $\delta = \mathbb{E}_{p\in\mathcal{M}}\ \max\big(\hat{\mathbf{D}}_p/\mathbf{D}_p, \mathbf{D}_p/\hat{\mathbf{D}}_p\big) < 1.25$. Spann3R is not evaluated because it outputs point clouds only in world coordinates. For MoGe v2, we first align the predicted depth to the sparse LoFTR points on each frame to recover frame-wise depth, and then apply an additional single global scale factor to the whole sequence for fair video-level evaluation.

Quantitative comparisons of video depth estimation are shown in Table 7. MoGe v2 is limited by inaccurate matching information. Similarly, VGGT outperforms other multi-view methods, including Spann3R and Fast3R. Benefiting from our analysis and improvements, CARVE achieves performance on par with the strongest baseline overall, while outperforming prior methods on several datasets and metrics.

4.3. Camera Pose and Intrinsics Estimation

For camera pose estimation, we follow [51] to align the predicted camera pose with the ground truth and evaluate the absolute trajectory error (ATE), relative pose error of rotation (RPE-R), and translation (RPE-T). For camera intrinsics, we evaluate the accuracy with the "FoV Rel", which is defined as the absolute relative error of the field of view (FoV Rel$= \mathbb{E}_t\ |\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t|/\boldsymbol{\theta}_t$) to ensure the evaluation of camera intrinsics is independent of image resolution. For MoGe v2, we only evaluate the FoV Rel metric.

Quantitative comparisons are shown in Table 8 and Table 9. The results show that our proposed CARVE achieves the best overall average rank on KITTI, 7-Scenes, TUM, and HO3D, and ties for the best average rank on HAMMER, Bonn, and ETH3D.

4.4. Monocular Depth Estimation

Similar to video depth evaluation, we compare the monocular depth estimation metrics with other feed-forward reconstruction methods. We continue to use the absolute relative error (Rel.) and the threshold accuracy ($\delta < 1.25$, denoted as δ) for monocular depth estimation, but unlike video depth estimation, we perform per-image alignment. Quantitative comparisons are shown in Table 10. MoGe exhibits strong performance and demonstrates notable competitiveness in monocular depth estimation tasks. Remarkably, despite not being explicitly optimized for monocular depth estimation, our proposed CARVE achieves competitive performance.

5. Conclusion

In this work, we explore the critical factors for visual geometry estimation, focusing on training data, objective design, and high-resolution modeling. Based on these insights, we introduce a consistency loss and a lightweight feature-fusion module for accurate and efficient high-resolution inference. Together, these improvements lead to CARVE, which achieves strong overall performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62576315).

References

[1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Adv. Neural Inform. Process. Syst.*, 2021.

[2] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *Int. Conf. Learn. Represent.*, 2024.

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv: Comp. Res. Repository*, 2020.

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5410–5418, 2018.

[5] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl Wellington. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Trans. Signal Process.*, 38(1):68–86, 2020.

[6] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22831–22840, 2025.

[7] Kai Cheng, Yiting Ma, Bin Sun, Yang Li, and Xuejin Chen. Depth estimation for colonoscopy images with self-supervised learning from videos. In *Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 119–128. Springer, 2021.

[8] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Adv. Neural Inform. Process. Syst.*, 33:22158–22169, 2020.

[9] Hainan Cui, Xiang Gao, Shuhan Shen, and Zhanyi Hu. Hsfm: Hybrid structure-from-motion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1212–1221, 2017.

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017.

[11] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *Int. Conf. Learn. Represent.*, 2024.

[12] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8585–8594, 2022.

[13] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Eur. Conf. Comput. Vis.*, pages 834–849. Springer, 2014.

[14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):611–625, 2017.

[15] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11444–11453, 2020.

[16] 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 *Eur. Conf. Comput. Vis.*, pages 241–258. Springer, 2024.

[17] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *Int. J. Comput. Vis.*, 85(1):1–15, 2009.

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.*, 32(11):1231–1237, 2013.

[19] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Int. Conf. Comput. Vis.*, pages 1–8, 2007.

[20] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3196–3206, 2020.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[22] Xiankang He, Guangkai Xu, Bo Zhang, Hao Chen, Ying Cui, and Dongyan Guo. Diffcalib: Reformulating monocular camera calibration as diffusion-based dense incident map generation. In *Proc. AAAI Conf. Artif. Intell.*, pages 3428–3436, 2025.

[23] Mu Hu, Wei Yin, Chi 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 Trans. Pattern Anal. Mach. Intell.*, 2024.

[24] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2821–2830, 2018.

[25] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1071–1081, 2025.

[26] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 780–791, 2023.

[27] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9492–9502, 2024.

[28] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Eur. Conf. Comput. Vis.*, pages 71–91. Springer, 2024.

[29] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10486–10496, 2025.

[30] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *Int. Conf. Learn. Represent.*, 2026.

[31] Biyang Liu, Huimin Yu, and Yangqi Long. Local similarity pattern and cost self-reassembling for deep stereo matching networks. In *Proc. AAAI Conf. Artif. Intell.*, pages 1647–1655, 2022.

[32] Ming Liu. Robotic online path planning on point cloud. *IEEE Trans. Cybern.*, 46(5):1217–1228, 2015.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2019.

[34] Bilawal Mahmood, SangUk Han, and Dong-Eun Lee. Bim-based registration and localization of 3d point clouds of indoor scenes using geometric features for augmented reality. *Remote Sens.*, 12(14):2302, 2020.

[35] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4981–4991, 2023.

[36] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Int. Conf. Comput. Vis.*, pages 3248–3255, 2013.

[37] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Trans. Robot.*, 31(5):1147–1163, 2015.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[39] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019.

[40] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *Eur. Conf. Comput. Vis.*, pages 58–77. Springer, 2024.

[41] Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 20524–20531, 2025.

[42] Alessio Pierluigi Placitelli and Luigi Gallo. Low-cost augmented reality systems via 3d point cloud sensors. In *Int. Conf. Signal Image Technol. Internet-Based Syst.*, pages 188–192. IEEE, 2011.

[43] 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 *Int. Conf. Comput. Vis.*, pages 10912–10922, 2021.

[44] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4938–4947, 2020.

[45] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4104–4113, 2016.

[46] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Eur. Conf. Comput. Vis.*, pages 501–518. Springer, 2016.

[47] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3260–3269, 2017.

[48] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2930–2937, 2013.

[49] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019.

[50] B Starly, Z Fang, W Sun, A Shokoufandeh, and W Regli. Three-dimensional reconstruction for medical-cad modeling. *Comput. Aided Des. Appl.*, 2(1-4):431–438, 2005.

[51] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D slam systems. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012.

[52] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8922–8931, 2021.

[53] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Adv. Neural Inform. Process. Syst.*, 34:16558–16569, 2021.

[54] Phillip Thomas, Lars Pandikow, Alex Kim, Michael Stanley, and James Grieve. Open synthetic dataset for improving cyclist detection, 2021.

[55] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8942–8952, 2021.

[56] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *Int. Conf. 3D Vision*, 2025.

[57] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5294–5306, 2025.

[58] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10510–10522, 2025.

[59] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5261–5271, 2025.

[60] 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. In *Adv. Neural Inform. Process. Syst.*, 2025.

[61] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20697–20709, 2024.

[62] 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 *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 4909–4916. IEEE, 2020.

[63] Xixun Wang, Yoshiki Mizukami, Makoto Tada, and Fumitoshi Matsuno. Navigation of a mobile robot in a dynamic environment using a point cloud map. *Artif. Life Robot.*, 26 (1):10–20, 2021.

[64] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *Int. Conf. Learn. Represent.*, 2026.

[65] Guangkai Xu, Wei Yin, Hao Chen, Chunhua Shen, Kai Cheng, and Feng Zhao. Frozenrecon: Pose-free 3d scene reconstruction with frozen depth models. In *Int. Conf. Comput. Vis.*, pages 9276–9286. IEEE, 2023.

[66] Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? In *Int. Conf. Learn. Represent.*, 2025.

[67] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21924–21935, 2025.

[68] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10371–10381, 2024.

[69] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Adv. Neural Inform. Process. Syst.*, 37:21875–21911, 2024.

[70] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Eur. Conf. Comput. Vis.*, pages 767–783, 2018.

[71] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1790–1799, 2020.

[72] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Int. Conf. Comput. Vis.*, pages 12–22, 2023.

[73] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 204–213, 2021.

[74] 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. In *Int. Conf. Comput. Vis.*, pages 9043–9053, 2023.

[75] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21936–21947, 2025.

[76] Quanwen Zhu, Long Chen, Qingquan Li, Ming Li, Andreas Nüchter, and Jian Wang. 3d lidar point cloud based intersection recognition for autonomous driving. In *IEEE Intell. Veh. Symp.*, pages 456–461. IEEE, 2012.

Unlocking the Power of Critical Factors for 3D Visual Geometry Estimation

Supplementary Material

6. More Analysis

Comparison with VGGT Fine-tuned on the Same Data. For a fair comparison, we further fine-tune VGGT on "Data3" under the same final training setting as CARVE. The model is initialized from the official VGGT pretrained weights and fully fine-tuned for 30K iterations. Other settings, including the optimizer, data preprocessing, and evaluation protocol, follow Sec. 7. Quantitative results are reported in Table 11.

Scaling up the training data significantly improves VGGT. However, CARVE still achieves better overall performance across point cloud estimation, video depth estimation, and camera pose/intrinsic estimation. This suggests that the gains of CARVE come not only from stronger training data, but also from our improved training objective and architecture.

Qualitative Effect of Removing $\mathcal{L}_{sg}$ and $\mathcal{L}_{conf}$. We directly run inference using the models trained with the corresponding loss ablation settings in the main paper. Representative examples are shown in Figure 3. Removing $\mathcal{L}_{sg}$ and $\mathcal{L}_{conf}$ leads to only limited visual differences, while the overall scene geometry and depth structure remain similar. This is consistent with the quantitative results in Table 1, suggesting that these terms mainly affect optimization behavior rather than the overall prediction structure.

Figure 3. Removing the spatial gradient loss and confidence loss has minimal impact on qualitative results when continuing training from VGGT pretrained weights.

7. Experimental Setting Details

In the supplementary material, we provide additional details and quantitative results. 1) We present more training and evaluation details for the ablation study and main experiments; 2) We include extended visualization results in Figure 4.

Common Training Details. The experiments were conducted on a server running Ubuntu 22.04 equipped with two Intel Xeon Platinum 8558 CPUs (192 threads in total) and 1.8 TB system memory. The system was configured with eight NVIDIA H200 GPUs using NVIDIA driver 570.133.20 and CUDA 12.4.

The model is initialized with the VGGT [57] pretrained weights for common parameters. Unless otherwise specified, we freeze the ViT feature extractor and train the remaining components. The regression loss of the camera head is scaled by a factor of 5 to balance between tasks.Training is performed using the AdamW optimizer [33] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a weight decay of 0.01. The learning rate is scheduled using the OneCycleLR policy [49]. The longer side of the low-resolution input image is resized to 518 pixels, and the shorter side is then randomly cropped to one of (448, 378, 308, 238) pixels. For data augmentation, we use random Gaussian blur, Gaussian noise, color jittering, and grayscale. The predicted point cloud, depth map, and camera translation are aligned with the ground-truth values via a scale factor for each sequence before computing the training loss. For the training datasets, they are categorized into three groups: indoor scenes, autonomous driving, and others. To ensure balanced dataset components, we normalize the dataset sizes such that each group contributes an equal volume of data, and individual datasets in each group are expanded to maintain intra-group balance. To accelerate training and reduce CUDA memory requirement, we employ Flash Attention v2 [11] in all attention blocks and utilize ZeRO Stage 2 optimization provided by the HuggingFace Accelerate framework. We use PyTorch's gradient checkpointing technique to reduce the CUDA memory usage. A random seed of 2025 is used in our experiment.

Training Details for Ablation Study. The model is trained with a learning rate of 3e-6 for 30K iterations on a single NVIDIA H200 GPU, and we employ a dynamic batch size with the sequence length varying between 2 and 24 frames. We constrain the total number of input images to a maximum of 24 for each iteration.

For the training data ablation study, we adopt the original loss component $\mathcal{L}_{reg}$ + $\mathcal{L}_{sg}$ + $\mathcal{L}_{conf}$. For the training objective ablation, we use the training dataset component of "Data3", including ScanNet++ [72], Hypersim [43], ScanNet [10], ARKitScenes [1], GraspNet [15], Virtual KITTI2 [3], MVS-Synth [24], Parallel Domain [54], Spring [35], UnrealStereo4K [55], Tartanair [62], TartanGround [41], and BlendedMVS [71].

Training Details for CARVE. Based on the preceding analysis, we adopt our efficient high-resolution adaptation with two cross-attention blocks to handle the input high-resolution image.

During training, the corresponding high-resolution im-

Table 11. Supplementary quantitative comparisons between VGGT finetuned and CARVE (Ours). Colors indicate full-rank colormap within each two-row block.

(a) Point cloud estimation on KITTI, 7-Scenes, and TUM.

Method	KITTI				7-Scenes				TUM				Rank↓
	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	
VGGT finetuned	0.250	0.281	0.746	0.886	0.048	0.668	0.990	0.998	0.034	0.816	0.989	0.997	1.67
CARVE (Ours)	0.238	0.257	0.767	0.892	0.043	0.720	0.986	0.998	0.029	0.861	0.991	0.997	1.17

(b) Point cloud estimation on HAMMER, Bonn, and ETH3D.

Method	HAMMER				Bonn				ETH3D				Rank↓
	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	C-L1↓	F@5↑	F@25↑	F@50↑	
VGGT finetuned	0.010	0.995	1.000	1.000	0.045	0.684	0.990	0.998	0.240	0.383	0.753	0.865	1.58
CARVE (Ours)	0.012	0.999	1.000	1.000	0.043	0.720	0.986	0.998	0.236	0.423	0.765	0.867	1.17

(c) Video depth estimation on seven datasets.

Method	KITTI		7-Scenes		TUM		HO3D		HAMMER		Bonn		ETH3D		Rank↓
	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	
VGGT finetuned	0.082	0.934	0.065	0.938	0.039	0.977	0.248	0.845	0.021	0.996	0.041	0.957	0.023	0.995	1.50
CARVE (Ours)	0.082	0.933	0.062	0.940	0.040	0.976	0.220	0.869	0.020	0.996	0.041	0.959	0.023	0.997	1.21

(d) camera pose and intrinsics on KITTI, 7-Scenes, TUM, and HO3D.

Method	KITTI				7-Scenes				TUM				HO3D	Rank↓
	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	
VGGT finetuned	0.074	1.282	0.015	2.369	0.036	0.057	0.062	0.105	0.043	0.046	0.037	0.065	0.039	1.62
CARVE (Ours)	0.078	0.664	0.016	1.740	0.024	0.052	0.064	0.104	0.049	0.041	0.032	0.060	0.039	1.31

(e) camera pose and intrinsics on HAMMER, Bonn, and ETH3D.

Method	HAMMER				Bonn				ETH3D				Rank↓
	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	FoV Rel↓	ATE↓	RPE-R↓	RPE-T↓	
VGGT finetuned	0.036	0.002	0.004	0.003	0.023	0.038	0.025	0.048	0.030	0.233	0.025	0.270	1.50
CARVE (Ours)	0.035	0.001	0.004	0.003	0.028	0.044	0.029	0.056	0.018	0.184	0.022	0.223	1.33

(f) Monocular depth estimation on seven datasets.

Method	KITTI		7-Scenes		TUM		HO3D		HAMMER		Bonn		ETH3D		Rank↓
	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	Rel↓	$\delta\uparrow$	
VGGT finetuned	0.106	0.889	0.070	0.939	0.052	0.968	0.250	0.833	0.029	0.996	0.048	0.979	0.037	0.981	1.86
CARVE (Ours)	0.106	0.885	0.066	0.941	0.049	0.969	0.236	0.851	0.028	0.994	0.035	0.985	0.033	0.985	1.14

age maintains twice the resolution of the low-resolution input. For each GPU, we employ a dynamic batch size with the sequence length varying between 2 and 50 frames, and constrain the total number of input images to a maximum of 50 for each iteration. We use a learning rate of 1e-5, and train the model for 30K iterations on 8 NVIDIA H200 GPUs. For training loss, we adopt the final loss plan of "$\mathcal{L}_{\text{reg}}(\mathbf{W}_{\text{inv}}) + \mathcal{L}_{\text{F}} + \mathcal{L}_{\text{consis}}$".

For training data, we leverage a diverse set of datasets same to "Data3", including ScanNet++ [72], Hypersim [43], ScanNet [10], ARKitScenes [1], GraspNet [15], Virtual KITTI2 [3], MVS-Synth [24], Parallel Domain [54], Spring [35], UnrealStereo4K [55], Tartanair [62], TartanGround [41], and BlendedMVS [71]. The training datasets are listed in Table 12.

Evaluation Details. To demonstrate the generalization capability of each method and assess their practical applicability, we evaluate visual geometry estimation across multiple datasets, including KITTI [18], 7-Scenes [48], HO3D [20], TUM [51], ETH3D [47], HAMMER [26], and Bonn [39]. For the ablation study, we evaluate on 7-Scenes, Bonn, KITTI, and TUM. Similar to the evaluation of FrozenRecon [65], each dataset comprises multiple sequences, from which we uniformly sample keyframes for evaluation using a pre-defined stride between consecutive frames, with a maximum keyframe number of 200. We perform evaluations on the NVIDIA H200 GPU. Due to CUDA memory constraints, we limit the number of frames per sequence to a

Table 12. The data type, quality, sequence count and image count of training datasets.

Training Dataset	Data Type	Data Quality	Sequences	Images
ScanNet++ [72]	Indoor	High	280	175661
Hypersim [43]	Indoor	High	743	72019
ScanNet [10]	Indoor	Middle	1513	2477378
ARKitScenes [1]	Indoor	Middle	2312	2049625
GraspNet [15]	Indoor	Middle	380	97280
Virtual KITTI2 [3]	Driving	High	100	42520
MVS-Synth [24]	Driving	High	120	12000
Parallel Domain [54]	Driving	High	367	347480
Spring [35]	Other	High	74	10000
UnrealStereo4K [55]	Other	High	18	16400
Tartanair [62]	Other	High	738	613274
TartanGround [41]	Other	High	14	18484
BlendedMVS [71]	Other	Middle	615	132961
Total	–	–	7274	6065082

Table 13. The data type, sequence count, average frames per sequence of evaluation datasets. "Stride" means that we sample every "Stride" element from the sequence.

Eval Dataset	Data Type	Sequences	Avg. Frames	Stride
KITTI [18]	Driving	6	107.8	1
7-Scenes [48]	Indoor	46	187.0	5
HO3D [20]	Indoor & Object	13	198.5	5
TUM [51]	Indoor	9	199	3
ETH3D [47]	Indoor & Outdoor	11	34.5	1
HAMMER [26]	Indoor & Object	9	130.0	1
Bonn [39]	Indoor	26	185.7	3

maximum of 200. The evaluation datasets are listed in Table 13. For KITTI, we use the sequences of 2011_09_26_0001, 2011_09_26_0009, 2011_09_26_0091, 2011_09_28_0001, 2011_09_29_0004, and 2011_09_29_0071.

For point cloud estimation, we aggregate the predictions of each sequence in world coordinates by stacking the individual estimations. To assess both per-view accuracy and cross-view consistency, we align the stacked predicted point cloud with the corresponding stacked ground-truth point cloud using a similarity transformation comprising a scale factor, a rotation matrix, and a translation vector.

For camera pose translation vector and video depth estimation, we align the predictions with the ground truth through a scale value for each sequence. For monocular depth estimation, we align a scale value for each image.

Figure 4. More quantitative results of our CARVE model.