# VGGT-Ω

Jianyuan Wang[1,2] Minghao Chen[1] Shangzhan Zhang[1] Nikita Karaev[1]
Johannes Schönberger[2] Patrick Labatut[2] Piotr Bojanowski[2] David Novotny
Andrea Vedaldi[1,2] Christian Rupprecht[1]

[1]Visual Geometry Group, University of Oxford [2]Meta AI

## Abstract

*Recent feed-forward reconstruction models, such as VGGT, have proven competitive with traditional optimization-based reconstructors while also providing geometry-aware features useful for other tasks. Here, we show that the quality of these models scales predictably with model and data size. We do so by introducing VGGT-Ω, which substantially improves reconstruction accuracy, efficiency, and capabilities for both static and dynamic scenes. To enable training this model at an unprecedented scale, we introduce architectural changes that improve training efficiency, a high-quality data annotation pipeline that supports dynamic scenes, and a self-supervised learning protocol. We simplify VGGT's architecture by using a single dense prediction head with multi-task supervision and removing the expensive high-resolution convolutional layers. We also use registers to aggregate scene information into a compact representation and introduce register attention, which restricts inter-frame information exchange to these registers, in part replacing global attention. In this way, during training, VGGT-Ω uses only ∼30% of the GPU memory of its predecessor, which allows us to train VGGT-Ω with 15× more supervised data than prior work and to leverage vast amounts of unlabeled video data. VGGT-Ω achieves strong results for reconstruction of static and dynamic scenes across multiple benchmarks,* e.g.*, improving over the previous best camera estimation accuracy on Sintel by 77%. We also show that the learned registers can improve vision-language-action models and support alignment with language, suggesting that reconstruction can be a powerful and scalable proxy task for spatial understanding. Project page:* http://vggt-omega.github.io/

## 1. Introduction

Recent work [91, 101, 181, 186, 189] has shown that feed-forward reconstruction models can, in many cases, match and even surpass traditional structure-from-motion (SfM) pipelines [57, 140, 149]. Furthermore, the tokens learned by such models have been used as effective geometry-aware representations in many other tasks [1, 9, 18, 51, 66, 72, 90, 93, 132, 135, 156, 172, 173, 196, 200, 204, 205, 214, 220, 222, 225, 227, 230]. This indicates that reconstruction can serve as a proxy task for learning representations useful for spatial understanding in general, with a foundational value. However, compared to foundation models where the role of scale is well understood [80, 147, 215], this is less explored in 3D computer vision. In this paper, we therefore ask whether feed-forward reconstruction models can be scaled up, and what benefits such scaling brings. To answer this question, we introduce VGGT-Ω, scaling feed-forward reconstruction to significantly larger data and, optionally, model size than prior work.

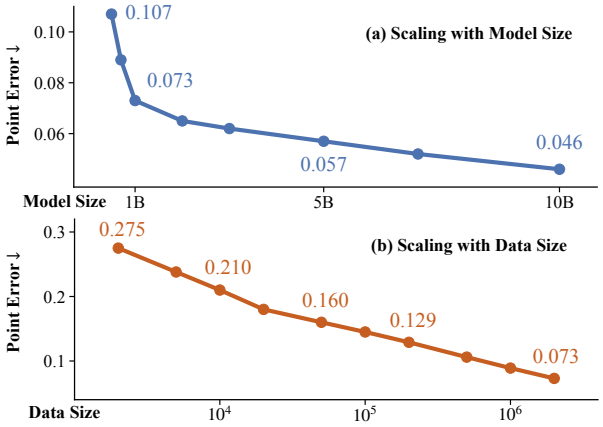


Figure 1. **Performance Gains from Data and Model Parameter Scaling.** As model size increases from 0.2B to 10B parameters and data scale grows from 2K to 2M sequences, performance improves consistently, as measured by 3D point error (lower is better; note the different axis scales). All models are trained on approximately the same number of tokens and evaluated by averaging over six datasets, with details provided in Sec. 4.2.

Compared to VGGT [181], the new model introduces a number of architectural improvements, beginning with how it uses registers. Recent works [32, 76, 117] noted that vision transformers (ViTs) spontaneously use a small number of image tokens to carry global information, and introduced learnable registers to do so more directly and efficiently. While VGGT already has per-frame registers, VGGT-$\Omega$ further introduces *register attention*: in a subset of the global attention layers, information exchange among frames is restricted to the registers. The updated registers then interact with other tokens locally within frame attention layers, thereby forming a bottleneck for aggregating and redistributing multi-frame information. This design encourages the registers to aggregate information about the scene as a whole, and we also call them 'scene' tokens.

There are two benefits to this design. First, while in other architectures registers are often treated as auxiliary and discarded at inference time [32], we instead show that they carry useful global information. In particular, although without explicit supervision, they provide useful features for vision-language-action (VLA) models and language alignment. Second, register attention also improves efficiency. Global attention is the main computational bottleneck in VGGT, but its attention maps are very sparse [143, 174]. We find that register attention, by aggregating global information, can also serve as an efficient substitute for full global attention. Specifically, replacing 25% of global attention layers with register attention incurs no measurable performance drop, while saving around 23% FLOPs and 16% memory in the backbone during training[1].

Registers aside, we also note that high-resolution convolutional layers in dense prediction heads (*e.g.*, DPT) consume a disproportionate amount of GPU memory for storing forward activations, despite accounting for only a small fraction of the model's parameters. Techniques like FSDP or gradient checkpointing cannot eliminate the cost of storing these activations. Instead, our second change is to replace the most memory-intensive convolutional layers in the dense predictors with a single MLP followed by a pixel shuffle operator. This uses little memory without performance degradation, both quantitatively and qualitatively.

Finally, in VGGT, we showed that multi-task training, where depth maps, point maps, and tracking features are supervised directly, is beneficial. Here, we find that additional dense *heads* are unnecessary to achieve these benefits. Our third change is to still use multi-task *losses*, but to retain only a single dense head for depth prediction and a single sparse head for camera prediction.

These three changes save 70% of GPU memory during training and modestly improve inference speed.

In addition to efficiency, we find that the amount, diversity, and quality of the training data are critical for scaling. In particular, handling *dynamic* content is essential as it unlocks orders of magnitude more Internet-like videos for training. Therefore, we develop a high-quality data annotation pipeline that can produce annotations for both rigid and dynamic videos at scale. The pipeline integrates VLM-based pre-filtering, VGGT, COLMAP, modern image-matching models, and supervised geometric post-filtering. Applied to around $40$ million internal Internet-style videos, the filtering pipeline retains $0.8$ million sequences with accurate annotations, roughly one-third of which contain dynamic content. Combined with existing datasets (both real and synthetic), this yields a total of $4$M diverse scenes/sequences with accurate reconstruction annotations, more than $15\times$ as many as VGGT.

To further improve generalization, we introduce a self-supervised learning protocol inspired by DINO and related momentum teacher-student methods [19, 60, 124, 147, 160]. We maintain teacher and student models initialized from a supervised VGGT-$\Omega$ checkpoint. Both models process the same input sequences under different augmentations and frame permutations. The student is trained to match the teacher's predictions and feature distribution (after aligning the frame order), while the teacher is updated via an exponential moving average of the student. We use this protocol to train on $18$ million unlabeled videos.

These improvements allow us to investigate the scaling properties of feed-forward reconstruction models. As illustrated in Fig. 1, we observe a consistent power-law-like improvement in reconstruction accuracy (measured by point error) as we increase the model capacity from 0.2B to 10B parameters and expand the training data from a few thousand to two million different sequences.

Overall, VGGT-$\Omega$ delivers a new level of feed-forward reconstruction performance, achieving state-of-the-art results in three static and three dynamic benchmarks by a wide margin. In particular, it substantially outperforms post-optimization methods such as MegaSaM and recent feed-forward methods such as Depth Anything 3 [101]. On Sintel, VGGT-$\Omega$ attains AUC@3° of $40.0$ vs. $22.5$ (by $77\%$) and AUC@30° of $79.1$ vs. $58.3$ (by $35\%$) for camera estimation, as well as $\delta_{1.25}$ of $93.5$ vs. $74.1$ (by $26\%$) for depth estimation, while being $50\times$ faster than MegaSaM. Finally, we show that the learned registers can be reused beyond reconstruction, improving VLA models and supporting alignment with language.

## 2. Related work

**3D Reconstruction.** There is a long and rich history of research on 3D reconstruction, beginning with seminal works that established the theory of multi-view geometry [43, 58, 123, 125]. Follow-up work led to major practical advances, including robust SfM systems such as

[1] Replacing *all* global attention layers with register attention reduces FLOPs to just 6% of the original, but leads to a considerable performance drop.

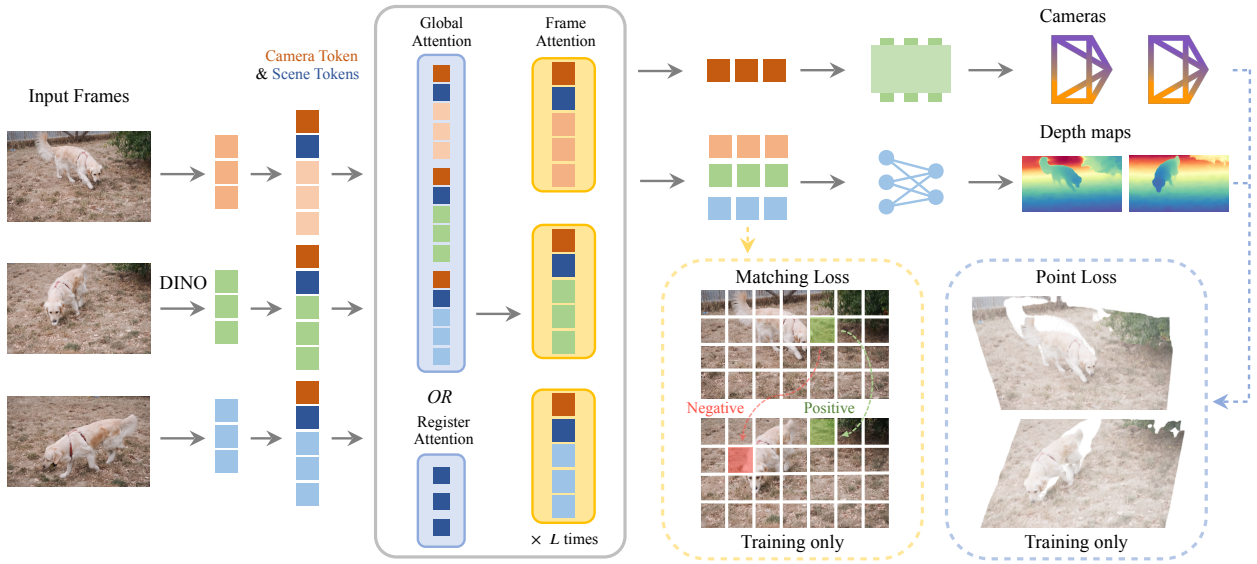


Figure 2. **Architecture Overview.** VGGT-Ω appends camera and scene tokens (registers) to image tokens, and then alternates between global attention (or register attention) and frame attention layers. We replace the redundant dense heads of VGGT with training-only losses.

COLMAP [140] and other pipelines [2, 33, 46, 106, 149]. In this paper, we focus on *feed-forward reconstruction models*, *i.e.*, neural networks that infer scene geometry and camera poses directly from one or more images. While recent SfM pipelines increasingly include learnable components such as keypoint detectors [38, 40, 170, 210] and feature matchers [21, 102, 138, 145], our work is most closely related to end-to-end differentiable SfM frameworks that learn geometry estimation directly [10, 158, 165, 166, 171, 178–180, 192, 232]. While these works demonstrate that end-to-end learning is possible in SfM, they still *combine* elements of classical SfM pipelines. DUSt3R [186] and its follow-up MASt3R [39] estimated both scene geometry and camera parameters (extrinsics and intrinsics) directly from images. However, similar to stereo approaches [47, 54, 113, 121, 129, 193, 207, 208, 223], the neural networks in DUSt3R and MASt3R operate only on image pairs and still require post-optimization to process additional views. A key improvement came from methods that process multiple images jointly, removing the need for optimization across views, including [41, 159, 175, 184, 197, 203, 221]. Among these, VGGT [181] is a representative approach that first surpassed post-optimization methods (*e.g.*, using bundle adjustment) while relying solely on feed-forward inference, prompting many follow-up works [16, 24, 25, 35, 36, 64, 75, 79, 83, 87, 95, 100, 105, 110, 114, 115, 118, 131, 156, 183, 189, 199, 211, 213, 234]. Several works improve the computational scalability of VGGT-style models through token merging, sparse attention, or descriptor-based aggregation, making many-view feed-forward reconstruction more efficient [115, 143, 174]. Other works address the quadratic cost of global self-attention by using stateful, bounded scene representations with a linear update cost. Examples include methods that use test-time-training layers [26, 78, 219]. Others [22, 87, 157, 234] process frames causally while maintaining a persistent geometric state by caching tokens, key-value pairs, local windows, or maps. Some have used VGGT to reconstruct scenes locally, fusing results by means of alignment, localization, or SLAM algorithms [35, 36, 114, 176].

Other works have extended VGGT-like models to take additional geometric information as input (*e.g.*, cameras [83]), add sensors (*e.g.*, LiDAR [183]), or model camera rigs [95]. Other extensions predict normals [42] or use mixture-of-experts designs [50, 191]. PI3 [189] removes the reliance on a fixed reference view, while Depth Anything 3 (DA3) [101] adopts vanilla DINO as its backbone, with both supporting dynamic scenes. Some studies have investigated what these models learn, probing for correspondences, epipolar structure, and connections to 3D shape perception in humans [9, 11]. SelfEvo [68] explored a self-distillation scheme using spatiotemporal context asymmetry. Beyond reconstruction itself, the features learned by feed-forward reconstruction models have also been applied as geometry-aware representations for other tasks. This includes video generation and novel view synthesis [66, 72, 75, 109, 156], vision-language models [90, 132, 173, 196, 205, 214, 220, 222, 225, 227], vision-language-action models [1, 93, 135, 172], and perception tasks such as detection, segmentation, matching, occupancy prediction, and place recognition [18, 37, 51, 200, 204, 230].

Monocular dynamic 3D reconstruction, or 4D reconstruction, aims to recover scene geometry that changes over time. This line of research also has a long history, with

early work by Bregler et al. [12] and Torresani et al. [168]. Among recent contributions, MegaSaM [99] has been particularly influential, combining feed-forward depth prediction with optimization-based non-rigid reconstruction. ViPE [65] further builds on this direction. Several works have explored feed-forward 4D reconstruction with reduced optimization. MonST3R [218] and D$^2$USt3R [55] extend DUSt3R to handle dynamic 3D content. Align3R [112] builds on DUSt3R to infer cameras and align monocular depth predictions over time, though it still relies on optimization beyond two views. CUT3R [184] and Point3R [197] support incremental reconstruction alongside dynamic scenes. Geo4D [77] fine-tunes a video generator to recover 4D geometry. PI3 [189] and DA3 [101] adopt VGGT-style models and train them with dynamic data, while PAGE-4D [231] adapts VGGT through a module that separates static and moving regions. Human3R focuses on human-scene reconstruction [27]. SpatialTracker [199], St4RTrack [44], DPM [153], V-DPM [154], Uni4D [206], Any4D [82], and related methods unify dynamic reconstruction and point tracking. D4RT [217] introduces a unified feed-forward transformer that reconstructs dynamic scenes by querying point-level 4D scene information.

**Registers in Vision Transformers (ViTs).** Recent works [32, 117] have found that a small number of tokens in ViTs encode not only local patch information but also global information. These outliers have high norms and disrupt the spatial coherence of the patch features. This, in turn, has motivated the use of register tokens, which are separate from image tokens and help preserve coherence among patch representations. Further studies have investigated the causes of this phenomenon and the interactions between register and image tokens [76, 88, 117, 144]. Here, we add register tokens to each input image frame and use them to aggregate and exchange global information across images. In this way, our registers carry information about the sequence as a whole. Rather than discarding them as is often done in prior work, we show that they are useful in downstream applications.

# 3. Method

VGGT-Ω builds on the original VGGT, improving it in several ways. We begin by describing the new architecture (Sec. 3.1) and training pipeline (Sec. 3.2) of VGGT-Ω. Then, we introduce a self-supervised training protocol to exploit unlabeled data (Sec. 3.4) and a new data pipeline for robustly annotating millions of videos (Sec. 3.5).

## 3.1. A New Scalable Architecture

VGGT-Ω, illustrated in Fig. 2, is a feed-forward transformer $f$ that maps $N$ input images $I_1, \dots, I_N \in \mathbb{R}^{3\times H\times W}$ to corresponding cameras and depth maps:

$$((\boldsymbol{g}_1, D_1), \dots, (\boldsymbol{g}_N, D_N)) = f(I_1, \dots, I_N),$$

where $D_i \in \mathbb{R}^{H\times W}$ is the depth map of image $I_i$ and $\boldsymbol{g}_i = (\boldsymbol{q}_i, \boldsymbol{t}_i, \boldsymbol{f}_i) \in \mathbb{R}^9$ is the concatenation of the rotation quaternion $\boldsymbol{q}_i \in \mathbb{R}^4$, the translation vector $\boldsymbol{t}_i \in \mathbb{R}^3$, and the field of view $\boldsymbol{f}_i \in \mathbb{R}^2$ describing the corresponding camera. As is commonly done [140, 180, 181], we assume that the principal point is at the center of the image. The problem formulation is thus similar to VGGT [181], except that the model *does not* predict point maps or tracking features directly (although it still supervises them, as discussed in Sec. 3.2). The network $f$ encodes each image into tokens (Sec. 3.1.1), aggregates features across views with alternating-attention (Sec. 3.1.2), and maps the tokens to the final predictions using lightweight heads (Sec. 3.1.3).

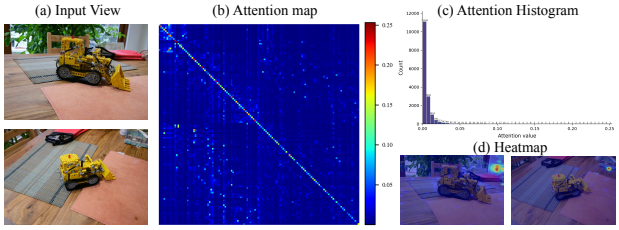


Figure 3. **Visualization of Global Attention in VGGT**. As shown in (b) the global attention matrix, (c) its value distribution, and (d) spatial heatmaps, the attention in layer 13 is quite sparse.

### 3.1.1. Feature Extraction and Tokenization

We tokenize each image $I_i$ with a DINOv3-initialized vision transformer [147], obtaining $\boldsymbol{z}_i^F = \mathrm{DINO}(I_i) \in \mathbb{R}^{H'W'\times C}$, where $H' = H/r$ and $W' = W/r$ for patch size $r$. For each image $I_i$, we also append one *camera token* $\boldsymbol{z}_i^{\text{cam}} \in \mathbb{R}^{1\times C}$ and sixteen *registers (scene tokens)* $\boldsymbol{z}_i^{\text{scene}} \in \mathbb{R}^{16\times C}$. The camera token is used to predict the camera parameters, and the registers aggregate information about the scene. As in [181], these tokens can take one of two learnable parameters, one if image $I_i$ is the *reference image* and the other otherwise. These are then concatenated to form the set of tokens $\boldsymbol{z} = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_N) \in \mathbb{R}^{N\times(H'W'+17)\times C}$ where $\boldsymbol{z}_i = (\boldsymbol{z}_i^F, \boldsymbol{z}_i^{\text{cam}}, \boldsymbol{z}_i^{\text{scene}})$ are the tokens of image $I_i$.

### 3.1.2. Register Attention

Recall that VGGT uses *alternating-attention* [181], interleaving between frame-wise self-attention within each image and global self-attention across all images, which are, by definition, permutation equivariant. Frame-wise attention eliminates the need for frame-index embeddings, which would otherwise restrict the model's ability to generalize to a flexible number of input frames. Therefore, none of the tokens has an explicit encoding of the corresponding image identity (except for indicating whether a frame is the reference). Global attention is the standard attention layer


Figure 4. **Qualitative Results.** VGGT-Ω handles both static and dynamic content, as evidenced by the overlaid traffic flow and the tennis player's trajectory. It also generalizes to hard scenes, *e.g.*, underwater coral reefs. Each example uses 64, 4, 9, 16, and 32 input frames.

applied to all tokens $\boldsymbol{z}$, which we denote $\boldsymbol{z}' = \mathrm{attn}(\boldsymbol{z})$. Frame-wise attention is similar, but applied independently to $\boldsymbol{z}_i$ for each image $I_i$, which we denote $\boldsymbol{z}' = \mathrm{attn}_f(\boldsymbol{z}) = (\mathrm{attn}(\boldsymbol{z}_1), \ldots, \mathrm{attn}(\boldsymbol{z}_N))$. Global attention is where different frames interact, and thus where multi-frame scene information is formed. At the same time, global attention is computationally expensive, as it attends to all tokens from all frames, with a cost that is quadratic in the total number of tokens. Moreover, we observe that global attention maps are typically sparse, as shown in Fig. 3, which may suggest that a small number of tokens are enough to exchange the corresponding information. This is consistent with recent findings [143, 174]. We therefore replace 25% of the global attention layers with *register attention*, in which self-attention is restricted to the registers of all frames. Formally, register attention updates only the registers: $\boldsymbol{z}' = \mathrm{attn}_{\mathrm{scene}}(\boldsymbol{z})$ where $(\boldsymbol{z}_1^{\mathrm{scene}\prime}, \ldots, \boldsymbol{z}_N^{\mathrm{scene}\prime}) = \mathrm{attn}(\boldsymbol{z}_1^{\mathrm{scene}}, \ldots, \boldsymbol{z}_N^{\mathrm{scene}})$, *i.e.*, only the registers participate in self-attention across frames in these blocks. The updated registers then interact with each frame's image tokens in subsequent frame-wise attention blocks, redistributing the aggregated scene information back to the image tokens. This encourages the final registers to carry global scene information, while also reducing the cost of global attention.

#### 3.1.3. Decoding

The final set of tokens $\boldsymbol{z}' = (\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_N)$ produced by the attention layers is decoded into depth maps and cameras.

**Depth.** In VGGT, all dense decoders are implemented with Dense Prediction Transformer (DPT) layers [134]. However, the final convolutional blocks in these DPT heads maintain several high-resolution feature maps, which are memory-intensive. To reduce this cost, we replace the blocks operating above $1/4$ of the input resolution with a lightweight upsampling head via a single MLP followed by a pixel-shuffle operator. The MLP outputs $2u^2$ channels ($u = 4$ in our implementation), and the pixel-shuffle operator rearranges them from $(H' \times W', 2u^2)$ to $(uH') \times (uW') \times 2$, where the two output channels correspond to depth and confidence.

We also explored a fully convolution-free decoder that maps tokens to dense predictions using MLPs only. While

this works well on benchmarks, qualitatively it produces blocky artifacts in the predicted depth map, especially for outdoor scenes with distant structures such as sky or mountains, where depth is unbounded and thus not well-defined. As such, we retain the early low-resolution convolutional layers in DPT since they are computationally inexpensive.

**Camera.** The cameras $(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_N)$ are predicted jointly by applying a lightweight transformer to the camera tokens and registers $\{(\boldsymbol{z}_i^{\text{cam}}, \boldsymbol{z}_i^{\text{scene}})\}_{i=1}^N$, followed by an MLP on each updated camera token. Unlike VGGT, our camera head predicts camera parameters in a single pass, without iterative refinement.

### 3.2. Training Losses

In VGGT, we found it beneficial to predict redundant dense heads (*e.g.*, point maps and tracks), but doing so is expensive during training. Instead, VGGT-Ω contains a single dense head for depth prediction. Although the model does not *directly* predict point maps and tracks, we *still* supervise these quantities through corresponding losses. We found this yields nearly the same performance as using multiple dense prediction heads while saving a significant amount of memory. We thus optimize the loss:

$$\mathcal{L} = \lambda_{\text{cam}}\mathcal{L}_{\text{cam}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{point}}\mathcal{L}_{\text{point}} + \lambda_{\text{match}}\mathcal{L}_{\text{match}} \quad (1)$$

where $\lambda_{\text{cam}}$, $\lambda_{\text{depth}}$, $\lambda_{\text{point}}$, and $\lambda_{\text{match}}$ are weights.

**Camera loss.** The camera loss $\mathcal{L}_{\text{cam}} = \sum_{i=1}^N |\hat{\boldsymbol{g}}_i - \boldsymbol{g}_i|$ compares the predicted cameras $\hat{\boldsymbol{g}}_i$ to the ground-truth ones $\boldsymbol{g}_i$ using an $\ell_1$ objective, which we found to be more stable than the Huber loss used in VGGT.

**Depth loss.** Following VGGT, the depth loss uses aleatoric uncertainty and a gradient consistency term. Additionally, we account for the relative scale with respect to the ground truth. Therefore, we have: $\mathcal{L}_{\text{depth}} = \sum_{i=1}^N \left[\left\|c_i^D \odot \left(1 + D_i^{-1}\right) \odot e_i\right\| + \left\|c_i^D \odot \nabla e_i\right\|\right] - \alpha \sum_{i=1}^N \log c_i^D$, where $e_i = \hat{D}_i - D_i$, $c_i^D$ is the predicted uncertainty map, and $\odot$ denotes element-wise product.

**Point loss.** Point maps assign to each pixel the coordinates of the corresponding 3D point in the frame of the reference camera. The point maps can thus be inferred from the depth maps and the camera parameters via unprojection. Accordingly, our point loss $\mathcal{L}_{\text{point}}$ is the same as the depth loss $\mathcal{L}_{\text{depth}}$ up to replacing the residuals with $e_i = \pi^{-1}(\hat{D}_i, \hat{\boldsymbol{g}}_i) - P_i$, where $\pi^{-1}$ denotes unprojection and $P_i$ is a point map.

**Matching loss.** The matching loss $\mathcal{L}_{\text{match}}$ is applied to the tokens output by the last attention layer. It pulls together features of positive token pairs (corresponding to the same 3D location) and pushes apart negative pairs: $\mathcal{L}_{\text{match}} = \mathbb{E}_{\text{pos}}\big[-\log \sigma(s)\big] + \mathbb{E}_{\text{neg}}\big[-\log(1-\sigma(s))\big]$, where $s$ is the cosine similarity between $\ell_2$-normalized tokens, $\sigma$ is the sigmoid function, and $\mathbb{E}$ denotes averaging over positive and negative pairs, *i.e.*, a weighted binary cross-entropy. Details of how to construct the pairs are provided in the supplement.

### 3.3. Dynamic Reconstruction

VGGT-Ω supports reconstructing dynamic scenes, which, among other benefits, unlocks orders-of-magnitude more training data (since most videos contain motion). Dynamic reconstruction requires a statistical prior that constrains movement. Classical non-rigid structure from motion [12, 31, 161] imposes hand-designed low-rank or local-rigidity constraints, which can be brittle and limited in generality. A data-driven model like VGGT-Ω has the potential to learn a better prior from data. However, the output representation determines how camera motion and scene motion are coupled. In particular, some recent methods [186] use point maps to represent the scene and recover camera parameters from them. This works well for static scenes, but in dynamic scenes it requires either segmenting out moving pixels, as in MonST3R [218], or introducing extensions such as dynamic point maps [153, 154]. Another option is to predict depth and ray maps [101]. However, ray maps add an expensive dense output and can entangle camera information with pixel-wise appearance changes. For example, a stationary camera observing a dancer has large motion but fixed camera parameters. Therefore, as stated above, we predict only depth maps and camera parameters, and avoid explicit dynamic outputs such as motion masks.

### 3.4. Self-supervised Training

Inspired by common practice in 2D vision [19, 60, 124, 147], we use a teacher-student strategy for self-supervised learning with unlabeled videos. Specifically, we maintain a *student* network that is updated by gradient descent and a *teacher* network that is updated only via an exponential moving average of the student network. Both networks are initialized from the VGGT-Ω checkpoint trained on supervised data. Given a video sequence, we feed the same set of frames to both networks but apply independent stochastic augmentations, including color jittering and blurring, random 90° rotations, random patch masking, and random frame reordering (which affects the selection of the reference frame). After restoring both streams to a common order, we require the student to match the teacher in two ways. An $\ell_2$ feature-matching loss aligns the student's tokens with the teacher's across multiple layers. Regression losses supervise the camera and depth. To prevent collapse, the camera and depth heads are *frozen* during self-supervision. The teacher's parameters are updated as $\theta^{\text{T}} \leftarrow m\,\theta^{\text{T}} + (1-m)\,\theta^{\text{S}}$ instead of gradient descent. This distillation scheme thus enforces invariance to appearance changes and frame order, enabling effective learning from millions of unlabeled videos.

### 3.5. Training Data

An important aspect of VGGT-Ω is to scale the training data, which we achieve by combining a large number of publicly available datasets (Sec. 3.5.1) with a new annotation pipeline that we develop to handle dynamic content in off-the-shelf videos (Sec. 3.5.2).

#### 3.5.1. Data Sources

We first collect several publicly available datasets: Aria series [126], Bedlam [8], BEHAVIOR-1K [92], Co3Dv2 [136], uCo3D [108], DL3DV [103], Dynamic Replica [81], EDEN [89], EFM3D [151], HOT3D [6], Habitat [139], Hypersim [137], Mapfree [4], Mapillary Metropolis [116], MPSD [3], Megadepth [97], Megasynth [74], Mid-Air [45], Mvssynth [69], ParallelDomain-4D [61], Replica [150], SAIL-VOS [63], ScanNet Series [29, 209], TartanAirV2 [187], TartanGround [128], Taskonomy [212], UnrealStereo4K [169], Virtual KITTI [15], Waymo [155], and WildRGBD [198]. We exclude Kubric [53] and PointOdyssey [228] used by VGGT because their background geometry is fake and yields invalid depth. Additionally, we use several internal datasets, which include artist-created object assets, rigid and dynamic synthetic environments, real-world device captures, and related sources. For non-synthetic datasets (*e.g.*, those annotated by SfM pipelines), we remove noisy depth values via a multi-view consistency check and discard sequences with too few valid depth pixels. In total, these datasets contain approximately 3M sequences, each containing between 10 and 20,000 images.

#### 3.5.2. Data Annotation Pipeline

To further expand our training data, we built a large internal video collection of roughly 40M Internet-style videos. We first assess each video for suitability for reconstruction, *e.g.* filtering out clips with large watermarks or abrupt shot changes. Videos that pass this check are used for self-supervised training as described in Sec. 3.4. We then introduce a new pipeline to annotate videos with camera parameters and depth maps, for both static and dynamic scenes. We prioritize annotation quality over quantity and aggressively reject low-quality data. We also discard depth annotations in regions that are likely to be dynamic. This may exclude extreme camera motions or highly dynamic scenes, but these are still well represented in the synthetic datasets. Overall, we obtain a collection of about 200K dynamic scenes and 600K static scenes with high-quality camera and depth annotations.

**VLM pre-filtering.** We prompt a Vision-Language Model (VLM) to discard videos that are unlikely to be reconstructed using multi-view geometry. The VLM classifies $50\%$ of the clips as too difficult to reconstruct, *e.g.*, because they contain multiple clips, extreme motion blur, or heavy overlays or watermarks. It also classifies $40\%$ of them as reconstructible, but potentially with low accuracy due to insufficient parallax, lack of non-repetitive texture, etc. The remaining $10\%$ of videos go to the next stage. In the same pass, the VLM extracts metadata, such as whether the scene appears dynamic, for use by later stages.

**Dynamic mask extraction.** We use Grounding DINO [107] to detect bounding boxes of potentially movable object categories, such as people and cars. These regions are then excluded from matching, tracking, and verification.

**Feature matching and tracking.** We extract matches and tracks across frames with an ensemble of methods, using SIFT [111], SuperPoint & SuperGlue [38, 138], ALIKED & LightGlue [102, 226], and VGGSfM Tracker [180]. Matches within dynamic regions are discarded.

**Reconstruction and filtering.** We use the original VGGT to initialize camera parameters when RANSAC-based essential matrix estimation yields too few inliers, and then run COLMAP [140] for iterative bundle adjustment and filtering based on the correspondences computed above. For successful reconstructions, we discard sequences that fail heuristic checks, *e.g.*, an image registration ratio $< 99.5\%$, a field of view outside $[30°, 120°]$, or a distortion ratio $> 0.1$. These criteria aggressively remove cases with degenerate motion or extreme zoom. Then, we estimate per-frame dense depth maps using patch-based multi-view stereo [141].

**Multi-view consistency.** For each frame, we unproject the depth map to 3D, reproject the points into other views, and compare them with the depths there. Pixels that satisfy this cross-view consistency check are marked valid. We discard sequences with fewer than $5\%$ of pixels with valid depth, which typically, though not always, indicates low-quality cameras.

**Supervised geometric filtering.** Finally, we obtain cameras, depths, and valid masks for every sequence. We use handcrafted features, such as camera-up-vector consistency, parallax angle, and trajectory smoothness, to describe each sequence. We hand-annotated 500 static and 500 dynamic sequences, respectively, to train a classifier to remove low-quality reconstructions. The classifier is an ensemble of XGBoost [23], random forests [13], and CatBoost [130].

## 4. Experiments

Here, we provide additional details about our framework, benchmark it against state-of-the-art rigid and dynamic reconstruction methods, and ablate key design choices.

Table 1. **Camera Pose Estimation** across static and dynamic benchmarks. The metric AUC is higher-is-better. Feed-forward models (DA3, PI3, VGGT) are robust across datasets, but still lag behind the dynamic optimization-based method MegaSaM at strict thresholds on Sintel (*e.g.*, AUC@3° of 16.2 *vs*. 22.5). In contrast, dynamic optimization methods (MegaSaM, MonST3R) degrade on wide-baseline scenes (*e.g.*, AUC@30° of 38.1 *vs*. 86.4 on ETH3D). Our method, however, substantially advances the state of the art across all scenarios, *e.g.*, improving AUC@3° from 22.5 to 40.0 on Sintel, with a 77% relative improvement.

| Method | Static Scenes | | | | | | Dynamic Scenes | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 7 Scenes | | NRGBD | | ETH3D | | DyCheck | | Sintel | | TUM-Dynamic | |
| | AUC@3° | AUC@30° | AUC@3° | AUC@30° | AUC@3° | AUC@30° | AUC@3° | AUC@30° | AUC@3° | AUC@30° | AUC@3° | AUC@30° |
| MonST3R | 9.0 | 68.3 | 13.9 | 79.7 | 1.7 | 14.3 | 11.5 | 45.4 | 4.3 | 45.8 | 7.7 | 48.5 |
| MapAnything | 5.8 | 61.4 | 35.2 | 88.9 | 13.2 | 51.0 | 6.1 | 60.3 | 2.9 | 31.6 | 4.3 | 40.2 |
| MegaSaM | 10.6 | 71.8 | 17.2 | 83.1 | 5.9 | 38.1 | 26.8 | 53.1 | 22.5 | 58.3 | 15.4 | 59.0 |
| VGGT | 10.9 | 74.4 | 81.7 | 97.7 | 18.8 | 62.1 | 21.0 | 78.7 | 15.0 | 50.0 | 16.6 | 61.2 |
| PI3 | 13.3 | 77.0 | 83.8 | 98.2 | 35.3 | 79.6 | 23.3 | 81.0 | 14.8 | 53.5 | 16.1 | 59.2 |
| DA3 | 18.7 | 78.2 | 86.4 | 98.4 | 46.1 | 87.0 | 32.1 | 83.9 | 16.2 | 52.7 | 20.8 | 62.7 |
| Ours-1B | 29.6 | 83.1 | 89.7 | 98.8 | 49.8 | 88.5 | 38.4 | 87.3 | 35.3 | 73.0 | 30.2 | 82.3 |
| Ours-10B | **36.4** | **88.2** | **92.5** | **99.1** | **56.3** | **90.4** | **43.7** | **90.9** | **40.0** | **79.1** | **36.4** | **87.5** |

Table 2. **Depth Estimation** across static and dynamic benchmarks. The metric $\delta_{1.25}$ denotes the percentage of predicted depths within a factor of the ground truth (higher is better), and AbsRel is the mean absolute relative error (lower is better).

| Method | Static Scenes | | | | | | Dynamic Scenes | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 7 Scenes | | NRGBD | | ETH3D | | DyCheck | | Sintel | | TUM-Dynamic | |
| | $\delta_{1.25}$ | AbsRel | $\delta_{1.25}$ | AbsRel | $\delta_{1.25}$ | AbsRel | $\delta_{1.25}$ | AbsRel | $\delta_{1.25}$ | AbsRel | $\delta_{1.25}$ | AbsRel |
| MonST3R | 92.4 | 0.075 | 98.4 | 0.030 | 95.8 | 0.056 | 93.3 | 0.068 | 71.9 | 0.263 | 85.0 | 0.148 |
| MapAnything | 92.9 | 0.070 | 98.7 | 0.022 | 96.3 | 0.035 | 97.0 | 0.049 | 72.5 | 0.251 | 93.1 | 0.052 |
| MegaSaM | 93.8 | 0.065 | 96.2 | 0.057 | 94.8 | 0.083 | 97.4 | 0.042 | 74.1 | 0.207 | 92.9 | 0.083 |
| VGGT | 91.9 | 0.073 | 99.1 | 0.019 | 97.4 | 0.036 | 95.2 | 0.055 | 79.2 | 0.189 | 92.2 | 0.064 |
| PI3 | 92.8 | 0.068 | 99.2 | 0.011 | 99.6 | 0.016 | 97.4 | 0.041 | 82.5 | 0.144 | 95.5 | 0.046 |
| DA3 | 93.0 | 0.063 | 99.5 | 0.010 | 99.6 | 0.015 | 97.7 | 0.039 | 86.1 | 0.118 | 94.3 | 0.049 |
| Ours-1B | 94.6 | 0.058 | 99.6 | 0.010 | 99.8 | 0.012 | 98.4 | 0.038 | 89.5 | 0.097 | 97.4 | 0.041 |
| Ours-10B | **96.3** | **0.050** | **99.7** | **0.007** | **99.8** | **0.009** | **98.7** | **0.030** | **93.5** | **0.081** | **98.3** | **0.035** |

## 4.1. Implementation Details

We focused on four model variants: 200M, 500M, 1B, and 10B parameters, with $12/12/24/16$ alternating-attention blocks and hidden sizes $384/768/1024/4096$, respectively. The vision transformer is initialized from DINOv3 [147] and is not frozen during training. Each block contains one global attention (or register attention) layer and one frame-wise attention layer. Optimization uses AdamW for 240K iterations, with 160K supervised, 50K self-supervised, and a final 30K supervised stage. The learning rate follows a linear warm-up over 5% of training and cosine decay for the remaining 95%, with a peak value of $2 \times 10^{-4}$ for supervised training and $1 \times 10^{-4}$ for self-supervised training. For each batch, the number of frames is drawn uniformly from $[1, 24]$. We augment images by randomly varying the aspect ratio within $[0.33, 1.33]$, keeping the image area approximately $512 \times 512$ pixels, and applying color jittering, grayscale conversion, and random patch masking. Training used 128 96GB H100 GPUs, bfloat16 mixed precision, gradient checkpointing, and FSDP.

## 4.2. Benchmarking

**Quantitative Comparison.** We compare VGGT-Ω with recent approaches: (i) feed-forward reconstruction models and (ii) optimization-based dynamic reconstruction methods. We evaluate on three static datasets (7 Scenes [146], NRGBD [5], and ETH3D [142]) and three dynamic datasets (DyCheck [49], Sintel [14], and TUM-Dynamic [152]). For each scene or sequence, we randomly sample 10 frames. We use the original released models for all methods. For DA3, we use its largest variant, Giant (1B parameters).

Following [181], we report the standard AUC for camera pose estimation (higher is better), computed as the area under the curve of the fraction of image pairs whose relative rotation and translation errors fall below an angular threshold (*e.g.*, 3°, 30°). As shown in Tab. 1, feed-forward models generally exhibit strong performance on static benchmarks and at more relaxed thresholds, while optimization-based, dynamic-aware MegaSaM is more competitive on challenging dynamic sequences such as Sintel but degrades on wide-baseline or low-texture scenes. In contrast, our models consistently outperform all baselines across both static and dynamic datasets and at both strict and relaxed thresholds.

We also evaluate the accuracy of the predicted depths using absolute relative error (AbsRel; lower is better) and $\delta_{1.25}$ (higher is better), which measures the percentage of pixels for which the ratio of the predicted depth to the ground-truth depth is within a specified threshold. As shown in Tab. 2, our models outperform the baselines in the static benchmarks, further lowering AbsRel on datasets where existing methods perform strongly, such as ETH3D,

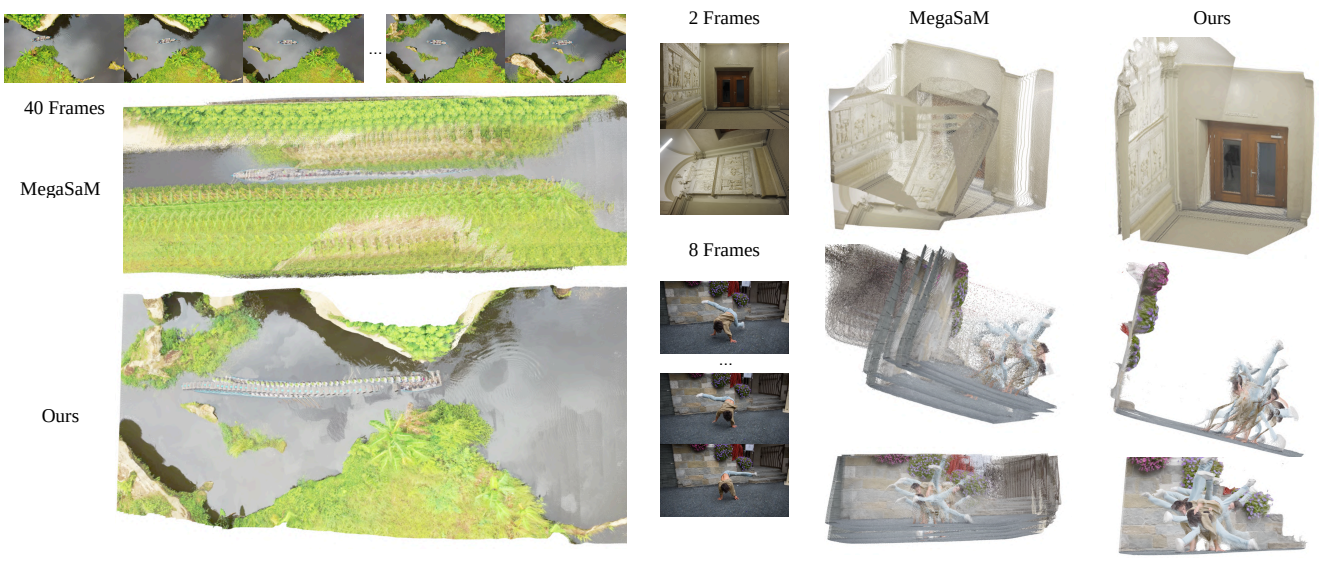


Figure 5. **Qualitative Comparison to MegaSaM.** MegaSaM often suffers from geometric inconsistencies. In contrast, our method produces globally consistent reconstructions across sparse, dynamic, and aerial scenarios. This is especially evident in the aerial scene in the left column, where MegaSaM exhibits severe geometric drift and texture smearing, resulting in repeated patterns and a distorted global layout. Note that the second frame in the top right example is not upright, which makes this case particularly challenging.

and even more so in dynamic scenes, where they reduce depth errors and increase $\delta_{1.25}$ (*e.g.*, on Sintel, improving $\delta_{1.25}$ from 86.1 to 93.5 and AbsRel from 0.118 to 0.081).

The larger 10B variant consistently outperforms the 1B model, indicating that scaling up the reconstruction model directly benefits both camera and depth accuracy.

**Qualitative Results.** Figure 4 illustrates our results on static and dynamic scenes, including traffic, human motion, natural landscapes, and underwater environments.

We further compare VGGT-Ω with DA3 and MegaSaM on challenging cases in Figs. 5 and 6. In Fig. 6, DA3 struggles with repeated textures, estimating little to no camera motion in the snow lift sequence, and with strong camera roll, reconstructing the tower several times in the drone sequence. MegaSaM can also break down in sparse indoor scenes with substantial camera roll (top right) or textureless walls (bottom right), producing disjoint structures or misaligned planes, while certain aerial sequences (left) also challenge its pose estimation. In contrast, VGGT-Ω's reconstructions are globally consistent, likely due to the strength of the geometric priors learned from diverse data.

**Inference Memory and Speed.** We now compare the *inference* memory and speed of VGGT-Ω with VGGT and DA3 (note that this is quite different from the training efficiency discussed in Sec. 3.1).

For these comparisons, we first correct the original VGGT implementation. That model caches intermediate tensors from all 24 layers during inference. Although these cached tensors are useful during training, the prediction heads only require intermediate features from four layers at test time. We thus avoid caching unnecessary information, which substantially reduces inference memory usage.

With this correction in place, we compare VGGT, DA3, and VGGT-Ω in Fig. 7. For a fair comparison, all measurements are conducted on a single 80GB A100 GPU using PyTorch *scaled_dot_product_attention*, with a flash attention v2 backend. We use similar input resolutions matched to each model's patch size: $518 \times 336$ for VGGT and DA3 with 14-pixel patches, and $512 \times 336$ for VGGT-Ω with 16-pixel patches. We increase the number of input frames until each method runs out of memory, as indicated by the cross markers in Fig. 7.

As for memory efficiency, VGGT (with the correction) and VGGT-Ω are similar, and can process more than 1,000 frames on a single A100 GPU. As for speed, VGGT-Ω is faster, primarily due to using DINOv3, with a patch size of 16, instead of DINOv2, with a patch size 14, which reduces the number of image tokens by about 25%. In addition, VGGT-Ω replaces 25% of the global attention layers with register attention by default, reducing FLOPs and yielding a 20–25% speedup. Overall, VGGT and VGGT-Ω scale more favorably than DA3 in both memory and runtime, *e.g.*, DA3 runs out of memory at around 750 frames in our testing, whereas VGGT and VGGT-Ω can handle around 1250 frames. An even more aggressive variant of VGGT-Ω that replaces all global attention layers with register attention can further improve speed, reducing the runtime on 1000 frames from 240.2 seconds to 11.7 seconds, albeit at the cost of lower reconstruction accuracy (Sec. 5).

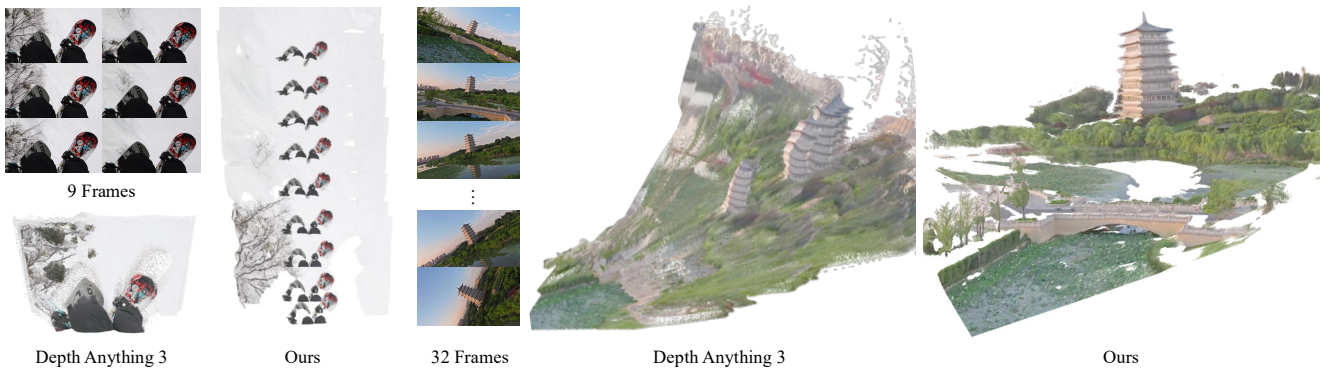


Figure 6. **Qualitative Comparison to Depth Anything 3.** *Left:* a snow lift sequence over a snow-covered field, where the repetitive terrain misleads DA3 into estimating little to no camera motion. Our method recovers the correct camera trajectory. *Right:* a drone sequence in which the camera rolls while flying toward a tower. Under this strong viewpoint change, DA3 produces severe ghosting and duplicated tower structures, whereas our reconstruction remains globally consistent.

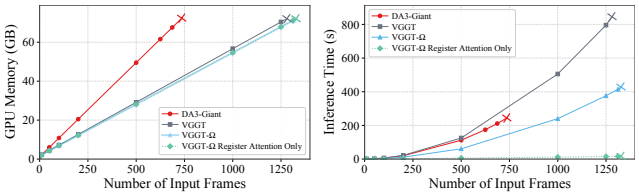


Figure 7. **Memory and Speed Comparison.** We compare the inference memory usage and runtime of VGGT, DA3, and VGGT-Ω on an 80GB A100 GPU with flash attention v2 enabled. For VGGT, we first address an implementation detail which improves its efficiency (see text). Cross markers indicate the first setting at which each method runs out of memory.

While perhaps surprising, we stress that, during inference, the main benefit of removing global attention is speed, not memory. As shown in Fig. 7, replacing all global attention layers with register attention produces a large speedup, while peak inference memory remains almost unchanged. The speedup is because register attention avoids computing interactions between all pairs of image tokens, unlike global attention. However, with PyTorch *scaled_dot_product_attention* using flash attention v2, the full attention matrix is not explicitly materialized in memory. Instead, the kernel computes attention in a tiled streaming manner, updates the softmax normalization online, and directly accumulates the output. Peak memory is instead dominated by tensors whose size is proportional to the number of frames and image tokens, such as frame-attention activations or feed-forward intermediates. This explains why memory grows approximately linearly with the number of frames in Fig. 7.

### 4.3. Ablation Studies

Unless otherwise specified, ablations use the 1B model variant trained on 2M sequences with 64 GPUs for 150k supervised steps. To jointly assess camera and depth accuracy, we unproject depth maps into 3D using the cameras and compute the $\ell_2$ distance between the predicted and ground-truth points, which we call *point error*. We use point error rather than Chamfer distance because nearest-neighbor matching over unordered point sets can be dominated by large surface regions such as walls and floors. All the models are trained on approximately the same number of tokens.

**Model and data size.** We observe that scaling either the model or the data consistently improves performance, as shown in Fig. 1. Increasing the number of training sequences in $10\times$ steps yields a monotonic drop in point error, from $0.275$ to $0.073$. Overall, the shape of both curves suggests that power laws might characterize scaling in this class of models.

**Register attention.** A variant that uses only global attention layers achieves a point error of $0.071$. Replacing $25\%$ of the global attention layers with register attention yields performance nearly identical to the original ($0.073$).

**Multi-task learning.** Removing the point and matching losses increases the point error from $0.073$ to $0.078$. For reference, adopting VGGT's original multi-head, multi-task setup achieves $0.070$ but requires multiple dense heads, making scaling difficult.

**Self-supervised training.** Replacing $10\%$ of training steps from supervised training with self-supervised training slightly reduces point error from $0.073$ to $0.070$. This performance gain stems from training on unlabeled data, which is more diverse. We also observe improved out-of-distribution generalization.

**Annotation quality.** To validate the quality of the pseudo ground truth produced by our annotation pipeline, we compare it against MegaSaM [99] on Sintel, which provides

synthetic camera and depth ground truth. For a fair comparison, we evaluate only the sequences and pixels that satisfy both our filtering criteria and MegaSaM's validation, excluding $8/23$ sequences and all dynamic regions. Under this protocol, our pipeline achieves $96.4\%$ AUC@30° for camera pose and $99.3\%$ $\delta_{1.25}$ for depth, compared with $62.1\%$ and $77.2\%$ for MegaSaM, respectively, confirming the high quality of the resulting annotations. Our goal in pseudo-label generation is not to maximize yield, but to retain only sequences and pixels that are very likely to be correct, as we find that a smaller set of highly accurate pseudo ground truth is more beneficial in practice than a larger but noisier collection. Hence, the annotation pipeline is intentionally conservative: if a sequence is even mildly ambiguous, or if a pixel cannot be validated reliably, we prefer to discard it rather than risk injecting noisy supervision.

### 4.4. Applications of Registers

**Robotics.** Recent work has explored the use of reconstruction models to improve spatial understanding in VLA systems [1, 93]. We evaluate whether VGGT-Ω can be a plug-and-play geometric encoder for VLA. Given the input images, we extract registers (scene tokens) from VGGT-Ω and concatenate them with the original OpenVLA-OFT input tokens [85]. We then train OpenVLA-OFT using its standard protocol, while keeping VGGT-Ω fixed throughout. As shown in Tab. 3, the geometry-aware registers consistently improve performance across all LIBERO tasks [104].

Table 3. **Performance on the LIBERO benchmark.** We freeze our pretrained model, feed the scene tokens as additional input to OpenVLA-OFT, and report the success rate (SR) (higher is better). The clear gains validate the effectiveness of our scene tokens in aggregating spatial information.

| Method | Spatial SR (%) | Object SR (%) | Goal SR (%) | Long SR (%) | Average SR (%) |
|---|---|---|---|---|---|
| Diffusion Policy | 78.3 | 92.5 | 68.3 | 50.5 | 72.4 |
| TraceVLA | 84.6 | 85.2 | 75.1 | 54.1 | 74.8 |
| Octo | 78.9 | 85.7 | 84.6 | 51.1 | 75.1 |
| OpenVLA | 84.7 | 88.4 | 79.2 | 53.7 | 76.5 |
| Dita | 84.2 | 96.3 | 85.4 | 63.8 | 82.4 |
| CoT-VLA | 87.5 | 91.6 | 87.6 | 69.0 | 83.9 |
| $\pi_0$-FAST | 96.4 | 96.8 | 88.6 | 60.2 | 85.5 |
| $\pi_0$ | 96.8 | 98.8 | 95.8 | 85.2 | 94.2 |
| UniVLA | 96.5 | 96.8 | 95.6 | 92.0 | 95.2 |
| OpenVLA-OFT | 97.6 | 98.4 | 97.9 | 94.5 | 97.1 |
| OpenVLA-OFT + Our Frozen Scene Tokens | **99.3** | **99.2** | **99.0** | **96.7** | **98.5** |

**Language Alignment.** To further verify whether the registers contain high-level information, we investigate whether they can be aligned to natural language. The procedure, illustrated in Fig. 8, follows the spirit of CLIP-style contrastive alignment [133, 216]. However, to keep it as simple as possible, we use a well-trained VGGT-Ω, fix the language encoder and only fine-tune our model.

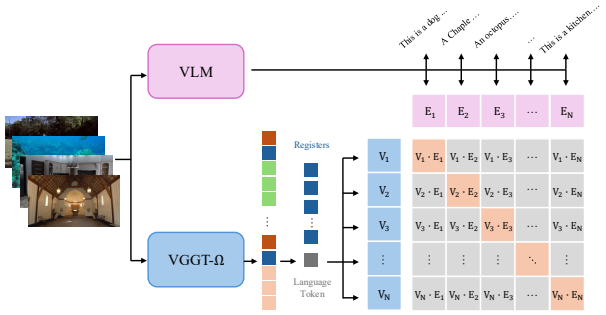


Figure 8. **Language Alignment.** We conduct language alignment to verify whether the registers contain high-level information. For each sequence, a VLM describes the scene content, coarse layout, and appearance, and the generated text tokens are mean-pooled to form the language embedding. On the VGGT-Ω side, a small self-attention stack takes the registers and a learnable language token as input, and the output language token is projected to form the register-derived embedding. The two embeddings are optimized with a symmetric InfoNCE loss over all sequence-description pairs in the global batch.

In detail, we use a VLM and VGGT-Ω to separately extract a global descriptor for a given image sequence and then align the two. To extract this global descriptor, the VLM observes all input views and is prompted to describe the scene content, coarse layout, and appearance as a single coherent scene. Then, the hidden states of the text tokens generated by the VLM are mean-pooled and $\ell_2$-normalized. To obtain a global descriptor from VGGT-Ω, a small self-attention stack takes the registers and a new learnable language token (randomly initialized) as input. The output language token is projected and $\ell_2$-normalized to produce the register-derived embedding from VGGT-Ω.

It is worth noting that the language token we introduced never directly observes image patch tokens, as it can only read out the registers. Thus, a successful alignment indicates that the registers themselves carry scene-level information that can be matched to language.

We maximize the cosine similarity of the matched register-derived embeddings and VLM language embeddings while minimizing the similarity of mismatched pairs with a symmetric InfoNCE loss. We fine-tune the models on the same image sequences as in the main reconstruction training. As in distributed CLIP training, embeddings are gathered across GPUs so that the global batch provides in-batch negatives. The VLM is frozen, and VGGT-Ω is fine-tuned end-to-end with a constant learning rate of $1 \times 10^{-5}$, rather than kept fixed as in the robotics experiment.

After only 10K iterations with a small learning rate, the model already transfers well to language retrieval. For evaluation, we construct a benchmark of 100 manually curated internet videos spanning a diverse set of scenarios, including cooking, praying, car racing, and more.

Given the register-derived embedding of the aligned VGGT-Ω, we retrieve the paired language embedding for each candidate video using cosine similarity and report top-$K$ accuracy. Using the VLM embedding employed during alignment, the model achieves $76.8\%$ top-1 accuracy and $97.0\%$ top-3 accuracy. That is, the correct language description ranks among the top three candidates for almost all videos. To test whether this alignment transfers beyond the exact training target, we replace the VLM embedding with a text-only LLM embedding, without any additional training. Specifically, we prompt the VLM to generate a video description and feed only this description to the Qwen3 LLM [201]. The model still obtains $47.5\%$ top-1 accuracy and $77.8\%$ top-3 accuracy in this zero-shot transfer setting.

Overall, these results show that the registers carry high-level information, likely quite semantic, that can be aligned with language space. At the same time, there is no degradation on the geometric tasks after alignment fine-tuning. The example text description and the prompt are provided in the supplementary materials. This may indicate that representations learned by a strong geometry model can align naturally with those of other modalities, such as language. This observation is consistent with the platonic representation hypothesis [56, 71], which suggests that sufficiently capable models trained on different modalities tend to converge toward a shared representation space.

**General Applications.** Registers provide a general mechanism for extracting both sequence-level and frame-level representations from the model. This design naturally extends to additional prediction tasks by introducing task-specific learnable tokens. For sequence-level predictions, a learnable token can be concatenated to the register tokens and decoded into the desired output, similar to the language token discussed above. For frame-level predictions, the same token can be replicated across frames, concatenated with the corresponding registers, and used to produce per-frame estimates. We have conducted preliminary experiments on several such tasks, including metric scale estimation, gravity direction estimation, and human presence detection, and observed promising results. We leave the exploration of these possibilities to the community. From this perspective, a camera token can also be viewed as a register-like learnable token, with the main distinction that it is trained under direct camera supervision.

## 5. Further Insights

As we developed VGGT-Ω, we made several empirical observations that are not yet rigorously established but that we think are still useful to share with other researchers interested in developing similar models.

**Model Souping: Where Is the Information Stored?** To better understand the model's behavior, we use model souping [194] as a probe. Specifically, we fuse the public VGGT checkpoint into VGGT-Ω by directly averaging specific subsets of their weights, without any further training. Perhaps surprisingly, although the two models use different architectures (*e.g.*, VGGT uses DINOv2 with a patch size of 14, whereas VGGT-Ω uses DINOv3 with a patch size of 16 and register attention), the averaged models can still produce visually reasonable reconstructions.

By fusing different subsets of the weights, we can gain insights into where different types of information are stored in the model. Our first observation is that depth and field-of-view information is largely stored in the FFNs of the attention blocks, primarily in the frame-wise attention blocks and, to a lesser extent, in the global attention blocks. This is consistent with the observations in the language community [30, 52, 119]. For example, as discussed in Sec. B, the model can memorize idiosyncratic noisy signals in the training data, such as treating humans as part of the ground in certain scenarios. When we fuse the FFN weights of VGGT's frame-wise attention blocks into VGGT-Ω with a 50%-50% average, these errors are resolved. In contrast, fusing the Q/K/V projection weights does not produce the same effect. Camera extrinsic information appears to be encoded at a higher level and is not controlled solely by the FFN weights. Finally, the ability to generalize to a variable number of input frames is also closely related to the frame-wise attention blocks.

Figure 9. **Motion-Aware Representations.** We cluster PCA-reduced intermediate image tokens by $k$-means and observe that the resulting clusters consistently separate the moving dancer from the static crowd and background, with red indicating high-response regions. This suggests that VGGT-Ω learns motion-aware representations from reconstruction training, without explicit motion supervision.

**Motion Awareness.** Since the model can reconstruct dynamic sequences with large non-camera motion, we ask whether it learns to localize the moving regions. To this end, we analyze the feature distributions of its intermediate tokens. Specifically, we normalize intermediate image tokens across space and time, reduce their dimensionality with PCA, and group them using $k$-means. No labels, optical flow, or learned probes are used at any stage.

As shown in Fig. 9, one PCA cluster consistently tracks the moving dancer across all frames while the static crowd and background remain in the other clusters. This suggests

that the ability to discriminate motion emerges automatically as a byproduct of the reconstruction objective, without explicit motion supervision. Note also that the model is never given the temporal order of the input frames, either during training or inference.

We further examine how this signal evolves across layers. Early layers (*e.g.*, layer 4 in Fig. 9) produce the cleanest motion segmentation, isolating the dancer with minimal background response. Middle layers, such as layer 13, retain a weaker but still discernible motion signal. At the deepest layers (*e.g.*, layer 23), the same clustering procedure highlights all people in the scene, indicating that the representations become increasingly global and semantic.

**Normalizing Predictions.** In VGGT, we discussed whether predictions should be normalized online to a unit coordinate space. We further analyze this question here. Consistent with the observation in VGGT, once the model has converged, we observe no difference in quantitative performance between training with and without prediction normalization. The main benefit of normalization is qualitative: the final point clouds appear more spatially well spread. The drawback is that the optimization becomes less stable, with a steeper learning curve and a greater need for careful hyperparameter tuning to avoid gradient explosion. If prediction normalization is necessary, we suggest initializing from a pretrained model to stabilize training, as in PI3.

**DINO Initialization.** As discussed above, VGGT-Ω is initialized from DINOv3. We examined this choice and found that, without DINO initialization (*i.e.*, from scratch), the model can still converge to similar performance, but requires 4–8× more training iterations. DINOv3 substantially eases optimization and improves training efficiency.

**Invalid Area Prediction.** We experimented with an additional branch for predicting invalid regions, as in [83, 185], with the hope that it would help the model ignore regions such as sky and prevent sky pixels from sticking to foreground objects in the point cloud. Although the model predicted the invalid masks accurately, sky pixels still appeared in the foreground of the depth estimates, likely due to the lack of supervisory signals in these regions. Therefore, in line with the idea of using as few dense heads as possible, we did not include this predictor in our final model.

**Register Attention Only.** We also tested using register attention only, without any full global attention. In this setting, image tokens in one frame can never attend to image tokens in other frames. All the inter-frame information exchange relies on registers instead. Although this design reduces FLOPs to 6% of the original model, performance drops to the level of the original VGGT. This trade-off may be worth exploring for on-device applications.

**Auxiliary Inputs.** Theoretically, incorporating auxiliary inputs, such as temporal order, camera parameters, depth maps, or scale factors, can further enhance performance. However, we empirically observe that introducing these priors during pretraining, even when applied randomly or masked across training iterations, is often detrimental. Conversely, our preliminary experiments indicate that providing conditional auxiliary inputs exclusively during the fine-tuning phase is highly effective, improving task-specific performance without compromising the integrity of the learned representations. While a comprehensive exploration is beyond the scope of this paper, we believe this represents a promising direction for future research.

**Synthetic and Real Data Mixture.** Consistent with the observations in VGGT [181], we find that synthetic and real data play complementary roles during training. At a high level, although this distinction is not exact, synthetic data contributes more directly to accuracy, while real data improves generalization. The main role of real data is to adapt the model to real-world appearance and expose it to more diverse camera trajectories. In practice, we recommend sampling roughly 80% synthetic data and 20% real data in each epoch. If sufficiently clean synthetic annotations are available, *e.g.* after addressing the issues discussed in Sec. B, increasing the synthetic data ratio to around 90% may be even better.

**How to fine-tune VGGT/VGGT-Ω.** When fine-tuning VGGT/VGGT-Ω for reconstruction tasks in a new data domain, we recommend two practical choices. First, it is best to use a full learning-rate schedule, *i.e.*, linear warm-up followed by cosine decay. The peak learning rate should remain small, and the number of iterations need not be large, but completing the full schedule is typically preferable to using a constant learning rate. Second, we found that the aleatoric uncertainty (confidence) loss can be unstable during fine-tuning. We therefore recommend initially removing this loss, particularly when the fine-tuning dataset is very small. If the downstream task requires confidence estimates, the uncertainty head can be fine-tuned separately. When fine-tuning VGGT/VGGT-Ω for tasks besides reconstruction, we suggest increasing the warm-up ratio, *e.g.* from 5% to 10–15%, and use a full scheduling cycle.

**Self-Supervised Training.** In principle, self-supervised training is a powerful tool to scale training data. So far, we have found it useful for improving model generalization, especially for out-of-distribution data, however, it has had little impact on most benchmarks. It is non-trivial to do better. Concurrent work has made a similar observation [67]. We spent considerable time exploring other self-supervised protocols such as new view synthesis, including variants similar to RayZer [73] and E-RayZer [224], generating tokens instead of pixels, NeRF representations [7] or Gaussian

Splats [84]. We also tried masking image tokens, distinguishing objects across frames, incorporating temporal order, and related variants. Only the student-teacher approach helped, in our implementation. For example, we found methods like E-RayZer, which may work well with static scenes, struggle with dynamic ones. This was counterintuitive to us, because much of self-supervised training in 2D relies on some form of *invariance* (*e.g.*, image augmentation in DINO), and reconstruction should naturally encode such invariance. We expected self-supervised reconstruction to work from scratch, whereas our successful approach still requires a pretrained model. Therefore, although the teacher-student self-supervised training in Sec. 3.4 is not necessary to achieve our benchmark results, we still share it with the community, even though it makes our training pipeline look more complicated. Self-supervised reconstruction remains an open problem for the community. It may ultimately need to be integrated into a unified or omni-style model, as we discuss later.

**Dense Prediction Heads.** In principle, strong latent representations should allow dense prediction tasks to be decoded with MLPs alone, without relying on convolutional layers, as evidenced by recent results such as JiT [96] or LagerNVS [156]. We explored this idea for a time because it aligned with our goal of keeping the framework as simple and scalable as possible. However, we found that MLP-only heads consistently produced visible patch/block artifacts in the predicted depth maps, although they were often better than convolution-based alternatives in quantitative metrics, were much faster and more memory-efficient, and provided more stable gradients during training. These artifacts are especially noticeable to humans, as they introduce geometric discontinuities that would otherwise be smooth. We tried several remedies, including mipmap-style supervision [7] (introducing local depth variance), probabilistic modeling (estimating depth with a probabilistic mixture of various independent channels), and related variants, but they did not reliably remove the artifacts. Interestingly, the problem was much more prevalent in outdoor scenes than in indoor scenes, especially when the scene contained distant objects. We suspect that this behavior is related to the numerical distribution of the prediction target. For image generation tasks as studied by JiT, the output lies in a well-bounded numerical space. In contrast, the range of geometric quantities, such as depth, is effectively unbounded. This has motivated our trade-off, where we maintain a small number of convolutional layers followed by MLP layers. The shallow convolutions operate at low feature resolutions, such as $16 \times 16$ or $32 \times 32$, and introduce little computational overhead, while accounting for most of the observed improvement in spatial smoothness. Notably, we used the shallow layers of DPT to keep the architecture comparable to prior methods, but any feature-pyramid-style convolution architecture can address the patch-artifact problem. Although we use the trade-off for VGGT-$\Omega$, we still believe that MLP-only dense decoding heads are a promising and important direction for future exploration.

## 6. Discussion

We now discuss some decisions in designing VGGT-$\Omega$, which are in part motivated by how we think feed-forward models can be most beneficial to the community.

**Prioritizing Simplicity.** We identified several architectural modifications that can further improve performance. For example, while our model predicts the camera parameters in a single pass, better results can be obtained by using the iterative refinement design of the original VGGT [181]. The depth output can also be improved by injecting raw RGB values into the dense prediction head. Using these and other techniques, we observed further improvements of 4%–6% in AUC@3° and about 2% in $\delta_{1.25}$ across multiple datasets, and we expect that further task-specific modifications can yield even more gains. Even so, we deliberately decided to prioritize the overall simplicity of the model and the quality of the representation extracted by the feature backbone (aggregator). This choice is motivated by our observation that, once the backbone is well-trained, training a new or improved prediction head typically requires only 5–10K iterations. Additionally, maintaining the simplicity of the architecture yields a cleaner base model, which we expect will be easier for the community to build upon.

**Benefits of Feed-forward Reconstruction.** Compared to traditional reconstruction pipelines, feed-forward reconstruction has three important advantages: (a) efficiency, (b) robustness, and (c) representational power. First, it is substantially faster, from tens to hundreds of times in some settings, and the gap can become even larger by streaming or by using multiple GPUs in parallel. Second, it is much more robust than methods that rely heavily on explicit geometry optimization, like bundle adjustment. For example, it can handle cases with little or no parallax, which are difficult to handle with triangulation-based methods. Feed-forward reconstruction also readily extends to processing dynamic scenes, as shown in our experiments. Third, and perhaps most importantly, feed-forward reconstruction can extract versatile geometry-aware representations. 3D vision lacks powerful off-the-shelf general-purpose feature extractors comparable to the older VGG16 [148] or ResNet [59] for 2D vision, let alone to modern 2D vision or vision-language foundation models. As we show with several examples in this paper, feed-forward reconstruction is a promising proxy task for learning such a representation for 3D tasks.

Optimization-based reconstruction methods like COLMAP [140] still have their strengths. In well-conditioned settings, bundle adjustment can estimate the

camera parameters to extremely high precision, reducing angular errors to a few hundredths of a degree. Such accuracy remains valuable for applications like novel view synthesis with NeRFs [120] or Gaussian splatting [84]. Feed-forward reconstruction is not in conflict with optimization, instead, it can serve as a strong initialization for subsequent optimization procedures such as bundle adjustment. Even so, given the rapid progress of feed-forward reconstruction in the short time since VGGT was introduced, we remain optimistic about future accuracy improvements and the applicability of this paradigm.

**The Role of 3D/4D in the Era of Large Models.** Spatial understanding is an area of increasing interest in vision-language, vision-language-action, vision-action, and world models [17, 20, 28, 62, 86, 94, 122, 127, 163, 235]. This is a natural consequence of developing embodied agents that operate in the physical world, which is three-dimensional. In the short term, reconstruction models can be used as external tools to obtain explicit 3D information, such as depth and camera parameters. They can also provide 'structured' tokens that encode spatial information implicitly, as discussed in this paper. In the long term, reconstruction might become a citizen in future unified models ('omni-model'), *i.e.*, in large models pre-trained using multiple modalities, such as language and vision, and targeting multiple capabilities, such as understanding and generation [34, 162, 164, 167, 177, 188, 195, 229]. This could be achieved by adding reconstruction-oriented data to model training without significant changes to the architecture or training framework. For example, camera parameters can be predicted autoregressively as text, while pixel-wise quantities such as depth can be formulated as image generation, as recently explored in [202].

**The next generation of reconstruction models, and perhaps more broadly perception systems, may be built on unified models.** First, the biggest gain is likely to come from data, which is currently the main bottleneck for perception tasks like reconstruction. Large-scale text and video corpora contain rich implicit descriptions of the physical world. When geometry tasks are trained jointly with these modalities, they can tap into this massive, broader source of supervision. Second, most perception problems are severely under-constrained when viewed in isolation. A unified model naturally enforces cross-task consistency, allowing ambiguities in one domain (*e.g.*, textureless regions in depth estimation) to be resolved by priors from another (*e.g.*, semantic context). Third, there is a paradigm-level reason to move in this direction: recent observations may indicate generative vision models scale more easily than perception-only vision models, and it seems generative models can transfer to perception to some extent [48]. We expect stronger, more robust perception models to emerge from multi-modal, multi-task training, rather than from reconstruction or perception objectives in isolation.

## 7. Conclusion

We presented VGGT-Ω, a feed-forward reconstruction model that achieves strong results across static and dynamic benchmarks. We improved the original VGGT in terms of architecture, data, and training by introducing register attention, using a single dense prediction head with multi-task losses, a large-scale annotation pipeline that handles dynamic content, and a self-supervised training protocol that leverages vast amounts of unlabeled videos. These ingredients allowed us to train our model at an unprecedented scale. Empirically, we found that VGGT-Ω scales predictably with model capacity and data size. Beyond geometry, we found that the learned registers carry useful global information, improving VLA models and supporting alignment with language. We hope VGGT-Ω will be a useful foundation for the community to build on.

**Acknowledgments.** Jianyuan Wang was supported by Facebook AI Research. Christian Rupprecht was partially supported by ERC starting grant 'Volute' (No. 101222037). Many people helped bring this work together. Please see our project page for full acknowledgments.

## References

[1] Ali Abouzeid, Malak Mansour, Qinbo Sun, Zezhou Sun, and Dezhen Song. Geoaware-VLA: Implicit geometry aware vision-language-action model. *arXiv*, 2025. 1, 3, 11

[2] 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), 2011. 3

[3] Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *Proc. ECCV*, 2020. 7

[4] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *Proc. ECCV*, 2022. 7

[5] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *Proc. CVPR*, 2022. 8

[6] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. Introducing HOT3D: an egocentric dataset for 3D hand and object tracking. *arXiv*, 2406.09598, 2024. 7

[7] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan.

Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. ICCV*, 2021. 13, 14

[8] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: a synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proc. CVPR*, 2023. 7

[9] Tyler Bonnen, Jitendra Malik, and Angjoo Kanazawa. Human-level 3D shape perception emerges from multi-view learning. *arXiv*, 2026. 1, 3

[10] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *Proc. ECCV*, 2024. 3

[11] Jelena Bratulić, Sudhanshu Mittal, Thomas Brox, and Christian Rupprecht. On geometric understanding and learned data priors in VGGT. *arXiv*, 2025. 3

[12] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. CVPR*, 2000. 4, 6

[13] Leo Breiman. Random forests. *Machine learning*, 45(1), 2001. 7

[14] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. ECCV*, 2012. 8

[15] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv*, 2001.10773, 2020. 7

[16] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *Proc. CVPR*, 2025. 3

[17] Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. DepthLM: Metric depth from vision language models. In *Proc. ICLR*, 2026. 15

[18] Yang Cao, Feize Wu, Dave Zhenyu Chen, Yingji Zhong, Lanqing Hong, and Dan Xu. VGGT-Det: Mining VGGT internal priors for sensor-geometry-free multi-view indoor 3D object detection. *arXiv*, 2026. 1, 3

[19] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021. 2, 6

[20] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proc. CVPR*, 2024. 15

[21] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *Proc. CVPR*, 2021. 3

[22] Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. Geometric context transformer for streaming 3D reconstruction. *arXiv*, 2026. 3

[23] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proc. Knowledge Discovery and Data Mining*, 2016. 7

[24] Tianrun Chen, Yuanqi Hu, Yidong Han, Hanjie Xu, Deyi Ji, Qi Zhu, Chunan Yu, Xin Zhang, Cheng Chen, Chaotao Ding, Ying Zang, Xuanfu Li, et al. HD-VGGT: High-resolution visual geometry transformer. *arXiv*, 2026. 3

[25] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. *arXiv*, 2025. 3

[26] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. In *Proc. ICLR*, 2026. 3

[27] Yue Chen, Xingyu Chen, Yuxuan Xue, Anpei Chen, Yuliang Xiu, and Gerard Pons-Moll. Human3R: Everyone everywhere all at once. *arXiv*, 2025. 4

[28] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded spatial reasoning in vision-language models. In *Proc. NeurIPS*, 2024. 15

[29] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. CVPR*, 2017. 7

[30] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proc. ACL*, 2022. 12

[31] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *IJCV*, 107, 2014. 6

[32] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *Proc. ICLR*, 2024. 2, 4

[33] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction, 2012. 3

[34] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv*, 2505.14683, 2025. 15

[35] Junyuan Deng, Heng Li, Tao Xie, Weiqiang Ren, Qian Zhang, Ping Tan, and Xiaoyang Guo. SAIL-Recon: Large SfM by augmenting scene regression with localization. In *Proc. 3DV*, 2026. 3

[36] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. VGGT-Long: chunk it, loop it, align it—pushing vggt's limits on kilometer-scale long RGB sequences. *arXiv*, 2025. 3

[37] Tianchen Deng, Xun Chen, Ziming Li, Hongming Shen, Danwei Wang, Javier Civera, and Hesheng Wang. Unipr-3d: Towards universal visual place recognition with visual geometry grounded transformer. *arXiv*, 2025. 3

[38] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: self-supervised interest point detection and description. In *Proc. CVPR Workshop*, 2018. 3, 7

[39] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud.

MASt3R-SfM: a fully-integrated solution for unconstrained structure-from-motion. In *Proc. 3DV*, 2025. 3

[40] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *Proc. CVPR*, 2019. 3

[41] Sven Elflein, Qunjie Zhou, and Laura Leal-Taixé. Light3R-SfM: towards feed-forward structure-from-motion. In *Proc. CVPR*, 2025. 3

[42] Xianze Fang, Jingnan Gao, Zhe Wang, Zhuo Chen, Xingyu Ren, Jiangjing Lyu, Qiaomu Ren, Zhonglei Yang, Xiaokang Yang, Yichao Yan, and Chengfei Lyu. Dens3R: A foundation model for 3D geometry prediction. *Proc. ICLR*, 2026. 3

[43] Olivier D. Faugeras and Stephen J. Maybank. Motion from point matches: Multiplicity of solutions. *IJCV*, 4(3), 1990. 2

[44] 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. In *Proc. ICCV*, 2025. 4

[45] Michael Fonder and Marc Van Droogenbroeck. Mid-Air: A multi-modal dataset for extremely low altitude drone flights. In *Proc. CVPR Workshop*, 2019. 7

[46] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building Rome on a cloudless day. In *Proc. ECCV*, 2010. 3

[47] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-Neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. In *Proc. NeurIPS*, 2022. 3

[48] Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T. Barron, Kyle Genova, Nithish Kannen, et al. Image generators are generalist vision learners. *arXiv*, 2026. 15

[49] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Proc. NeurIPS*, 2022. 8

[50] Jingnan Gao, Zhe Wang, Xianze Fang, Xingyu Ren, Zhuo Chen, Shengqi Liu, Yuhao Cheng, Jiangjing Lyu, Xiaokang Yang, and Yichao Yan. MoRE: 3D visual geometry reconstruction meets mixture-of-experts. *arXiv*, 2025. 3

[51] Yulu Gao, Bohao Zhang, Zongheng Tang, Jitong Liao, Wenjun Wu, and Si Liu. VGGT-Segmentor: Geometry-enhanced cross-view segmentation. *arXiv*, 2026. 1, 3

[52] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proc. Empirical Methods in Natural Language Processing*, 2021. 12

[53] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, et al. Kubric: a scalable dataset generator. In *Proc. CVPR*, 2022. 7

[54] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proc. CVPR*, 2020. 3

[55] Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. $D^2$USt3R: Enhancing 3D reconstruction with 4d pointmaps for dynamic scenes. *arXiv*, 2504.06264, 2025. 4

[56] Junlin Han, Shengbang Tong, David Fan, Yufan Ren, Koustuv Sinha, Philip Torr, and Filippos Kokkinos. Learning to see before seeing: Demystifying llm visual priors from language pre-training. In *Proc. NeurIPS Workshop*, 2025. 12

[57] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1

[58] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004. 2

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 14

[60] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, 2020. 2, 6

[61] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. *arXiv*, 2405.14868, 2024. 7

[62] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv*, 2023. 15

[63] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. SAIL-VOS 3D: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *Proc. CVPR*, 2021. 7

[64] Guichen Huang, Ruoyu Wang, Xiangjun Gao, Che Sun, Yuwei Wu, Shenghua Gao, and Yunde Jia. LongSplat: Online generalizable 3D Gaussian splatting from long sequence images. In *Proc. ICCV*, 2025. 3

[65] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, et al. ViPE: video pose engine for 3D geometric perception. *arXiv*, 2508.10934, 2025. 4, 27

[66] Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. Gen3R: 3D scene generation meets feed-forward reconstruction. In *Proc. CVPR*, 2026. 1, 3

[67] Nan Huang, Pengcheng Yu, Weijia Zeng, James M Rehg, Angjoo Kanazawa, Haiwen Feng, and Qianqian Wang. Self-improving 4d perception via self-distillation. *arXiv preprint arXiv:2604.08532*, 2026. 13

[68] Nan Huang, Pengcheng Yu, Weijia Zeng, James M Rehg, Angjoo Kanazawa, Haiwen Feng, and Qianqian Wang.

Self-improving 4D perception via self-distillation. *arXiv*, 2026. 3

[69] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *Proc. CVPR*, 2018. 7

[70] Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. *arXiv*, 2026. 28

[71] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proc. ICML*, 2024. 12

[72] Wooseok Jang, Seonghu Jeon, Jisang Han, Jinhyeok Choi, Minkyung Kwon, Seungryong Kim, Saining Xie, and Sainan Liu. Repurposing geometric foundation models for multi-view diffusion. *arXiv*, 2026. 1, 3

[73] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: a self-supervised large view synthesis model. In *Proc. ICCV*, 2025. 13

[74] Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, Jiuxiang Gu, Qixing Huang, et al. MegaSynth: Scaling up 3d scene reconstruction with synthesized data. In *Proc. CVPR*, 2025. 7

[75] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: feed-forward 3D Gaussian Splatting from unconstrained views. *arXiv*, 2505.23716, 2025. 3

[76] Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *arXiv*, 2025. 2, 4

[77] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4D: Leveraging video generators for geometric 4D scene reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 4

[78] Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Holynski. ZipMap: linear-time stateful 3D reconstruction via test-time training. In *Proc. CVPR*, 2026. 3

[79] Dongki Jung, Jaehoon Choi, Yonghan Lee, Sungmin Eum, Heesung Kwon, and Dinesh Manocha. MoRe: Monocular geometry refinement via graph optimization for cross-view consistency. In *Proc. WACV*, 2026. 3

[80] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv*, 2020. 1

[81] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[82] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025. 4

[83] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, et al. MapAnything: universal feed-forward metric 3D reconstruction. In *Proc. 3DV*, 2026. 3, 13

[84] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *Proc. SIGGRAPH*, 42(4), 2023. 14, 15

[85] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proc. RSS*, 2025. 11

[86] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, et al. OpenVLA: an open-source vision-language-action model. In *Proc. CoRL*, 2025. 15

[87] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. STream3R: Scalable sequential 3D reconstruction with causal transformer. In *Proc. ICLR*, 2026. 3

[88] Alexander Lappe and Martin A Giese. Register and [CLS] tokens induce a decoupling of local and global features in large ViTs. In *Proc. NeurIPS*, 2025. 4

[89] Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. Eden: Multimodal synthetic dataset of enclosed garden scenes. In *Proc. WACV*, pages 1579–1589, 2021. 7

[90] Seonho Lee, Jiho Choi, Inha Kang, Jiwook Kim, Junsung Park, and Hyunjung Shim. 3D-aware vision-language models fine-tuning with geometric distillation. In *Proc. EMNLP*, pages 10628–10647, 2025. 1, 3

[91] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with MASt3R. In *Proc. ECCV*, 2024. 1

[92] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, et al. BEHAVIOR-1K: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv*, 2024. 7

[93] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv*, 2025. 1, 3, 11

[94] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Robotics: Science and Systems*, 2026. 15

[95] Samuel Li, Pujith Kachana, Prajwal Chidananda, Saurabh Nair, Yasutaka Furukawa, and Matthew Brown. Rig3R: Rig-aware conditioning for learned 3D reconstruction. In *Proc. NeurIPS*, 2025. 3

[96] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv*, 2026. 14

[97] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proc. CVPR*, 2018. 7

[98] Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, Zizhen Li, Fanrui Zhang, et al. Sekai: A video dataset towards world exploration. *arXiv*, 2025. 28

[99] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2025. 4, 10, 27

[100] Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. WinT3R: Window-based streaming reconstruction with camera token pool. In *Proc. ICLR*, 2025. 3

[101] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv*, 2511.10647, 2025. 1, 2, 3, 4, 6

[102] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: local feature matching at light speed. In *Proc. ICCV*, 2023. 3, 7

[103] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, et al. DL3DV-10K: a large-scale scene dataset for deep learning-based 3d vision. In *Proc. CVPR*, 2024. 7

[104] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In *Proc. NeurIPS*, 2023. 11

[105] Changkun Liu, Bin Tan, Zeran Ke, Shangzhan Zhang, Jiachen Liu, Ming Qian, Nan Xue, Yujun Shen, and Tristan Braud. PLANA3R: Zero-shot metric planar 3D reconstruction via feed-forward planar splatting. *arXiv*, 2025. 3

[106] Shaohui Liu, Yidan Gao, Tianyi Zhang, Rémi Pautrat, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Robust incremental structure-from-motion with hybrid features. In *Proc. ECCV*, 2024. 3

[107] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. *Proc. ECCV*, 2024. 7

[108] Xingchen Liu, Piyush Tayal, Jianyuan Wang, Jesus Zarzar, Tom Monnier, Konstantinos Tertikas, Jiali Duan, Antoine Toisoul, Jason Y. Zhang, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, et al. uCO3D uncommon objects in 3D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 7

[109] Xuanyi Liu, Deyi Ji, Liqun Liu, Lanyun Zhu, Xuhang Chen, Qianxiong Xu, Peng Shu, Huan Yu, Jie Jiang, Feng Gao, and Siwei Ma. CamGeo: Sparse camera-conditioned image-to-video generation with 3D geometry priors. In *Proc. ICML*, 2026. 3

[110] Xuanyi Liu, Chunan Yu, Deyi Ji, Qi Zhu, Lingyun Sun, Xuanfu Li, Jin Ma, Tianrun Chen, and Lanyun Zhu. StreamCacheVGGT: Streaming visual geometry transformers with robust scoring and hybrid cache compression. *arXiv*, 2026. 3

[111] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999. 7

[112] 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 *Proc. CVPR*, 2025. 4

[113] Zeyu Ma, Zachary Teed, and Jia Deng. Multiview stereo with cascaded epipolar RAFT. In *Proc. ECCV*, 2022. 3

[114] Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the $sl(4)$ manifold. In *Proc. NeurIPS*, 2025. 3

[115] Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3R: Training-free token eviction for memory-bounded streaming visual geometry transformers. *arXiv*, 2025. 3

[116] Mapillary. Metropolis dataset, 2025. 7

[117] Alexis Marouani, Oriane Siméoni, Hervé Jégou, Piotr Bojanowski, and Huy V. Vo. Revisiting [CLS] and patch token interaction in vision transformers. *arXiv*, 2026. 2, 4

[118] Kirill Mazur, Marwan Taher, and Andrew J. Davison. 4D Primitive-Mâché: Glueing primitives for persistent 4D scene reconstruction. In *Proc. CVPR*, 2026. 3

[119] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Proc. NeurIPS*, 2022. 12

[120] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, 2020. 15

[121] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proc. CVPR*, 2020. 3

[122] NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, et al. Cosmos world foundation model platform for physical ai. *arXiv*, 2501.03575, 2025. 15

[123] John Oliensis. A critique of structure-from-motion algorithms. *CVIU*, 80(2), 2000. 2

[124] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 2, 6

[125] Onur Özyesil, V. Voroninski, R. Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26, 2017. 2

[126] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng (Carl) Ren. Aria digital twin: A new benchmark dataset for egocentric 3D machine perception. In *Proc. ICCV*, 2023. 7

[127] Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models, 2025. 15

[128] 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 *Proc.IROS*, 2025. 7

[129] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *Proc. CVPR*, 2022. 3

[130] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. In *Proc. NeurIPS*, 2018. 7

[131] Dexin Qi, Tao Tao, Zhihong Zhang, and Xuesong Mei. Fupad: Scalable pose estimation by fusing patch-wise vggt with dense bundle adjustment. In *Proc. ICIRA*, 2025. 3

[132] Kangan Qian, ChuChu Xie, Yang Zhong, Jingrui Pang, Siwen Jiao, Sicong Jiang, Zilin Huang, Yunlong Wang, Kun Jiang, Mengmeng Yang, Hao Ye, Guanghao Zhang, et al. Xembodied: A foundation model with enhanced geometric and physical cues for large-scale embodied environments. *arXiv*, 2026. 1, 3

[133] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021. 11

[134] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proc. ICCV*, 2021. 5

[135] Zhifeng Rao, Wenlong Chen, Lei Xie, Xia Hua, Dongfu Yin, Zhen Tian, and F. Richard Yu. AugVLA-3D: Depth-driven feature augmentation for vision-language-action models. In *arXiv*, 2026. 1, 3

[136] 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 *Proc. ICCV*, 2021. 7

[137] 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 *Proc. ICCV*, 2021. 7

[138] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: learning feature matching with graph neural networks. In *Proc. CVPR*, 2020. 3, 7

[139] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proc. ICCV*, 2019. 7

[140] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 1, 3, 4, 7, 14, 27

[141] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 7

[142] 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 *Proc. CVPR*, 2017. 8

[143] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: training-free acceleration of visual geometry transformer. In *Proc. ICLR*, 2026. 2, 3, 5

[144] Cheng Shi, Yizhou Yu, and Sibei Yang. Vision transformers need more than registers. *arXiv*, 2026. 4

[145] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. ClusterGNN: cluster-based coarse-to-fine graph neural network for efficient feature matching. In *Proc. CVPR*, 2022. 3

[146] 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 *Proc. CVPR*, 2013. 8

[147] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, et al. DINOv3. *arXiv preprint*, 2025. 1, 2, 4, 6, 8

[148] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 14

[149] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Trans. on Graphics (TOG)*, 2006. 1, 3

[150] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv*, 2019. 7

[151] Julian Straub, Daniel DeTone, Tianwei Shen, Nan Yang, Chris Sweeney, and Richard Newcombe. EFM3D: A benchmark for measuring progress towards 3D egocentric foundation models. *arXiv*, 2024. 7

[152] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proc.IROS*, 2012. 8

[153] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic Point Maps: A versatile representation for dynamic 3D reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025. 4, 6

[154] Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-DPM: Video reconstruction with dynamic point maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 4, 6

[155] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. CVPR*, 2020. 7

[156] Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. LagerNVS: Latent geometry for fully neural real-time novel view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 1, 3, 14

[157] Marwan Taher, Ignacio Alzugaray, Kirill Mazur, Xin Kong, and Andrew J. Davison. KV-Tracker: Real-time pose tracking with transformers. *arXiv*, 2025. 3

[158] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. In *Proc. ICLR*, 2019. 3

[159] 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 *Proc. CVPR*, 2025. 3

[160] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proc. NeurIPS*, 2017. 2

[161] Jonathan Taylor, Allan D Jepson, and Kiriakos N Kutulakos. Non-rigid structure from locally-rigid motion. In *Proc. CVPR*, 2010. 6

[162] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv*, 2024. 15

[163] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, et al. Gemini robotics: Bringing ai into the physical world. *arXiv*, 2025. 15

[164] Meituan LongCat Team. LongCat-Next: Lexicalizing modalities as discrete tokens. *arXiv*, 2026. 15

[165] Zachary Teed and Jia Deng. DeepV2D: video to depth with differentiable structure from motion. In *Proc. ICLR*, 2020. 3

[166] Zachary Teed and Jia Deng. DROID-SLAM: deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Proc. NeurIPS*, 2021. 3

[167] Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, Naila Murray, Marjan Ghazvininejad, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv*, 2603.03276, 2026. 15

[168] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30(5), 2008. 4

[169] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. SMD-Nets: Stereo mixture density networks. In *Proc. CVPR*, 2021. 7

[170] MJ Tyszkiewicz, P Fua, and E Trulls. DISK: learning local features with policy gradient. In *Proc. NeurIPS*, 2020. 3

[171] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: depth and motion network for learning monocular stereo. In *Proc. CVPR*, 2017. 3

[172] An Dinh Vuong, Minh Nhat Vu, and Ian Reid. Improving robotic manipulation with efficient geometry-aware vision encoder. In *arXiv*, 2025. 1, 3

[173] Chongyu Wang, Ting Huang, Chunyu Sun, Xinyu Ning, Di Wang, and Hao Tang. Let geometry GUIDE: Layer-wise unrolling of geometric priors in multimodal LLMs. *arXiv*, 2026. 1, 3

[174] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster VGGT with block-sparse global attention. *arXiv*, 2025. 2, 3, 5

[175] Hengyi Wang and Lourdes Agapito. Spann3R: 3D reconstruction with spatial memory. In *Proc. 3DV*, 2024. 3

[176] Hengyi Wang and Lourdes Agapito. AMB3R: accurate feed-forward metric-scale 3D reconstruction with backend. *arXiv*, 2511.20343, 2025. 3

[177] Haifeng Wang, Hua Wu, Tian Wu, Yu Sun, Jing Liu, Dianhai Yu, Yanjun Ma, Jingzhou He, Zhongjun He, Dou Hong, Qiwen Liu, Shuohuan Wang, et al. Ernie 5.0 technical report. *arXiv*, 2026. 15

[178] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proc. CVPR*, 2021. 3

[179] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: solving pose estimation via diffusion-aided bundle adjustment. In *Proc. ICCV*, 2023.

[180] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: visual geometry grounded deep structure from motion. In *Proc. CVPR*, 2024. 3, 4, 7

[181] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 3, 4, 8, 13, 14, 24

[182] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, Xiaoxiao Long, Hao Zhu, et al. SpatialVID: a large-scale video dataset with spatial annotations. *arXiv*, 2509.09676, 2025. 28

[183] Lijie Wang, Lianjie Guo, Ziyi Xu, Qianhao Wang, Fei Gao, and Xieyuanli Chen. LiDAR-VGGT: Cross-modal coarse-to-fine fusion for globally consistent and metric-scale dense mapping. *IEEE Robotics and Automation Letters (RA-L)*, 2026. 3

[184] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proc. CVPR*, 2025. 3, 4

[185] 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. *arXiv*, 2410.19115, 2024. 13

[186] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proc. CVPR*, 2024. 1, 3, 6, 24

[187] 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 *Proc.IROS*, 2020. 7

[188] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, et al. Emu3: Next-token prediction is all you need. *arXiv*, 2024. 15

[189] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning. *arXiv*, 2507.13347, 2025. 1, 3, 4

[190] Yunnan Wang, Kecheng Zheng, Jianyuan Wang, Minghao Chen, David Novotny, Christian Rupprecht, Yinghao Xu, Xing Zhu, Wenjun Zeng, Xin Jin, and Yujun Shen. SceneScribe-1M: A large-scale video dataset with comprehensive geometric and semantic annotations. In *Proc. CVPR*, 2026. 28

[191] Zichen Wang, Ang Cao, Liam J Wang, and Jeong Joon Park. MoE3D: A mixture-of-experts module for 3D reconstruction. *arXiv*, 2026. 3

[192] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. DeepSFM: structure from motion via deep bundle adjustment. In *Proc. ECCV*, 2020. 3

[193] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proc. ICCV*, 2021. 3

[194] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proc. ICML*, 2022. 12

[195] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proc. CVPR*, 2025. 15

[196] Changli Wu, Haodong Wang, Jiayi Ji, Yutian Yao, Chunsai Du, Jihua Kang, Yanwei Fu, and Liujuan Cao. MVGGT: Multimodal visual geometry grounded transformer for multiview 3D referring expression segmentation. *arXiv*, 2026. 1, 3

[197] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In *Proc. NeurIPS*, 2025. 3, 4

[198] 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 *Proc. CVPR*, 2024. 7

[199] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: 3D point tracking made easy. In *Proc. ICCV*, 2025. 3, 4

[200] Jingyi Xu, Zhangshuo Qi, Zhongmiao Yan, Xuyu Gao, Qianyun Jiao, Songpengcheng Xia, Xieyuanli Chen, and Ling Pei. VGGT-MPR: VGGT-enhanced multimodal place recognition in autonomous driving environments. *arXiv*, 2026. 1, 3

[201] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report. *arXiv*, 2025. 12

[202] Ceyuan Yang, Zhijie Lin, Yang Zhao, Fei Xiao, Hao He, Qi Zhao, Chaorui Deng, Kunchang Li, Zihan Ding, Yuwei Guo, Fuyun Wang, Fangqi Zhu, et al. Context unrolling in omni models. *arXiv*, 2026. 15

[203] 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. *Proc. CVPR*, 2025. 3

[204] Songlin Yang, Tianyi Wei, Yushi Lan, Zeqi Xiao, Anyi Rao, and Xingang Pan. Dense semantic matching with VGGT prior. *arXiv*, 2025. 1, 3

[205] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L Brown II, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Rob Fergus, Yann LeCun, et al. Cambrian-S: Towards spatial supersensing in video. In *Proc. ICLR*, 2026. 1, 3

[206] David Yifan Yao, Albert J. Zhai, and Shenlong Wang. Uni4D: unifying visual foundation models for 4D modeling from a single video. *arXiv*, 2503.21761, 2025. 4

[207] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *Proc. ECCV*, 2018. 3

[208] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Proc. NeurIPS*, 2020. 3

[209] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: a high-fidelity dataset of 3d indoor scenes. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 7

[210] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: learned invariant feature transform. In *Proc. ECCV*, 2016. 3

[211] Yuheng Yuan, Qiuhong Shen, Shizun Wang, Xingyi Yang, and Xinchao Wang. Test3R: Learning to reconstruct 3D at test time. In *Proc. NeurIPS*, 2025. 3

[212] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proc. CVPR*, 2018. 7

[213] Ying Zang, Yidong Han, Chaotao Ding, Yuanqi Hu, Deyi Ji, Qi Zhu, Xuanfu Li, Jin Ma, Lingyun Sun, Tianrun Chen, and Lanyun Zhu. Robust 4D visual geometry transformer with uncertainty-aware priors. *arXiv*, 2026. 3

[214] Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, Xing Wei,

and Ning Guo. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. *arXiv*, 2025. 1, 3

[215] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proc. CVPR*, pages 12104–12113, 2022. 1

[216] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proc. CVPR*, 2023. 11

[217] Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, et al. Efficiently reconstructing dynamic scenes one d4rt at a time. *arXiv*, 2512.08924, 2025. 4

[218] 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*, 2410.03825, 2024. 4, 6

[219] Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: long-context geometric reconstruction with hybrid memory. *arXiv*, 2026. 3

[220] Jiang Zhang, Shijie Zhou, Bangya Liu, Achuta Kadambi, and Zhiwen Fan. Spatialstack: Layered geometry-language fusion for 3d vlm spatial reasoning. *arXiv*, 2026. 1, 3

[221] 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 *Proc. CVPR*, 2025. 3

[222] Shihua Zhang, Qiuhong Shen, Shizun Wang, Tianbo Pan, and Xinchao Wang. Make geometry matter for spatial reasoning. *arXiv*, 2026. 1, 3

[223] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. GeoMVSNet: Learning multi-view stereo with geometry perception. In *Proc. CVPR*, 2023. 3

[224] Qitao Zhao, Hao Tan, Qianqian Wang, Sai Bi, Kai Zhang, Kalyan Sunkavalli, Shubham Tulsiani, and Hanwen Jiang. E-RayZer: Self-supervised 3D reconstruction as spatial visual pre-training. In *Proc. CVPR*, 2026. 13

[225] Ruosen Zhao, Zhikang Zhang, Jialei Xu, Jiahao Chang, Dong Chen, Lingyun Li, Weijian Sun, and Zizhuang Wei. SpaceMind: Camera-guided modality fusion for spatial reasoning in vision-language models. *arXiv*, 2025. 1, 3

[226] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. ALIKED: a lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Trans. on Instrumentation and Measurement*, 72, 2023. 7

[227] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3D world: Enhancing mllms with 3D vision geometry priors. *arXiv*, 2025. 1, 3

[228] 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 *Proc. CVPR*, 2023. 7

[229] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *Proc. ICLR*, 2025. 15

[230] Changqing Zhou, Yueru Luo, and Changhao Chen. Generalizing visual geometry priors to sparse gaussian occupancy prediction. *arXiv*, 2026. 1, 3

[231] Kaichen Zhou, Yuhan Wang, Grace Chen, Xinhai Chang, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. PAGE-4D: disentangled pose and geometry estimation for 4D perception. *arXiv*, 2510.17568, 2025. 4

[232] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017. 3

[233] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, Mingyu Liu, Dingning Liu, et al. OmniWorld: A multi-domain and multi-modal dataset for 4D world modeling. In *Proc. ICLR*, 2026. 28

[234] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *Proc. ICLR*, 2026. 3

[235] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proc. CoRL*, 2023. 15

# VGGT-Ω

## Supplementary Material

In this Supplement, we provide additional implementation details in Sec. A, a discussion of common data issues in Sec. B, and a discussion of limitations in Sec. C.

## A. Additional Details

In this section, we detail the training setup and architecture (Sec. A.1), the construction of positive and negative pairs (Sec. A.2), the VLM prompt (Sec. A.3), the annotation filtering criteria (Sec. A.4), and the language alignment procedure (Sec. A.5).

### A.1. Training and Architecture

To train our model, we set the loss weights in Eq. (1) to $\lambda_{\text{cam}} = 5.0$, $\lambda_{\text{depth}} = 1.0$, $\lambda_{\text{point}} = 0.5$, and $\lambda_{\text{match}} = 0.1$. For each scene, following [181, 186], we normalize the ground truth to a unit space. Concretely, we first transform all quantities into the first camera's coordinate frame and compute the average distance of all 3D points to the origin; we then scale the depth maps and translation vectors by this value. As in [181], this normalization is applied only to the ground truth and not to the predictions. To stabilize training, we apply gradient-norm clipping with a threshold of $1.0$ and use QKNorm inside the attention layers. At each iteration, we randomly sample the number of input frames from the range $[1, 24]$ and correspondingly set the batch size to saturate GPU memory. Each frame is independently masked by a rectangle whose height and width are uniformly sampled from $[32, 128]$ pixels, with a probability of $0.05$. Pixels inside the masked region are set to black, and the corresponding depth values are marked as invalid. For color jittering, we use PyTorch's implementation with brightness $0.5$, contrast $0.5$, saturation $0.5$, and hue $0.1$. During self-supervised training, the weights of the teacher model are updated using an exponential moving average with a decay of $0.999$. Our camera head follows the implementation in [181], using a ReLU activation for the focal length and no activation for the quaternion and translation parameters. For datasets with temporally ordered sequences (*e.g.*, videos), we sample frames from a local temporal window rather than based on covisibility. This can produce particularly challenging reconstruction subsets in which images are semantically related but share little or no visual overlap. We found that these samples improve the model's generalization.

### A.2. Constructing Positive and Negative Pairs

We expand on the matching loss introduced in the main paper (Sec. 3.2) by detailing how we construct the positive and negative token pairs.

Given per-pixel 3D points obtained by unprojecting the depth maps, we first sample valid pixels in the query frame and assign each sampled pixel to a query patch. Using the known camera intrinsics and extrinsics, we then project the corresponding 3D points into all other frames and retain only those projections that (i) fall inside the image and (ii) are depth-consistent with the target-frame depth maps (within a small relative tolerance of $1\%$ and excluding a narrow image boundary of 4 pixels).

For each target frame, we count how many 3D points from each query patch land in each target patch. This yields, for every query patch, a soft correspondence map over target patches in the form of projection overlap ratios. We select query patches that have sufficient valid projections across views and randomly sample up to a fixed number of them, with sampling probabilities proportional to their total number of correspondences. For each selected query patch, all target patches with $>10\%$ overlap form positive token pairs.

Because some sequences are dynamic, failing the positive-pair criteria does not automatically define a valid negative. Instead, we construct negative pairs by randomly sampling patches as query and target candidates, and then enforcing two constraints: (i) a geometric constraint, requiring the candidate target patch center to lie sufficiently far from the epipolar line induced by the query patch (*i.e.*, to have a large epipolar/Sampson distance), and (ii) an appearance constraint, requiring a sufficiently large $\ell_2$ distance between the mean RGB values of the two patches. Patches that satisfy both constraints are treated as negatives. We then subsample these negatives to obtain a balanced set of positive and negative patch pairs for supervising the matching loss on the last-layer tokens.

### A.3. VLM Prompt

We use a VLM to discard videos not suitable for multi-view geometry. We prompt the VLM as follows:

```
"You are an expert in computer vision,
photogrammetry, and Multi-View Geometry (MVG)."
"Analyze the video clip to determine if it is
suitable for high-quality 3D reconstruction."
"You must categorize the video and extract scene
metadata based on the strict criteria below."

STEP 1:  CHECK FOR HARD REJECTION FACTORS
``Assess the following.  If ANY are present, the
Classification is 'REJECT_HARD''':
1. Discontinuities & Editing:  Does the video
   contain cuts, dissolves, wipes, fades, or
   montage editing?  (Must be a single continuous
   shot).
2. Non-Physical/2D Content:  Is this a screen
```

```
   recording, a slideshow of static images,
   animation, or a cartoon?  (Must be real-world
   footage).
3. Extreme Visual Failure:  Severe motion blur
   (edges lost), severe rolling shutter
   (wobble/jello effect), or
   corruption/glitching.
4. Major Obstructions:  Are there heavy overlays
   (large text/UI), watermarks, or physical
   obstructions like a finger covering the lens?
5. Non-Pinhole Projections:  Is the footage 360°
   equirectangular or heavily distorted fisheye
   without calibration?

STEP 2:  CHECK FOR GEOMETRIC & TEXTURAL
SUITABILITY
"If the video passes Step 1, assess for
reconstruction quality.  If ANY are present, the
Classification is 'REJECT_SOFT':"
1. Insufficient Parallax:  Is the camera
   stationary?  Does it only rotate (pan/tilt) or
   zoom without physical translation?
   (Translation is required for depth).
2. Texture Issues:  Does the scene lack
   non-repetitive texture?  (e.g., blank white
   walls, clear blue sky only, dark shadows, or
   highly repetitive patterns like grid tiles).
3. Specularities & Transparencies:  Are the
   dominant features reflective (mirrors, water,
   glass) or transparent?  (These create 'ghost'
   geometry).
4. Focus & DOF Issues:  Is there severe focus
   hunting (pulsing), or is the depth-of-field so
   shallow that the background is entirely
   blurred (bokeh)?
5. Dynamic Dominance:  Is the frame dominated
   (>90%) by a moving subject (e.g., a close-up
   talking head) while the static background is
   visible but minimal?

STEP 3:  EXTRACT METADATA
"Determine the scene dynamics:"
1. Dynamic:  Moving objects are present (people
   walking, cars moving, wind blowing trees
   heavily).
2. Static:  The scene is rigid; only the camera
   is moving.

OUTPUT INSTRUCTIONS:
"Return a JSON object strictly following this
schema.  Do not output markdown formatting or
explanations outside the JSON."

{
"classification":
"ACCEPT" | "REJECT_HARD" | "REJECT_SOFT",
"reason": "Brief descriptions of the primary
flaw or 'Good candidate'",
"scene_dynamics": "Static" | "Dynamic"
}

"Label Definitions:"
REJECT_HARD: Unusable due to editing, synthetic
content, severe artifacts, or obstructions.
REJECT_SOFT: Technically usable but risks failure
due to rotation-only, reflections, lack of
texture, or focus issues.
ACCEPT: High-quality candidate.  Single shot,
clear translation/parallax, good texture, rigid
geometry.
```

## A.4. Annotation Filtering Criteria

To train the ensemble classifier (XGBoost, Random Forest, and CatBoost) described in Sec. 3.5.2, we extract several geometric features from each reconstruction. These features are designed to detect common failure modes such as collinear motion degeneracies, inconsistent scale, and noisy camera trajectories. Below, we provide definitions for the most important features implemented in our pipeline.

**Trajectory Smoothness.** We quantify the smoothness of the estimated camera trajectory using the acceleration of the translation vectors $\mathbf{t}$. The smoothness score $S_{\text{trans}}$ is calculated as the mean squared magnitude of the acceleration:

$$S_{\text{trans}} = \frac{1}{N-2} \sum_{i=1}^{N-2} \|\mathbf{t}_{i+1} - 2\mathbf{t}_i + \mathbf{t}_{i-1}\|^2 \qquad (2)$$

We compute a similar metric $S_{\text{rot}}$ for rotations using the second order difference of the rotation vectors. High values in $S_{\text{trans}}$ or $S_{\text{rot}}$ indicate jittery trajectories, often associated with poor SfM convergence.

**Parallax Angle Analysis.** To ensure sufficient baseline for multi-view stereo, we analyze the parallax angles of the reconstructed sparse point cloud. For a subset of sparse points $\mathcal{P}$, we identify the set of cameras $\mathcal{C}_p$ visible to point $p \in \mathcal{P}$. We compute the maximum angle subtended by any pair of cameras $c_j, c_k \in \mathcal{C}_p$ at point $p$. The final feature is the median of these maximum angles across all sampled points. Low median parallax indicates degenerate rotation-only motion or extreme distance.

**Point Cloud PCA Shape (Linearity & Planarity).** To detect geometric degeneracies such as "fly-by" straight-line reconstructions (which result in cylindrical ambiguity), we perform Principal Component Analysis (PCA) on the normalized point cloud coordinates. Let $v_1 \geq v_2 \geq v_3$ be the eigenvalues of the point cloud covariance matrix. We define *Linearity* as $(v_1 - v_2)/v_1$, *Planarity* as $(v_2 - v_3)/v_1$, and *Scattering* as $v_3/v_1$. Reconstructions with excessively high linearity are flagged as linear degeneracies and are typically discarded by the classifier.

**Depth Map Completeness.** This metric evaluates the density of the dense reconstruction stage. For each frame, we calculate the percentage of pixels containing valid, finite depth values relative to the total image resolution. The final feature is the average completeness across all frames. Extremely low values typically indicate failure in the Multi-View Stereo stage, often caused by textureless surfaces, high specularities, or bad cameras.

**Point Cloud Noise Level.** We estimate the signal-to-noise ratio using statistical outlier detection. For every point,

we compute the average distance to its nearest neighbors. Points are classified as noise if this distance exceeds the global average by more than two standard deviations. The feature is the percentage of points classified as noise, which allows us to detect reconstructions plagued by floating artifacts and outliers.

Note that this pipeline can provide depth values only for rigid pixels, so the metrics above assume that dynamic pixels are already masked out. We assume the model can learn to estimate depth for dynamic pixels from synthetic data.

### A.5. Language Alignment

In Sec. 4.4, we use a VLM to produce a sequence-level language embedding for each training sequence. Here, we provide the exact VLM prompt and representative generated descriptions. The hidden states of the generated text tokens are mean-pooled and projected to form the language embedding. The prompt and image tokens are excluded from this pooling. We use at most 128 newly generated tokens and keep the VLM fixed. The VGGT-Ω side is optimized during alignment fine-tuning, as described in Sec. 4.4. The images are provided as an unordered collection, both to the VLM and to our model.

**Language Alignment Prompt**

```
These images show the same scene from multiple
viewpoints.
Describe the full scene as one coherent scene,
not as separate images.

Use the following format:
Scene: <scene category or environment>.
Content: <main objects and coarse spatial
arrangement>.
Appearance: <stable visual attributes such as
material, color, or style>.

Write one short sentence for each field.
Mention information that is consistent across
multiple views, even if it is not visible in
every view.
Focus on shared scene content, major objects,
coarse layout, and stable appearance.
Mention dynamic objects only if they are
prominent and identifiable in multiple views.
Do not describe motion, actions, or
frame-specific changes.
Do not describe each image separately.
Do not mention camera motion, viewpoint order,
image quality, blur, exposure, lighting
artifacts, or uncertainty.
Keep the description concise, factual, and
semantically dense.
```

**Representative Descriptions**

```
Scene: A dimly lit room with a red hue,
featuring a door, a window, and a wall with a
patterned texture.
Content: The room contains a closed door on the
left, a window on the right, and a wall with a
textured pattern.
Appearance: The walls have a floral or damask
pattern, the floor is dark, and the overall
lighting is red.

- - - - - - - - - - - - - - - - - - - - - - - -

Scene: Snowy forest landscape with a winding
road.
Content: A snow-covered forest with tall pine
trees, a winding road, and a steep slope.
Appearance: The scene has a high-contrast look,
with dark clouds above and bright white snow
covering the ground and trees.

- - - - - - - - - - - - - - - - - - - - - - - -

Scene: Outdoor public area.
Content: A dark green plastic picnic table with
a hexagonal top and black legs, accompanied by
matching benches, situated on a paved surface.
Appearance: The table and benches are made of
uniform dark green plastic with a matte finish,
and the ground is composed of light gray square
tiles.

- - - - - - - - - - - - - - - - - - - - - - - -

Scene: Urban plaza with modern architecture.
Content: A central fountain with a large
circular sculpture, surrounded by paved walkways
and buildings.
Appearance: The fountain has a metallic,
silver-colored structure with a dark base, and
the surrounding area features light gray stone
tiles and modern buildings with glass and
concrete facades.
```

## B. Data Quality

One of the most important ingredients for training large 'foundation' models is data. Having access to a large amount of data is, however, insufficient; in fact, the *quality* of the data also has a strong effect on the model's behavior. For 3D reconstruction, we found that noisy data introduces particular failure modes in both self-supervised and supervised training.

For self-supervised training, we observe that unreasonably difficult videos, such as the ones containing discontinuous shot transitions as commonly seen in television footage, induce sharp loss spikes and may cause gradients to explode, resulting in a degradation of the model's performance. We therefore restrict self-supervised training to videos that pass the VLM pre-filtering check of Sec. 3.5.2, which reliably removes such cases.

In the supervised stage, the effects of noisy data are more subtle but equally important. We find that different types of errors in the annotations translate into specific failure modes at inference time. What is worse, these failure modes do not affect most of the images in standard benchmarks and may thus not be detected in the quantitative results. Instead, they emerge only when the model is tested on a new sample that resembles an incorrectly labeled sample seen during training. This behavior indicates that, while the model does learn

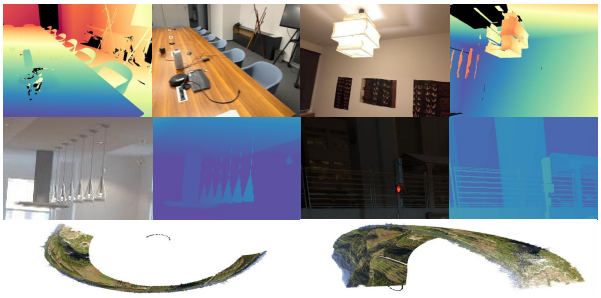

Figure 10. **Common Data Issues.** Top: examples from ScanNet++. Middle: examples from synthetic datasets such as Hypersim. Bottom: the doming effect.

the general principles of 3D reconstruction, it can nevertheless memorize idiosyncratic noise as well. Next, we discuss the most representative cases we have encountered.

**Sensors.** Some datasets annotate real-world scenes using sensors such as LiDAR or depth cameras. A common failure mode here is foreground-background leakage. For example, as shown in the top row of Fig. 10 with an example from ScanNet++, the depth of the chair's back corresponds to the background floor or wall, likely due to misalignment or holes in the sensor capture. We also observe spurious depth artifacts around the hanging light fixture, where the rendered depth contains fragmented foreground structures that do not correspond to the visible lamp geometry. Such artifacts are likely caused by unreliable sensor capture or reconstruction around over-exposed and partially translucent objects. We therefore exclude such datasets in the later phase of training.

**Thin Structure.** Synthetic data usually provides more accurate geometric annotations, but thin structures can be inaccurate in some synthetic datasets too. Objects such as fence bars often occupy only a few pixels and lie on sharp depth discontinuities. Although these structures are clearly visible in the RGB image, their rendered depth can be incomplete, overly smoothed, or misaligned. As a result, the label may assign a thin structure to the depth of the wall, window, floor, or distant background behind it, rather than to the actual object surface, as shown in the middle row of Fig. 10. At inference time, the model may ignore thin objects, recover only their thicker or lower parts, or produce depth maps in which thin structures are washed out by nearby large surfaces. This is a common problem in existing models, and we hypothesize that it can be mitigated with higher-resolution inputs or better synthetic data.

**Fake Background.** Another common issue in synthetic data is fake background. For example, in Kubric, PointOdyssey, BEDLAM, the background depth may correspond to proxy dome or floor geometry used for HDRI rendering, rather than the true 3D structure suggested by the appearance of the background. Although this is reasonable for rendering, it provides depth values that are inconsistent with the scene semantics. For datasets with a clear boundary between the real foreground and the synthetic background, such as BEDLAM, we filter out the background during training by thresholding the maximum valid foreground depth. We exclude the remaining from training.

**Doming Effect.** Methods using bundle adjustment, such as COLMAP [140], MegaSaM [99], and ViPE [65], can be used to generate pseudo ground truth. However, care is needed because they may produce degenerate solutions such as the doming effect, as shown in the bottom row of Fig. 10. This usually arises from weak global constraints in the image collection, *e.g.*, near-parallel viewing directions, insufficient loop closure, small triangulation angles, or inaccurate camera self-calibration, especially radial-distortion errors, which are common in Internet videos. Under these conditions, bundle adjustment can still achieve a low reprojection error while bending the recovered cameras and scene geometry into a globally inconsistent curved shape. Such reconstructions may appear locally plausible but provide incorrect large-scale geometry, making them harmful as supervision. They can be filtered out by supervised geometric filtering, as discussed in Sec. 3.5, or by comparing the annotated depths with predictions from existing monocular depth models or feed-forward reconstruction models.

**Humans in Walls.** We observe a recurring failure case in near-static street-view videos with pedestrians moving through the scene: almost all existing feed-forward reconstruction models often absorb humans into the static background, estimating them as part of nearby walls or buildings. This failure appears to be related to ambiguous boundary pixels in existing training datasets. For example, the most widely used multiview dataset Megadepth was annotated with COLMAP on phototourism images that often contain people. At human boundaries, the patch match stereo algorithm may assign some pixels to the surrounding static architecture, causing supervision to treat parts of people as background. Excluding MegaDepth or re-estimating its depths can substantially reduce this artifact.

**Data Ambiguities.** Inconsistencies in annotations across different datasets can lead to confusing model behaviors. For instance, in synthetic datasets such as Aria Synthetic Environments and HyperSim, outdoor scenes are often rendered as 2D textures applied to windows, meaning the GT depth corresponds to the window surface itself. Conversely, in other datasets, depth values correctly represent the actual physical objects visible through windows. This inherent GT mismatch across the training corpus confuses the model, which occasionally leads to strange predictive behaviors.

Several recent datasets have introduced alternative pipelines for annotating Internet videos, including SceneScribe-1M [190], PointWorld [70], Sekai [98], SpatialVID [182], and OmniWorld [233], which may also be worth exploring.

## C. Limitations

**Failure Cases.** Despite the generally strong performance of VGGT-Ω, we observe specific scenarios in which the model struggles. For example, the model's performance drops significantly in the presence of strong motion blur. Meanwhile, reconstruction quality often degrades if the field of view changes abruptly (e.g., shifting from $10^\circ$ to $160^\circ$ in a few seconds) or the camera is highly distorted. Additionally, because the model was exposed to some noisy data (*e.g.*, ScanNet++) during the early stage of training, its predictions are sometimes unstable in the cases like office scenes with many monitors. These limitations are primarily attributable to the distribution of our training data, and we hope to alleviate them in future work by incorporating more diverse and challenging sequences.

**Masked Sensitive Content.** Due to privacy and licensing constraints, some portions of the training data, such as human faces and trademarks, are masked or blurred. As a result, the model may rarely produce unexpected artifacts or unstable predictions in these regions. For example, we observed that the predicted depth for a person wearing black clothing can occasionally become less smooth. These artifacts can lead to visually distorted or qualitatively unappealing outputs.