# RayDer: Scalable Self-Supervised Novel View Synthesis from Real-World Video

Ulrich Prestel*,1,2, Stefan Andreas Baumann*,1,2,
Nick Stracke[1,2], and Björn Ommer[1,2]

[1] CompVis @ LMU Munich, Germany
[2] Munich Center for Machine Learning (MCML)

**Abstract.** Self-supervised novel view synthesis (NVS) remains challenging to scale, despite the abundance of video data, largely due to the brittleness of training on realistic videos and the hard-to-predict scaling behavior of multi-network system designs. We introduce RayDer, a unified, feed-forward transformer that consolidates camera estimation, scene reconstruction, and rendering into a single backbone, turning self-supervised NVS into a well-posed single-model scaling problem. A minimal dynamic state, treated as a nuisance factor, absorbs time-varying content and enables stable training on unconstrained real-world video. Importantly, RayDer keeps *static*-scene NVS as its target task: dynamic content is leveraged purely as scalable supervision, not reconstructed as in dynamic-scene (4D) NVS. Across multiple model sizes and orders of magnitude in data, RayDer exhibits clean power-law scaling with data and compute, and outperforms static-scene data mixtures. On a large number of benchmarks, RayDer achieves strong zero-shot open-set performance competitive with state-of-the-art supervised approaches.
**Project Page:** `https://compvis.github.io/rayder`.

**Keywords:** Novel view synthesis · Self-supervised learning · Video

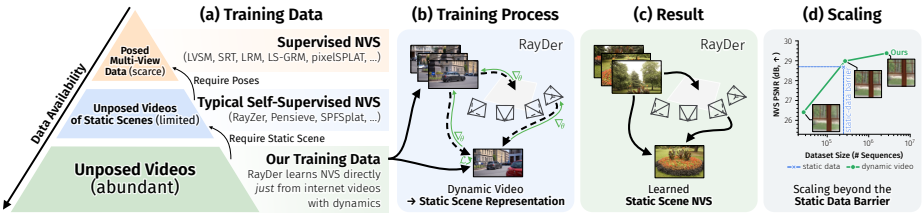



**Fig. 1: Training Static-scene Novel View Synthesis from Abundant Video.** Existing approaches rely on scarce data sources: supervised NVS requires posed multi-view images, while prior self-supervised methods require unposed videos/image collections of static scenes. Our method instead trains from generic unposed videos that may contain dynamic objects, enabling learning from the dominant form of visual data. This removes the static-scene data bottleneck and unlocks improved scaling with dataset size.

* Equal contribution.

## 1 Introduction

Novel view synthesis (NVS) should, in principle, be a highly scalable learning problem: given a set of posed views of a (static) scene, learn to predict other views. However, posed multi-view data is extremely scarce. Self-supervised NVS removes this pose requirement by learning camera geometry jointly with view synthesis, with recent methods even rivaling pose-supervised ones under real-world conditions [28]. Here, view synthesis is itself the *target* task, not a pretext task for learning transferable features: camera-pose labels are noisy and expensive to obtain on real video at scale, so removing them is precisely what makes abundant, unlabeled video usable for training the synthesis model itself. Despite this, and large amounts of video being available on the internet [1], these methods rely on restrictive assumptions – most notably, static scenes from curated datasets – that prevent reliable training at scale. We argue that the main obstacle to scalable self-supervised NVS is not data availability, but rather how current systems are designed to use that data.

Most existing approaches to self-supervised NVS [24, 25, 28, 46, 68, 73] are built as multi-network pipelines, with separate components for camera estimation, scene representation, and rendering. While effective at small scales, such designs make scaling difficult in practice: capacity has to be allocated across multiple interacting networks whose behavior is hard to predict and prohibitively expensive to sweep as models grow. As a result, even when more data or compute is available, scaling remains brittle and inconsistent.

A related challenge is robustness to the videos that are available at scale: unconstrained videos often contain dynamic scene content, which exposes instabilities in existing methods and prevents direct training on such scalable data sources. Importantly, our goal is *not* dynamic-scene (4D) NVS, but *static*-scene NVS learned from dynamic video: dynamic content is never reconstructed, only absorbed as a nuisance factor during training. Stable learning under these conditions is the prerequisite for accessing the data regime where scaling can be meaningfully studied.

We introduce **RayDer**, a unified, feed-forward transformer that enables scalable self-supervised novel view synthesis. Building on the RayZer [28] lineage, we develop a method that unifies camera estimation, scene reconstruction, and rendering in a single backbone, to enable scaling of self-supervised NVS. This simplification is not merely architectural: at fixed parameter counts, unification improves both pose estimation and novel view synthesis quality significantly, and enables straightforward, predictable scaling.

To ensure stable training on general video, RayDer uses a minimal explicit dynamic state variable that absorbs time-varying scene content during training. This variable is treated as a nuisance factor rather than a semantic representation and is not used at inference time – its role is solely to prevent scene dynamics information from corrupting camera pose representations, enabling stable training on unconstrained videos without changing the static-scene NVS task.

Across three orders of magnitude of training data and four model sizes, RayDer exhibits clean scaling in *both* data and compute simultaneously. Over the explored

regimes, compute-optimal performance is well described by a *single simple power-law fit*. The insight here is *not* the unsurprising observation that "more data helps". Rather, it is that self-supervised NVS from unconstrained video *becomes* a well-behaved scaling problem once two obstacles are removed: the brittle optimization of multi-network pipelines, and the corruption of camera representations by dynamic content. Once training is unified and dynamics are treated as a nuisance factor, self-supervised NVS scales like a standard single-model learning problem – cleanly and predictably in data, model size, and compute. We further show that aggregating existing static-scene datasets does not reproduce our scaled model's performance: the gains arise from *both* increased data availability and a system design that allows scaling to manifest.

Our main contributions are as follows:

- A unified single-network architecture for self-supervised NVS, replacing multi-network pipelines and enabling predictable scaling in model size and compute.
- A training formulation that remains stable on unconstrained video of dynamic scenes, treating scene dynamics as a nuisance factor to enable training on large-scale data.
- An empirical analysis of scaling behavior across data, model size, and compute, including compute-optimal scaling trends.

## 2 Related Work

**Feed-Forward Novel View Synthesis.** Per-scene optimization via NeRFs [45] or 3D Gaussian Splatting [33] produces high-quality views but requires dense pose captures and per-scene fitting at test time. Feed-forward models amortize this cost by predicting radiance fields [22], Gaussian primitives [4, 6, 7, 82, 86], or latent renderings [11, 29, 41, 47, 53–55, 55, 77–79, 84, 91] from posed images, but remain dependent on external pose pipelines [58, 70] at training and often at test time. Some methods reduce this dependency by leveraging flow, correspondence, or depth cues [5, 14, 21, 62], but retain partial geometric supervision. Generative approaches for camera-controlled synthesis [11, 41, 77–79, 84, 87, 91] often adapt pretrained (video) diffusion models for camera-controllable synthesis [41, 84, 91] and can hallucinate content beyond observed regions, but still require posed data for NVS training and take cameras as input rather than learning them. RayDer removes pose supervision entirely and learns camera representations jointly with view synthesis from raw video.

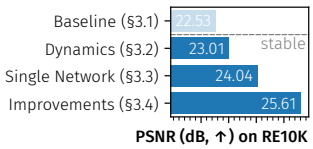
**Self-supervised and Pose-Free NVS.** Foundational work toward removing pose dependence in NVS includes UpSRT [57], which encodes unposed images into a latent scene representation, decoded using query rays that specify a relative target pose, and the Video Autoencoder [36], which learns to disentangle 3D structure and camera pose from video without pose supervision. RUST [56] removes train-time pose dependency by learning a latent pose code, but requires partial target views, limiting transferability [46]; DyST [59] handles dynamics but needs multi-view, multi-dynamics data that is difficult to obtain at scale. A second line learns explicit camera representations from monocular video. RayZer [28]

trains three separate ViTs end-ot-end on unposed *static*-scene video with only a photometric loss; Pensieve [73] adds Gaussian splatting and depth losses; Less3Depend [68] extends this to sparse images; E-RayZer [89] repurposes the same formulation for self-supervised 3D pretraining. These methods yield strong results, but are restricted to curated static-scene data whose combined scale [40, 42, 52, 92] is orders of magnitude smaller than available web video (Sec. 3.2), and employ multi-network pipelines that complicate scaling [16]. XFactor [46] further shows that many such systems learn pose shortcuts that fail to transfer across scenes. Pose-free Gaussian splatting methods [24, 25, 30, 39, 83] address a complementary setting – sparse unposed images –, but several bootstrap from geometric backbones [37, 74] pretrained with 3D supervision. RayDer addresses all three recurring limitations: static-data ceilings, multi-network complexity, and reliance on pretrained geometric priors.

**Large 3D Vision Models.** A growing line of "large 3D vision models" learns general 3D understanding by unifying multiple geometry tasks in a single transformer backbone, trading hand-engineered pipelines for scale and data breadth. Systems like VGGT [70] build upon the success of VGG-SfM [71] but remove most inductive biases, making everything into a single transformer backbone trained for multiple tasks simultaneously. They achieve (near-)state-of-the-art results across many tasks by just training on multiple supervised tasks across a large number of datasets. This has been further improved by works such as MapAnything [32], which further extended the set of tasks, $\pi^3$ [75], which removed the dependency on a specific canonical view, and others [9, 12, 13, 60]. Approaches like CUT3R [72] pursue similar problems from different perspectives, enabling incremental updates and improved efficiency. RayDer takes inspiration from this direction: just as replacing the individual components of VGG-SfM [71] with a single transformer in VGGT [70] led to improved scalability and thus improved performance, we aim to make self-supervised NVS scalable using, among other aspects, unification into a single transformer backbone.

## 3 Scaling Self-Supervised Novel View Synthesis

Our goal is to make self-supervised novel view synthesis (NVS) scalable in data, model size, and compute, without introducing task-specific supervision or brittle system design. Starting from a modern feed-forward baseline (§3.1), we identify three bottlenecks that prevent scaling:




- §3.2 Data: existing methods assume static scenes for training, severely limiting training data.
- §3.3 System: multi-network pipelines complicate scaling and optimization.
- §3.4 Quality: pose shortcuts and coarse patches limit reconstruction quality.

We address these with a sequence of targeted modifications, each validated by controlled ablations (see Tab. 1).

**Table 1: Ablation Summary.** We progressively address instability (§3.2), architectural scalability (§3.3), and synthesis quality (§3.4). NVS is zero-shot on RE10K [92], camera estimation via transferability [46] on DL3DV-10k [40]. Full table in Supp. Tab. A.1.

| Configuration | Stable Training | *Trained on SA-V [51]* NVS PSNR↑ w/o STATE | w/ STATE | Camera Est. R@10°↑ | T@0.1↑ | *Trained on SV-HQ [69]* NVS PSNR↑ w/o STATE | w/ STATE | Camera Est. R@10°↑ | T@0.1↑ |
|---|---|---|---|---|---|---|---|---|---|
| *§3.2: Stable Training on Dynamic Video* | | | | | | | | | |
| **A** RayZer-like [28] Baseline | ∼ | 22.53* | – | 59.8* | 6.5* | 22.69* | – | 66.0* | 7.7* |
| **B** + Dynamic State Prediction | ✓ | 13.42† | 24.01 | 56.1 | 7.0 | 13.48† | 24.67 | 54.4 | 6.0 |
| **C** + State Dropout | ✓ | 23.01 | 23.76 | 62.4 | 8.1 | 23.02 | 24.10 | 69.2 | 8.2 |
| *§3.3: Scalability through Consolidation* | | | | | | | | | |
| **D** + Single-network Consolidation | ✓ | 24.93 | 25.33 | 68.8 | 16.3 | 26.98 | 27.49 | 74.1 | 19.7 |
| **E** + Parallel-target Attention | ✓ | 24.04 | 25.12 | 70.1 | 15.6 | 25.91 | 26.21 | 70.9 | 18.2 |
| *§3.4: Improving Synthesis Quality* | | | | | | | | | |
| **F** + Autoregression over Views (ordered) | ✓ | 23.08‡ | 24.49‡ | 73.6 | 24.9 | 23.53‡ | 25.78‡ | 76.5 | 25.2 |
| **G** + Random-order Autoregression | ✓ | 25.45 | 26.28 | 84.4 | 37.2 | 27.27 | 29.57 | 86.0 | 39.1 |
| **H** + Local High-resolution Layers | ✓ | 25.61 | 26.87 | 85.0 | 40.2 | 27.78 | 30.23 | 88.7 | 42.4 |

*Results for **A** are from selected runs that did not diverge. †Dynamic state modeling without dropout creates inference-time dependency on state. ‡Ordered AR does not generalize to standard NVS test settings.

### 3.1 Preliminaries and Baseline

We start our exploration with RayZer [28], a feed-forward NVS method trained in a self-supervised manner on unposed, uncalibrated videos of *static* scenes with camera motion. Extending upon LVSM [29], RayZer consists of three distinct ViT [10] subnetworks (Fig. 2): a camera estimator $\mathcal{E}_{\text{cam}} : \{\mathbf{I}_i\} \mapsto \{\mathbf{p}_i\}$ maps views $\{\mathbf{I}_i\}$ to poses $\{\mathbf{p}_i\} \in SE(3)$ (and camera intrinsics). Then, the scene reconstructor $\mathcal{E}_{\text{scene}} : \{(\mathbf{I}_i, \mathbf{p}_i)\} \mapsto \mathbf{z}$ predicts a latent scene representation $\mathbf{z}$ from input views $\mathcal{I}_{\text{input}} = \{(\mathbf{I}_i, \mathbf{p}_i)\}$. Finally, a rendering decoder $\mathcal{D}_{\text{render}} : \mathbf{z}, \mathbf{p}_{\text{target}} \mapsto \hat{\mathbf{I}}_{\text{target}}$ predicts target views. All three networks are trained jointly end-to-end to optimize image-space reconstruction on target views $\mathcal{I}_{\text{target}}$ held out for the Scene Reconstructor, using poses jointly predicted for all views. Poses are encoded as Plücker rays [50].

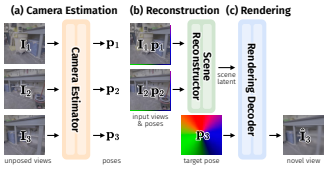



**Fig. 2: Preliminaries: RayZer [28].** RayZer uses three models responsible for different tasks: **a**) Camera Estimation, **b**) Reconstruction, **c**) Rendering.

**Baseline (Config A).** We use a scale-reduced RayZer-like model (∼140M params) for our baseline (Config **A**) and train on two complementary datasets: i) Segment Anything-Video [51], a diverse open-world video dataset with significant scene dynamics, and ii) SpatialVid-HQ [69], a curated, partially dynamic-scene dataset. Evaluation is *zero-shot* – models are evaluated on unseen benchmarks. We measure NVS quality on RealEstate-10K [92] in the pixelSplat [4] setting, and camera estimation via transferability [46] on DL3DV-10k [40].

### 3.2 Robust Learning from Dynamic Videos

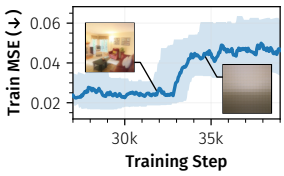



**Fig. 3:** Training RayZer directly on dynamic videos leads to instabilities and stalled training.

Scaling self-supervised NVS faces an immediate data bottleneck: truly static-scene videos, as required by current methods [28, 46, 68, 73], are a tiny subset of what is available at scale. However, training RayZer directly on dynamic video leads to gradient spikes and instabilities: the original RayZer [28] diverges consistently when trained on SpatialVid [69] or SA-V [51] (Fig. 3; see Sec. C.3 for a detailed analysis). We frame this as a *representation* problem rather than an optimization problem. In dynamic video, a target view $j$ is explained from an input view $i$ by two factors: the camera pose change $\mathbf{p}_i \to \mathbf{p}_j$ *and* the dynamic state change $\mathbf{s}_i \to \mathbf{s}_j$. Exposing only camera pose as conditioning forces the model to "hide" dynamic-state information inside the camera representation, causing representation drift and instabilities. Importantly, even when training on dynamic video, our target task remains *static*-scene NVS: the dynamic state is what lets us *learn* this static-scene task from dynamic data without modeling scene motion at inference, making training scalable rather than attempting dynamic-scene NVS.

**Dynamic State Prediction and Dropout (Config B, C).** We address this by predicting a per-view dynamic state embedding $\mathbf{s}_i$ alongside the camera pose:

$$\mathcal{E}_{\text{cam,state}} : \{\mathbf{I}_i\} \mapsto \{(\mathbf{p}_i, \mathbf{s}_i)\}, \quad \mathcal{D}_{\text{render,state}} : \mathbf{z}, (\mathbf{p}_{\text{target}}, \mathbf{s}_{\text{target}}) \mapsto \hat{\mathbf{I}}_{\text{target}}, \quad (1)$$

where $\mathbf{s}_i \in \mathbb{R}^{d_{\text{state}}}$ lets the model capture time-varying content without needing to interfere with the pose $\mathbf{p}_i$, and is provided to the renderer as an additional token. This intentionally minimal change – no motion fields, temporal losses, or disentanglement – eliminates training instabilities entirely: across all our experiments, not a single run that includes this change (Config **B**) has diverged. Since the target state $\mathbf{s}_{\text{target}}$ is unknown at inference, we randomly replace it with a zero vector during training [19], forcing the model to synthesize plausible views both with and without state conditioning (Config **C**). This retains the stability gains, while resolving the inference dependency on ground truth state and additionally improving camera estimation (Tab. 1, **A**→**C**), consistent with the dynamic state reducing pressure on the pose representation to encode dynamic information. We stress that the state is deliberately *not* a disentangled 4D representation: it is a nuisance variable whose only role is to prevent leakage into camera tokens. We probe what it captures in Sec. C.1, where transplanting states across views shows it primarily absorbs moving/time-varying scene content while the camera tokens carry pose; at inference on scenes with dynamic content, this manifests as the static scene being rendered from the correct novel pose while moving regions degrade to a temporal average (Sec. C.1, limitations in Sec. 4).

### 3.3 Scalability through Network Consolidation

With stable training on general video, the next bottleneck is architectural: scaling multi-network systems – which current self-supervised NVS methods [24, 25, 28,

46, 68, 73] rely on – is highly complex [16], as capacity must be distributed across interacting components whose scaling behavior is hard to predict.

**Single-Network Consolidation (Config D).** To reduce scaling decisions to a single network, which can allocate capacity between tasks as needed, and improve performance by sharing features, we unify all three components – camera/dynamic state estimation, scene reconstruction, and rendering (see Fig. 4) – in a single model $\mathcal{M}$. Besides scaling simplicity, this is motivated by the idea that pose estimation and view synthesis are not separate problems [70]: they can share features, and training signals can become cleaner when the clear separation between networks is removed. Our unified model $\mathcal{M}$ operates in two modes (where $\not{\cdot}$ denotes abscence of an input/output):

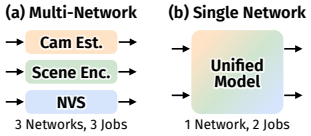



**Fig. 4: Consolidation.** We combine RayZer's three networks (**a**) into one (**b**).

$$\mathcal{M}: \begin{cases} \{(\mathbf{I}_i, \not{\mathbf{p}}, \not{\mathbf{s}})\} \mapsto \{(\mathbf{p}_i, \mathbf{s}_i)\} & \triangleright \text{ Cam. Est.} \\ \underbrace{\{(\mathbf{I}_i, \mathbf{p}_i, \not{\mathbf{s}})\}}_{\text{input views}} \cup \underbrace{(\not{\mathbf{I}}, \mathbf{p}_j, \mathbf{s}_j)}_{\text{target pose}} \mapsto \underbrace{\hat{\mathbf{I}}_j}_{\text{target view}} & \triangleright \text{ NVS} \end{cases} \tag{2}$$

All heavy computation lies in a single shared backbone, conditioned on token role via adaptive norms [26, 47]. In addition to significantly simplifying scaling decisions, empirically, this unification at fixed parameter count leads to significant gains in both NVS and camera estimation performance (Tab. 1, **C**→**D**).

**Parallel-target Attention (Config E).** Naively treating the consolidated model as decoder-only [29] reprocesses input views for each target view, which is prohibitively expensive. We factorize attention such that input tokens only attend to each other, while target tokens attend to themselves and input tokens (see Fig. 5). This enables KV caching during inference and parallel target prediction during training, reducing per-target compute by $\sim 7\times$ at a minor quality trade-off (Tab. 1, **D**→**E**).

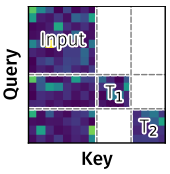



**Fig. 5:** Our attention mask.

### 3.4 Improving Synthesis Quality

The remaining issues are quality-related: pose representations can learn shortcuts in video, and large patch sizes sacrifice local details.

**Autoregressive Pose Learning (Config F, G).** When training on video frames, many input views make pose prediction easy to solve by using frame-order shortcuts rather than actual geometry (Fig. 6a). We find that in practice, this results in predicted poses primarily encoding time rather than the true viewpoint. In contrast, single- or few-view NVS requires the full pose to be encoded geometrically (Fig. 6b). We implement that by training autoregressively over views: given a subset of views, predict another, then condition on the expanded set. This forces the model to learn to predict poses that are useful for NVS in both sparse and dense settings. Extending upon our

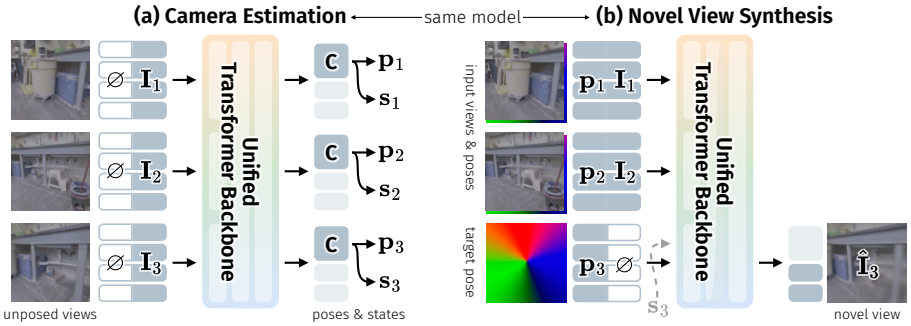



**Fig. 7: Final Architecture Overview.** RayDer unifies camera estimation (**a**) and novel view synthesis (**b**) in a single transformer backbone. Lightweight local intra-frame encoder and decoder layers handle high-resolution processing.

factorized attention pattern (Sec. 3.3), we make attention causal over input views and train next-view NVS for $|\mathcal{I}_{\text{input}}| = 1, 2, ..., |\mathcal{I}_{\text{total}}| - 1$ input views. Ordered autoregression (Config **F**) consequently improves camera estimation quality significantly (Tab. 1, **E**→**F**), but creates a train-test gap, since standard NVS settings do not condition on and generate frames in temporal order. Randomizing the autoregression order instead (Config **G**) closes this gap and further improves both camera estimation and NVS quality (Tab. 1, **E**→**G**).

**Local High-resolution Layers (Config H).** Large patch sizes hurt synthesis quality [29, 48, 66], but reducing the patch size $p$ scales cost as $\mathcal{O}(p^4)$, making it prohibitively expensive. Following Crowson et al. [8], we add shallow high-resolution local layers (using neighborhood attention [17]) around the main backbone. These layers operate intra-frame and provide extra high-frequency capacity at minimal cost (Tab. 1, **G**→**H**).

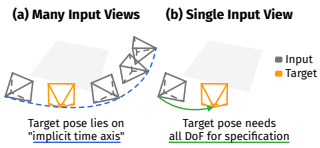



**Fig. 6:** Many input views (**a**) allow encoding camera poses via an implicit "time" axis; sparse views (**b**) require true relative camera poses.

### 3.5 Final Architecture

Config **H** is the final RayDer architecture, which we train at four different model scales (Tab. 2): a single transformer that i) predicts per-frame pose and state tokens $\{(\mathbf{p}_i, \mathbf{s}_i)\}$ from a set of images $\{\mathbf{I}_i\}$, ii) synthesizes target views via random-order autoregression with parallel-target attention, and iii) trains stably on general video. RayDer-S has the same scale (depth, width) as our ablation models (Config **A-H**), but is trained longer on more data; RayDer-B is scale-matched vs. RayZer [28]. An architecture overview is shown in Fig. 7.

## 4 Experiments

Our experiments address four major questions:

§4.1 Does RayDer exhibit predictable scaling behavior in data, model size, and compute?

§4.2 Do existing static-scene datasets support the same scaling regime, or is learning from general video essential?

§4.3,4.4 How does open-set self-supervised NVS compare to prior work with supervision & large-scale pretraining?

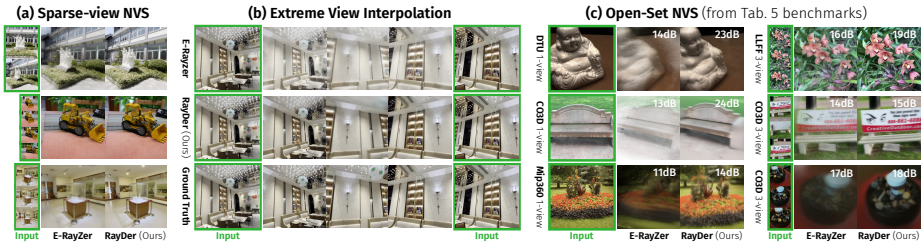



**Fig. 8:** *Zero-shot* qualitative samples of RayDer compared with E-RayZer [89] in **(a)** typical (non-dense view) NVS settings, **(b)** an extreme setting with ∼zero context view overlap, and **(c)** settings evaluated in Tab. 4. Our RayDer model, trained on large-scale non-static-constrained video data, outperforms E-RayZer – a prior model trained on a multi static dataset mixture – by a wide margin.

**Implementation Details.** We train all models on SpatialVid [69] (∼2.7M videos), extracting 8 views per clip with ∼0.5s spacing (randomly chosen per epoch), using AdamW [43] at batch size 256 and a resolution of $256^2$. We measure PSNR, LPIPS [88], and SSIM [76] on synthesized novel views for our evaluations, generally in *zero-shot* settings on unseen datasets and using $256^2$ resolution unless noted otherwise. For further details, see Supp. Secs. A and B.

**Train-Test Leakage.** To ensure that our gains with increased train data scale, stem from improved generalization rather than leakage, we also check for leakage between our train and test sets. Specifically, we check each test view from every dataset we evaluate on against the videos used for training our main models & models used for scaling evaluations (i.e., our copy of SpatialVid [69]), in two stages: first, we compute pHash and dHash perceptual hashes [3, 34] for all frames and flagged any train-test pair with Hamming distance $< 8$ bits as a candidate – yielding 16,381 candidate pairs. Second, to discard hash collisions on visually unrelated content, we keep only pairs with DINOv3-L [61] CLS cosine similarity $\geq 0.2$,

**Table 2: Model Scales.** We jointly scale depth, width, and head count.

| Model | Layers $N$ | Hidden Size $d$ | Heads | Parameters |
|---|---|---|---|---|
| RayDer-XS | 12 | 384 | 6 | 59M |
| RayDer-S | 18 | 512 | 8 | 145M |
| RayDer-B | 24 | 768 | 12 | 422M |
| RayDer-L | 24 | 1024 | 16 | 743M |

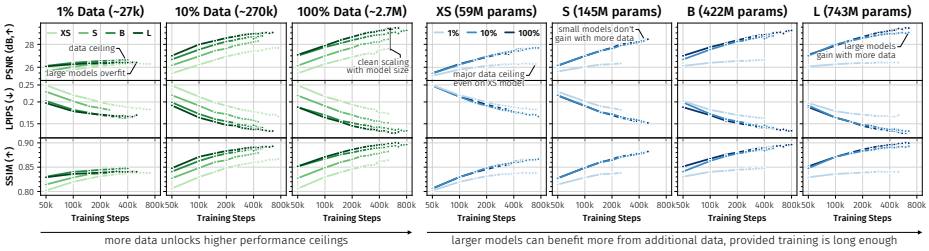



**Fig. 9: Scaling Across Data and Model Size.** We evaluate models trained on SpatialVid (2.7M total samples) at different model scales (visualized as shades of green) and dataset fractions (shades of blue), on RE-10k [92]. *Left:* Increasing data scale consistently improves performance, as long as model scale is not a limit. At small data scales, large models tend to overfit, resulting in *worse* performance than smaller ones. *Right:* Increasing model scale also consistently improves performance. However, insufficient dataset scale imposes an upper ceiling on achievable test performance.

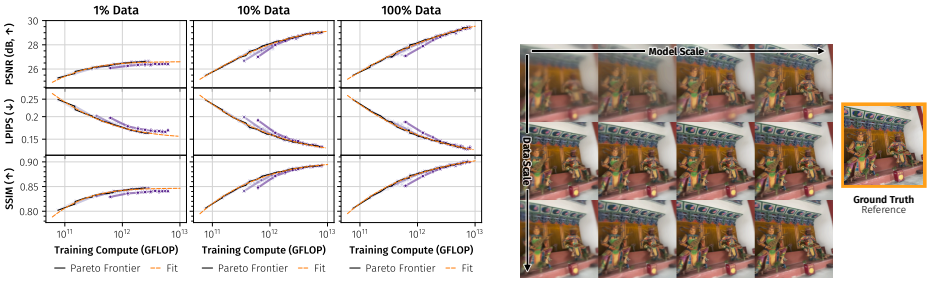
narrowing the set to 191 candidates. Manual inspection of all 191 remaining pairs revealed zero matches – therefore, our evaluations should be free of the influence of train-test leakage and instead represent genuine generalization.

### 4.1 Scaling Behavior across Data, Model Size, and Compute

Prior self-supervised NVS methods are fundamentally *data-limited*: trained on small, curated static-scene datasets that saturate quickly, they cannot meaningfully scale model capacity. By enabling stable training directly on dynamic video, RayDer allows scaling across multiple orders of magnitude. **Setup.** We train RayDer at four scales (**XS**/**S**/**B**/**L**; Tab. 2) on three dataset fractions of SpatialVid: **1%** (∼27k videos), **10%** (∼270k, matching the combined size of common static-scene NVS datasets [40, 42, 52, 92]), and **100%** (∼2.7M). Figure 9 shows that both data and model scaling consistently lead to improvements, provided neither is a bottleneck. Increasing data consistently improves performance when model capacity and training are sufficient, while small models saturate early. Large models overfit on small data, even underperforming against smaller models. These results highlight **scaling neither compute nor data alone is sufficient** – scaling requires *both*. All models at 1% data scale converge to a common ceiling, confirming data as the dominant bottleneck at small scale. Benefits continue beyond 10%, which matches the combined size of common static-scene NVS datasets, highlighting the need for more scalable data sources.

**Compute-optimal Scaling.** Building on LLM scaling analysis [18, 20, 31], we find that RayDer's compute-optimal Pareto frontier of NVS performance on unseen test sets across training compute $C$ (in GFLOP) and dataset size $D$ (in number of videos) can be modeled as [31]:

$$\underbrace{L(C,D)}_{\text{test metric}} = \underbrace{L_\infty}_{\text{irreducible part [18]}} + \Big(\underbrace{AC^{-\alpha}}_{\text{compute term}} + \underbrace{BD^{-\beta}}_{\text{data term}}\Big)^{\gamma}, \tag{3}$$




**(a) Compute-optimal Scaling.** RayDer's compute-optimal performance on unseen datasets (here, RE10K [92]) *across both compute and train dataset size* is well-approximated by a *single* power law.

**(b) Qualitative Scaling.** RayDer's qualitative behavior follows the trends seen in quantitative evals (Fig. 9): more data & compute jointly improve NVS quality.

**Fig. 10: Joint Data & Compute Scaling.**

with $L \in \{\text{MSE}, \text{LPIPS}, 1-\text{SSIM}\}$ (MSE corresponds to PSNR and is computed on image data scaled to $[-1, 1]$). The compute and data terms each capture the error that can only be removed by scaling that aspect. Fitting this power law to our compute-optimal Pareto frontier of models we trained across model and dataset scale (see Fig. 10a), we obtain an accurate ($R^2 > 0.99$) description of RayDer's behavior over both compute and data scale, confirming that its scaling is well-behaved. All metrics exhibit non-zero irreducible error terms $L_\infty > 0$, reflecting fundamental limits of the setting (e.g., occluded regions). Both compute and data terms contribute meaningfully, with the latter formalizing an important empirical insight: increasing compute yields diminishing returns unless sufficient training data is available, and scaling training data beyond curated static-scene datasets requires methods that can train directly on general video. Refitting the same power law on additional, harder zero-shot benchmarks (WildRGBD [81], CO3D [52]) remains similarly accurate (Sec. B.2), indicating the trend is not an artifact of fitting a single test set.

### 4.2 Static-Scene Data Does Not Enable the Same Scaling Regime

The scaling analysis in Sec. 4.1 establishes that data scale is a dominant factor for continued improvement. But can existing static-scene datasets, which offer domain-aligned training signals for standard NVS benchmarks, supply sufficient data to sustain this scaling regime? If so, the added complexity of training on dynamic video would be unnecessary. To test this, we train two additional RayDer-L models: a **static-only** model trained on a mixture of multiple large-scale static-scene NVS datasets (RE10K [92], DL3DV-10K [40], uCO3D [42]; ∼247k videos total; denoted as *static mix*), and a **mixed** model trained on both SpatialVid [69] (mixed with dynamic content, ∼2.7M videos) *plus* the static mix, with equal sampling between both during training.

Table 3 shows that the static-only model significantly underperforms despite using static-scene data closely aligned with the (static-scene) test setting. The static mix reflects the combined scale of commonly used static-scene NVS datasets

**Table 3: Training Data Comparison.** We train models at scale on three different dataset combinations: *Static Mix:* a combination of multiple public static-scene NVS datasets (RE10K [92], DL3DV-10K [40], uCO3D [42]; ∼247k videos total), *General:* SpatialVid [69] (∼2.7M videos total), and a combination of the two. Evaluation shows that combining even multiple public static-scene datasets underperforms substantially compared to training on general video. Combining both leads to minor additional gains.

| Model | Steps | Training Data | Scene Dynamics | RE10K NVS | | |
|---|---|---|---|---|---|---|
| | | | | PSNR↑ | LPIPS↓ | SSIM↑ |
| RayDer-L | 500k | Static Mix only (∼250k) | static | 28.68 | 0.158 | 0.888 |
| | | SpatialVid only (∼2.7M) | including dynamic | <u>29.38</u> | **0.135** | <u>0.899</u> |
| | | Static Mix + SpatialVid† | including dynamic | **29.42** | <u>0.136</u> | **0.901** |

†batch composition during training is equally distributed between static mix and SpatialVid.

and roughly matches the 10% data fraction in Sec. 4.1, where our scaling analysis already predicts that larger models cannot fully benefit due to limited data. Despite the domain alignment advantage, the scale deficit dominates: training on more (partially dynamic) video empirically outweighs a cleaner training distribution. Adding static data to the larger video dataset at the same training horizon yields only marginal gains, suggesting the static datasets are largely subsumed by the larger corpus. These results validate our core thesis: the bottleneck for self-supervised NVS scaling is not the quality of static-scene curation but the *quantity* of data, which requires moving beyond curated static-scene corpora.

### 4.3 Open-set Novel View Synthesis

Most prior self-supervised NVS methods [28, 46, 68] focus on closed-domain evaluation, training and testing on the same datasets. We train a single RayDer model on generic data and evaluate it zero-shot across a wide range of datasets (LLFF [44], DTU [27], CO3D [42], WildRGBD [81], Mip-NeRF 360 [2], and Tanks & Temples [35]), camera baselines, and numbers of input views, extending the extensive evaluation by Zhou et al. [91] in Table 4. This setting better reflects real-world deployments and avoids dataset-specific tuning. We note that, unlike the supervised baselines, RayDer uses its own predicted camera poses at inference, not the dataset's ground truth annotations, making this a strictly harder setting.

Despite being trained fully self-supervised, from scratch in a single stage, RayDer achieves **state-of-the-art or near-state-of-the-art performance across the majority of settings** at more than an order of magnitude less training compute. It is competitive with much larger models such as SEVA and Kaleido, which rely on large-scale video diffusion pretraining. RayDer achievse this while requiring neither pose supervision at train or test time nor pretrained foundation model weights – a substantially more constrained and scalable setup. This is unlike E-RayZer [90], which requires static-scene videos for training and, while combining a large number of static-scene datasets, is still significantly limited in the amount of training data it can use, limiting scaling ( Sec. 4.2).

**Table 4: Open-set Novel View Synthesis (PSNR↑).** We extend the evaluation by Zhou et al. [91] and compute PSNR across a large variety of settings (columns). Despite being trained fully self-supervised and without large-scale video diffusion pretraining, RayDer is (near-)state-of-the-art across the majority of datasets and evaluation settings.

| | | | *small-viewpoint* | | | | | | | | | | *large-viewpoint* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Dataset→ | LLFF | | DTU | | CO3D | | WRGBD | | M360 | T&T | CO3D | WRGBD | | M360 | | T&T | |
| | | Split→ | R | | R | | V | R | $S_e$ | $S_h$ | R | V | R | $S_h$ | | R | | S | |
| Model | Params | Self-sup. | $\lvert\mathcal{I}_{in}\rvert$→1 | 3 | 1 | 3 | 1 | 3 | 3 | 6 | 6 | 1 | 1 | 1 | 3 | 1 | 3 | 3 | 6 |
| MVSplat [6] | 12M | ✗ | 11.23 | 12.50 | 13.87 | 15.52 | 12.52 | 13.52 | 14.56 | 12.54 | 13.56 | 13.22 | – | – | – | – | – | – | – |
| DepthSplat [82] | 354M | ✗ | 12.07 | 12.62 | 14.15 | 16.24 | 13.23 | 13.77 | 15.93 | 14.23 | 14.01 | 14.35 | 10.42 | 9.35 | 13.53 | 10.49 | 12.54 | 9.78 | 10.12 |
| ViewCrafter† [84] | 1.4B | ✗ | 10.53 | 13.52 | 12.66 | 16.40 | 14.96 | 14.72 | 16.42 | 12.66 | 14.59 | 14.07 | 10.11 | 9.12 | 13.45 | 9.79 | 10.34 | 9.88 | 10.32 |
| SEVA† [91] | 1.3B | ✗ | 14.03 | 19.48 | 14.47 | **20.82** | 18.40 | **19.25** | 19.75 | **18.91** | 16.70 | 15.16 | 15.30 | 14.37 | **17.28** | 12.93 | 15.78 | 12.65 | 13.80 |
| Kaleido†‡ [41] | 3.1B | ✗ | 15.34 | 20.71 | – | – | – | – | – | – | **18.03** | – | – | – | – | 13.74 | **16.78** | 13.20 | **14.61** |
| E-RayZer* [89] | 246M | ✓ | 10.44 | 18.01 | 10.31 | 16.97 | 12.94 | 17.76 | 17.72 | 16.18 | 15.86 | 10.36 | 12.94 | 10.53 | 14.47 | 9.78 | 15.17 | 12.88 | 13.35 |
| RayDer-L-$576^2$ (Ours) | 743M | ✓ | **17.11** | **21.38** | **16.01** | 17.92 | **21.10** | 19.09 | **20.07** | 17.23 | 16.25 | **18.74** | **16.84** | **14.55** | 15.97 | **14.96** | 15.85 | **13.59** | 13.81 |

Split abbreviations: R: ReconFusion [80]; V: ViewCrafter [84]; $S_{\{e,h\}}$: SEVA [91], easy (e) and hard (h) variants.
Dataset references: LLFF [44], DTU [27], CO3D [42], WRGBD [81], M360 [2], T&T [35]
‡Kaleido evaluates at $512^2$ instead of $576^2$
†Diffusion-based models. *Multi-dataset Ckpt

On lab datasests such as DTU, just like E-RayZer [90], RayDer underperforms supervised methods. We find this is primarily due to unreliable pose estimation in regimes (perfectly clean backgrounds with no structure) not present in typical general video training data: RayDer is trained exclusively on unconstrained *real-world* video. We view this as an expected limitation of the training data distribution rather than a failure of the approach.

**Qualitative Results.** Figure 8 compares RayDer against E-RayZer [89] across three challenging regimes – sparse-view NVS, extreme wide-baseline interpolation, and some settings from Tab. 4 – where RayDer produces markedly sharper and more consistent novel views. Further samples are in Supp. Sec. E.

### 4.4 Closed-Set Static & Supervised Comparison

We compare to previous methods in closed-set settings on small-scale static datasets. In the dense 24-view DL3DV-10K [40] setting from RayZer [28] (Tab. 5), RayDer is competitive with the state-of-the-art, despite not being intended/optimized for this setting – our other experiments use one to three orders of magnitude more training data. This demonstrates that our adaptations for large-scale dynamic-scene training do not sacrifice small-scale static-scene capability.

**Can Supervision replace Self-Supervision when training on Dynamic Data?** An important question is whether supervised mthods could simply use off-the-shelf pose estimators to train on the same large-scale video data. We test this by training LVSM [29] on SpatialVid [69] using pseudo-ground truth camera poses from MegaSaM [38]. Our self-supervised RayDer outperforms the supervised LVSM by a wide margin (Tab. 6, +2.9dB PSNR), demonstrating that self-supervised pose learning can be substantially more effective than relying on pseudo-ground truth annotations in this data regime. This result is practically significant: obtaining pseudo-GT poses via MegaSaM for SpatialVid cost ∼69k GPU-h [69], more than an order of magnitude above the ∼1,2k GPU-h

**Table 5: Static-Dataset Comparison.** We extend the evaluation by Jiang et al. [28], training and evaluating on dense-view DL3DV. We train our model with the same settings (transformer size, view count, training steps) as the baselines. Despite the various adaptations to enable training on general video, our model is competitive also in this setting.

| Model | Training Data | | DL3DV (Even [28]) | | |
|---|---|---|---|---|---|
| | Dataset | GT Pose | PSNR↑ | LPIPS↓ | SSIM↑ |
| GS-LRM [86] | DL3DV [40] | ✓ | 23.49 | 0.252 | 0.712 |
| LVSM [29] | DL3DV [40] | ✓ | 23.69 | 0.242 | 0.723 |
| RayZer [28] | DL3DV [40] | ✗ | <u>24.36</u> | <u>0.209</u> | <u>0.757</u> |
| RayDer-B (Ours) | DL3DV [40] | ✗ | **24.51** | **0.142** | **0.758** |

**Table 6: Supervised Dynamic-Dataset Comparison.** Comparing our RayDer model with LVSM trained on dynamic videos with MegaSaM [38] camera poses at matched settings (transformer size, view count, training steps) results in a major performance gain, despite not using any pose supervision. We also note that obtaining these pseudo-GT camera poses costs an order of magnitude more compute than *training* either model.

| Model | Training Data | | RE10K NVS | | |
|---|---|---|---|---|---|
| | Dataset | GT Pose | PSNR↑ | LPIPS↓ | SSIM↑ |
| LVSM [29] | SpatialVid [69] | ✓ | 25.44 | 0.184 | 0.729 |
| RayDer-B (Ours) | SpatialVid [69] | ✗ | **28.35** | **0.151** | **0.879** |

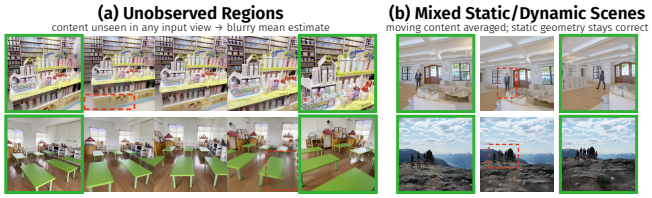



**Fig. 11: Limitations.** Both main failure modes arise from the regression objective collapsing under-constrained content to a low-frequency average, dashed boxes mark affected regions. **(a)** content unseen in any input view is rendered as a blurry mean estimate. **(b)** in presence of dynamic content, the static scene is rendered correctly from the novel pose; moving content is averaged.

required to *train* the RayDer-B model in this comparison (Supp. Tab. B.4), making the supervised path both less effective and less efficient.

### 4.5 Limitations

**Unobserved Regions.** Content not visible in any context view is rendered as a blurry, low-frequency "mean estimate" rather than plausible but hallucinated detail (Fig. 11a; see also Supp. Fig. C.4), a known consequence of the regression objective also shared by (E-)RayZer [28, 89], LVSM [29], etc. The model provides no explicit uncertainty signal there – a generative/uncertainty-aware decoder compatible with our unified backbone is a natural direction for future work.

**Mixed Static/Dynamic Scenes.** On real video containing both static structure and moving content, RayDer reconstructs the static geometry and renders it from the correct viewpoint, but dynamic content is not rendered faithfully, degrading to a mixture of blur and loose interpolation (Fig. 11b). This follows from treating the dynamic state as a nuisance factor (Sec. 3.2) rather than an explicit scene representation; we analyze the resulting entanglement of state

and pose in Sec. C.1. An explicit, disentangled treatment [59] would require multi-view videos of dynamic scenes, reintroducing exactly the dependence on scarce, curated data that our method is designed to avoid. Extending to full dynamic (4D) NVS while retaining the ability to train on abundant generic video is left to future work.

## 5 Conclusion

Self-supervised novel view synthesis (NVS) has long promised scalability through videos by not requiring ground-truth camera pose annotations, yet existing approaches remained constrained by hard-to-scale multi-network pipelines and restrictive static-scene assumptions. In this work, we introduced RayDer, a unified feed-forward transformer that consolidates camera estimation, scene reconstruction, and rendering into a single scalable backbone, enabling stable training on unconstrained real-world video while preserving the static-scene NVS objective. Through explicit dynamic state handling via a nuisance variable, architectural unification, and autoregressive pose learning, RayDer makes scaling of self-supervised NVS clean across data, model size, and compute. Empirically, this yields clean power-law scaling behavior, strong zero-shot open-set performance competitive with supervised and video diffusion-based systems, and transferable camera pose representations learned entirely without pose supervision.

Beyond the specific architecture, our results suggest a broader perspective: one major limitation of many prior self-supervised NVS methods was not the absence of supervision, but the inability to *use scalable data regimes*. By enabling stable learning from scratch on generic video and demonstrating clean scaling, RayDer positions self-supervised NVS within the same scaling-driven paradigm that has shaped progress in language and some vision foundation models.

Looking forward, RayDer opens up several directions for future work, including integration with partial supervision and generative modeling, extension toward 4D NVS, and continued scaling toward 3D world foundation models.

## Acknowledgments

This project has been supported by the Horizon Europe project ELLIOT (GA No. 101214398), the project “GeniusRobot” (01IS24083) funded by the Federal Ministry of Research, Technology and Space (BMFTR), the BMWE ZIM-project (No. KK5785001LO4) “conIDitional LoRA”, the German Federal Ministry for Economic Affairs and Energy within the project “NXT GEN AI METHODS - Generative Methoden für Perzeption, Prädiktion und Planung”, and the bidt project KLIMA-MEMES. The authors gratefully acknowledge the Gauss Center for Supercomputing for providing compute through the NIC on JUWELS/JUPITER at JSC and the HPC resources supplied by the NHR@FAU Erlangen. We thank Olga Grebenkova, Kosta Derpanis, and Tommaso Martorella for feedback, proofreading, and helpful discussions, and Owen Vincent for technical support.

## References


1. YouTube for Press — blog.youtube. `https://blog.youtube/press/`, [Accessed 09-11-2025]
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
3. Buchner, J.: ImageHash: A python perceptual image hashing module — github.com. `https://github.com/JohannesBuchner/imagehash` (2025)
4. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19457–19467 (2024)
5. Chen, Y., Lee, G.H.: Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24–34 (2023)
6. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European Conference on Computer Vision. pp. 370–386. Springer (2024)
7. Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.J., Cai, J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. Advances in Neural Information Processing Systems **37**, 107064–107086 (2024)
8. Crowson, K., Baumann, S.A., Birch, A., Abraham, T.M., Kaplan, D.Z., Shippole, E.: Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In: Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 9550–9575. PMLR (21–27 Jul 2024)
9. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. arXiv preprint arXiv:2507.16443 (2025)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
11. Elata, N., Kawar, B., Ostrovsky-Berman, Y., Farber, M., Sokolovsky, R.: Novel view synthesis with pixel-space diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26756–26766 (2025)
12. Fang, K., Zhou, C., Fu, Y., Li, H.H., Chen, Y.: IncVGGT: Incremental VGGT for memory-bounded long-range 3d reconstruction. In: The Fourteenth International Conference on Learning Representations (2026)
13. Feng, W., Qin, H., Wu, M., Yang, C., Li, Y., Li, X., An, Z., Huang, L., Zhang, Y., Magno, M., et al.: Quantized visual geometry grounded transformer. arXiv preprint arXiv:2509.21302 (2025)
14. Fu, Y., Liu, S., Kulkarni, A., Kautz, J., Efros, A.A., Wang, X.: Colmap-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20796–20805 (2024)
15. Gao, H., Li, R., Tulsiani, S., Russell, B., Kanazawa, A.: Monocular dynamic view synthesis: A reality check. In: NeurIPS (2022)

16. Ghorbani, B., Firat, O., Freitag, M., Bapna, A., Krikun, M., Garcia, X., Chelba, C., Cherry, C.: Scaling laws for neural machine translation. In: International Conference on Learning Representations (2022)
17. Hassani, A., Walton, S., Li, J., Li, S., Shi, H.: Neighborhood attention transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6185–6194 (2023)
18. Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T.B., Dhariwal, P., Gray, S., et al.: Scaling laws for autoregressive generative modeling. arXiv preprint arXiv:2010.14701 (2020)
19. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
20. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de las Casas, D., Hendricks, L.A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J.W., Sifre, L.: An empirical analysis of compute-optimal large language model training. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)
21. Hong, S., Jung, J., Shin, H., Han, J., Yang, J., Luo, C., Kim, S.: Pf3plat: Pose-free feed-forward 3d gaussian splatting. arXiv preprint arXiv:2410.22128 (2024)
22. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023)
23. Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., Zheng, Z., Fang, Y., Huang, Y., Zhao, W., et al.: Minicpm: Unveiling the potential of small language models with scalable training strategies. arXiv preprint arXiv:2404.06395 (2024)
24. Huang, R., Mikolajczyk, K.: No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. arXiv preprint arXiv:2508.01171 (2025)
25. Huang, R., Mikolajczyk, K.: Spfsplatv2: Efficient self-supervised pose-free 3d gaussian splatting from sparse views. arXiv preprint arXiv:2509.17246 (2025)
26. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017)
27. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 406–413. IEEE (2014)
28. Jiang, H., Tan, H., Wang, P., Jin, H., Zhao, Y., Bi, S., Zhang, K., Luan, F., Sunkavalli, K., Huang, Q., Pavlakos, G.: Rayzer: A self-supervised large view synthesis model (2025)
29. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: Lvsm: A large view synthesis model with minimal 3d inductive bias. In: The Thirteenth International Conference on Learning Representations (2025)
30. Kang, G., Yoo, J., Park, J., Nam, S., Im, H., Shin, S., Kim, S., Park, E.: Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22012–22022 (2025)
31. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
32. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)

33. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
34. Klinger, E., Starkweather, D.: pHash: The open source perceptual hash library. `https://www.phash.org/` (2010)
35. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)
36. Lai, Z., Liu, S., Efros, A.A., Wang, X.: Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9730–9740 (2021)
37. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024)
38. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10486–10496 (2025)
39. Li, Z., Dong, C., Chen, Y., Huang, Z., Liu, P.: Vicasplat: A single run is all you need for 3d gaussian splatting and camera estimation from unposed video frames. arXiv preprint arXiv:2503.10286 (2025)
40. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
41. Liu, S., Ng, K.W., Jang, W., Guo, J., Han, J., Liu, H., Douratsos, Y., Pérez, J.C., Zhou, Z., Phung, C., et al.: Scaling sequence-to-sequence generative neural rendering. arXiv preprint arXiv:2510.04236 (2025)
42. Liu, X., Tayal, P., Wang, J., Zarzar, J., Monnier, T., Tertikas, K., Duan, J., Toisoul, A., Zhang, J.Y., Neverova, N., Vedaldi, A., Shapovalov, R., Novotny, D.: Uncommon objects in 3d. In: arXiv (2024)
43. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
44. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) (2019)
45. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision. pp. 405–421. Springer (2020)
46. Mitchel, T.W., Ryu, H., Sitzmann, V.: True self-supervised novel view synthesis is transferable. arXiv preprint arXiv:2510.13063 (2025)
47. Nair, N.G., Kaza, S., Luo, X., Patel, V.M., Lombardi, S., Park, J.: Scaling transformer-based novel view synthesis with models token disentanglement and synthetic data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28567–28576 (2025)
48. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
49. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016)
50. Plucker, J.: Xvii. on a new geometry of space. Philosophical Transactions of the Royal Society of London (155), 725–791 (1865)

51. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
52. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: International Conference on Computer Vision (2021)
53. Rombach, R., Esser, P., Ommer, B.: Geometry-free view synthesis: Transformers and no 3d priors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14356–14366 (2021)
54. Safin, A., Duckworth, D., Sajjadi, M.S.M.: Repast: Relative pose attention scene representation transformer (2023)
55. Sajjadi, M.S., Duckworth, D., Mahendran, A., Van Steenkiste, S., Pavetic, F., Lucic, M., Guibas, L.J., Greff, K., Kipf, T.: Object scene representation transformer. Advances in neural information processing systems **35**, 9512–9524 (2022)
56. Sajjadi, M.S., Mahendran, A., Kipf, T., Pot, E., Duckworth, D., Lučić, M., Greff, K.: Rust: Latent neural scene representations from unposed imagery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17297–17306 (2023)
57. Sajjadi, M.S., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lučić, M., Duckworth, D., Dosovitskiy, A., et al.: Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6229–6238 (2022)
58. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
59. Seitzer, M., van Steenkiste, S., Kipf, T., Greff, K., Sajjadi, M.S.M.: DyST: Towards dynamic neural scene representations on real-world videos. In: The Twelfth International Conference on Learning Representations (2024)
60. Shen, Y., Zhang, Z., Qu, Y., Cao, L.: Fastvggt: Training-free acceleration of visual geometry transformer. arXiv preprint arXiv:2509.02560 (2025)
61. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3 (2025), https://arxiv.org/abs/2508.10104
62. Smith, C., Du, Y., Tewari, A., Sitzmann, V.: Flowcam: Training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. arXiv preprint arXiv:2306.00180 (2023)
63. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: Roformer: enhanced transformer with rotary position embedding. corr abs/2104.09864 (2021). arXiv preprint arXiv:2104.09864 (2021)
64. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A.,

Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open foundation and fine-tuned chat models (2023)
65. Usenko, V., Demmel, N., Cremers, D.: The double sphere camera model. In: 2018 International Conference on 3D Vision (3DV). pp. 552–560. IEEE (2018)
66. Wang, F., Yu, Y., Wei, G., Shao, W., Zhou, Y., Yuille, A., Xie, C.: Scaling laws in patchification: An image is worth 50,176 tokens and more. arXiv preprint arXiv:2502.03738 (2025)
67. Wang, H.: Open-Rayzer: a open-source Self-Reimplemented Version of the paper "RayZer: A Self-supervised Large View Synthesis Model" — github.com. `https://github.com/ou524u/Open-Rayzer` (2025)
68. Wang, H., Ye, K., Li, Y., Chen, W., Chen, B.: The less you depend, the more you learn: Synthesizing novel views from sparse, unposed images without any 3d knowledge. arXiv preprint arXiv:2506.09885 (2025)
69. Wang, J., Yuan, Y., Zheng, R., Lin, Y., Gao, J., Chen, L.Z., Bao, Y., Zhang, Y., Zeng, C., Zhou, Y., et al.: Spatialvid: A large-scale video dataset with spatial annotations. arXiv preprint arXiv:2509.09676 (2025)
70. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
71. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Vggsfm: Visual geometry grounded deep structure from motion (2023)
72. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025)
73. Wang, R., Ma, Y., Gao, S.: Recollection from pensieve: Novel view synthesis via learning from uncalibrated videos. arXiv preprint arXiv:2505.13440 (2025)
74. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024)
75. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: $\pi^3$: Scalable permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025)
76. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
77. Watson, D., Chan, W., Martin-Brualla, R., Ho, J., Tagliasacchi, A., Norouzi, M.: Novel view synthesis with diffusion models. arXiv preprint arXiv:2210.04628 (2022)
78. Watson, D., Saxena, S., Li, L., Tagliasacchi, A., Fleet, D.J.: Controlling space and time with diffusion models. arXiv preprint arXiv:2407.07860 (2024)
79. Wu, R., Gao, R., Poole, B., Trevithick, A., Zheng, C., Barron, J.T., Holynski, A.: Cat4d: Create anything in 4d with multi-view video diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26057–26068 (2025)
80. Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., et al.: Reconfusion: 3d reconstruction with diffusion priors. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21551–21561 (2024)
81. Xia, H., Fu, Y., Liu, S., Wang, X.: Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22378–22389 (2024)

82. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16453–16463 (2025)
83. Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.H., Peng, S.: No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. arXiv preprint arXiv:2410.24207 (2024)
84. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis (2024)
85. Zhang, B., Sennrich, R.: Root mean square layer normalization. Advances in Neural Information Processing Systems **32** (2019)
86. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: European Conference on Computer Vision. pp. 1–19. Springer (2024)
87. Zhang, Q., Zhai, S., Martin, M.A.B., Miao, K., Toshev, A., Susskind, J., Gu, J.: World-consistent video diffusion with explicit 3d modeling. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21685–21695 (2025)
88. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
89. Zhao, Q., Tan, H., Wang, Q., Bi, S., Zhang, K., Sunkavalli, K., Tulsiani, S., Jiang, H.: E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training (2025)
90. Zhao, Q., Tan, H., Wang, Q., Bi, S., Zhang, K., Sunkavalli, K., Tulsiani, S., Jiang, H.: E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training. In: CVPR (2026)
91. Zhou, J.J., Gao, H., Voleti, V., Vasishta, A., Yao, C.H., Boss, M., Torr, P., Rupprecht, C., Jampani, V.: Stable virtual camera: Generative view synthesis with diffusion models. arXiv preprint arXiv:2503.14489 (2025)
92. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
93. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5745–5753 (2019)