

4DGT: Learning a 4D Gaussian Transformer Using Real-World Monocular Videos

Zhen Xu^{1,2,*} Zhengqin Li¹ Zhao Dong¹ Xiaowei Zhou² Richard Newcombe¹ Zhaoyang Lv¹

¹Reality Labs Research, Meta ²Zhejiang University

Project page: <https://4dgt.github.io>

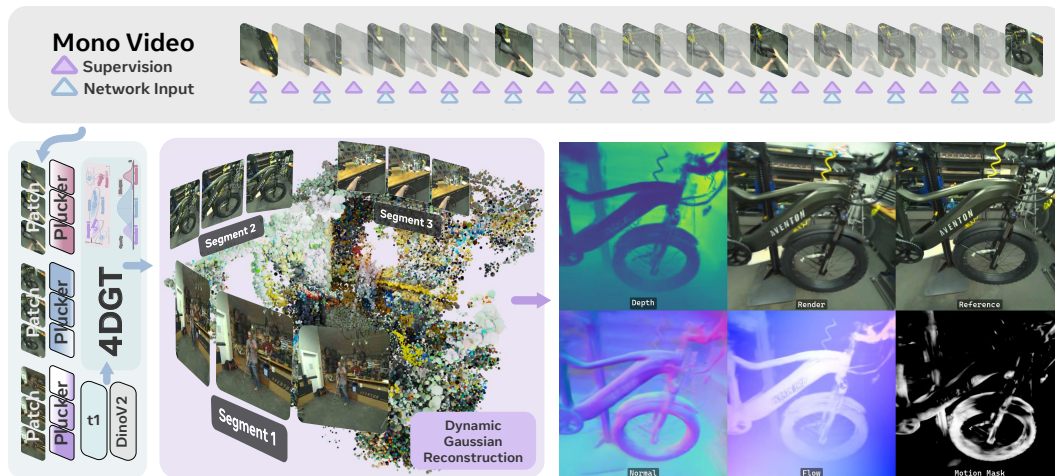


Figure 1: We propose a scalable 4D dynamic reconstruction model trained only on real-world monocular RGB videos. The feed-forward 4DGS (section 3.1) representation enables us to render the geometry and appearance of the dynamic scene from novel views in real-time. Even without explicit supervision, the model can learn to distinguish dynamic contents from the background and produce realistic optical flows. The figure shows an enlarged set of Gaussians for the purpose of visualization. **The embedded rendered videos only play in Adobe Reader or KDE Okular.**

Abstract

We propose 4DGT, a 4D Gaussian-based Transformer model for dynamic scene reconstruction, trained entirely on real-world monocular posed videos. Using 4D Gaussian as an inductive bias, 4DGT unifies static and dynamic components, enabling the modeling of complex, time-varying environments with varying object lifespans. We proposed a novel density control strategy in training, which enables our 4DGT to handle longer space-time input and remain efficient rendering at run-time. Our model processes 64 consecutive posed frames in a rolling-window fashion, predicting consistent 4D Gaussians in the scene. Unlike optimization-based methods, 4DGT performs purely feed-forward inference, reducing reconstruction time from hours to seconds and scaling effectively to long video sequences. Trained only on large-scale monocular posed video datasets, 4DGT can outperform prior Gaussian-based networks significantly in real-world videos and achieve on-par accuracy with optimization-based methods on cross-domain videos.

* Work done during internship at Meta.

1 Introduction

Humans record videos to digitize interactions with their surroundings. The ability to recover persistent geometry and 4D motion from videos has a profound impact on AR/VR, robotics, and content creation. Modeling 4D dynamic interactions from general-purpose videos remains a long-standing challenge. Prior work relying on multi-view synchronized capture or depth sensing is constrained to specific application domains. Recent progress in monocular dynamic video reconstruction via per-video optimization shows promise, but lacks scalability due to its time-consuming inference.

In this paper, we propose 4D Gaussian Transformer (4DGT), a novel transformer-based model that reconstructs dynamic scenes from posed monocular videos in a feedforward manner. We assume camera calibration and 6-degree-of-freedom (6DoF) poses are available from on-device SLAM [9] or offline pipelines [23, 31]. Inspired by recent feedforward reconstruction methods for static 3D scenes [69, 72], 4DGT learns reconstruction from data and adopts 4D Gaussian Splatting (4DGS) [55] as a unified representation for both static and dynamic content, differing only in lifespan. This design enables fast 4D reconstruction from short videos in seconds. For longer videos with global pose consistency, 4DGT predicts consistent world-aligned 4DGS using 64-frame rolling windows.

Training a 4D representation is challenging in defining appropriate supervision. While dynamic monocular videos are abundant, they lack space-time constraints. Multi-view video datasets [4, 22] are limited in both quantity and diversity, making them insufficient for training models that generalize in the wild. Prior methods [43] trained on synthetic object-level data suffer from a generalization gap when applied to complex real-world dynamics.

To address this, we train 4DGT exclusively on posed monocular videos from public datasets. We use two key strategies to mitigate space-time ambiguity. First, we leverage depth and normal predictions from expert models [40, 64, 66] as auxiliary supervision for guiding geometry learning. Second, we regularize predicted Gaussian properties to favor longer lifespans and reduce overfitting to specific views. These enable 4DGT to effectively disentangle space-time structure, yielding high-quality geometry, novel view synthesis, and emergent motion properties such as segmentation and flow. Our reconstructions also show better metric consistency than the expert models used for supervision.

Scaling a transformer to predict dense, pixel-aligned 4DGS presents two main challenges. First, dense pixel-aligned 4DGS predictions are computationally expensive for training and rendering. Inspired by density control in 3DGS [19], we introduce a pruning strategy that removes the redundant pixel-aligned 4DGS and further increases tokens in a second training with denser space-time samples. This effectively reduces 80% of Gaussians and enables a $16\times$ higher sampling rate with the same compute. Second, as space-time samples increase, the number of tokens grows, and vanilla self-attention scales quadratically. To address this, we propose a level-of-detail structure via multi-level spatiotemporal attention, achieving an additional $2\times$ reduction in computational cost.

4DGT is the first transformer-based method for predicting 4DGS in a feedforward manner using only real-world posed monocular videos in training. Extensive evaluations across datasets and domains show that 4DGT achieves comparable reconstruction quality to optimization-based methods while being three orders of magnitude faster, making it practical for long video reconstruction. Compared to prior methods that only train on synthetic object-level data [43], 4DGT generalizes better to complex real-world dynamics. Compared to the per-frame prediction pipeline [25], it also exhibits emergent motion properties.

In summary, we make the following technical contributions:

- We introduce 4DGT, a novel 4DGS transformer trained on posed monocular videos at scale, which produces consistent 4D video reconstructions in seconds at inference.
- We propose a training strategy to densify and prune space-time pixel-aligned Gaussians, reducing 80% of predictions, achieving $16\times$ higher sampling rate during training and a $5\times$ speed-up in rendering.
- We design a multi-level attention module to efficiently fuse space-time tokens, further reducing training time by half.
- Our experiments demonstrate strong scalability of 4DGT across real-world domains using mixed training datasets and can outperform the previous Gaussian network significantly. The performance of 4DGT is on par with optimization-based methods in accuracy in cross-domain videos recorded by similar devices used in training, while being 3 orders of magnitude faster.

2 Related Work

Nonrigid reconstruction. Recovering dynamic content from video has long been a holy grail challenge in 3D vision. Early approaches demonstrated promising non-rigid shape reconstruction from RGB-D videos [32, 47, 3], but relied heavily on depth input and struggled with complex dynamic scenes. Since the seminal work on Neural Radiance Fields (NeRF) [30], several methods have extended NeRF to 4D using multi-view videos [22, 10] or posed monocular videos [36, 37]. However, 4D NeRFs are slow to train and render, limiting their scalability for complex dynamic scenes. Recently, generative priors have shown promise in aiding 4D reconstruction [58], offering strong regularization for shape and motion across space and time. Still, optimizing 4D representations remains time-consuming, and reconstruction quality depends heavily on the generalizability of priors across different scene domains.

Dynamic Gaussian representations. Since the introduction of 3D Gaussian Splatting [19], several methods have extended it to 4DGS variants [65, 62, 55, 8], showing promising dynamic scene reconstruction from multi-view videos with faster training and real-time rendering support. However, optimizing dynamic Gaussians for monocular videos remains challenging. Recently, a few works have shown that complex 4D scenes composed of moving Gaussians can be recovered by leveraging depth, segmentation, and tracking priors from 2D expert models [57, 53, 21]. Despite strong performance, these methods involve complex processing, including manual annotation on dynamic regions. It further requires lengthy optimization, limiting its scalability in practical applications.

Large reconstruction models. Transformer-based 3D large reconstruction models (LRMs) have shown strong potential for learning high-quality 3D reconstruction from data, at the object level [14, 16, 35, 60, 24] and static scenes [59, 69, 72, 61]. LRMs can generate reconstructions in seconds from a few input views, achieving quality comparable to optimization-based neural methods. However, training LRMs requires large-scale multi-view supervision of the same instance, which is readily available in synthetic datasets or static scene captures, but remains scarce for real-world videos.

Some recent efforts explored training transformers to predict time-dependent 3D-GS [43, 25, 63, 41] using animated synthetic data [43], self-curated real-world internet videos [25] and street-level data [63], making them the closest related works in motivation. In contrast to these methods, which predict time-dependent 3D-GS representations, our 4DGT offers a holistic 4D scene representation that captures geometry better and enables motion understanding capabilities lacking in prior approaches. [63] requires multi-camera input and only focuses on street-level scenes. Compared to training on synthetic data in [43], our real-world training approach generalizes better to real-world scenes. Compared to B-Timer [25], which adopts a per-frame Gaussian prediction pipeline, our method produces explicit dynamic Gaussians thus can model explicit motion, showing emergent capabilities like motion segmentation. Compared to Pred. 3D Repr. [41], which adopts a tri-plane based implicit representation, our Gaussian model enables fast rendering after the feed-forward reconstruction.

3 Method

Given a posed monocular video, 4DGT uses a transformer to predict a set of 4DGS, which can be rendered in real time. We first describe the architecture and dynamic scene representation in section 3.1. To enable efficient training and rendering at scale, we introduce a pixel density control strategy and a level-of-detail structure based on spatial attention. Both techniques improve space-time sampling rates under fixed compute budgets. In section 3.3, we detail our training process and regularization strategies designed to resolve space-time ambiguities in monocular videos.

3.1 Feed-Forward Dynamic Gaussian Prediction

Input encoding. Given a posed monocular video, we extract a set of image frames $\mathbf{I}_i$ with camera calibration $\mathbf{P}_i$ and timestamp $\mathbf{T}_i$, denoted as $\{\mathbf{I}_i \in R^{H \times W \times 3}, \mathbf{P}_i \in R^{H \times W \times 6}, \mathbf{T}_i \in R^{H \times W \times 1} | i = 1 \dots N\}$, where $\mathbf{P}_i$ represents the Plücker coordinates [18] and N is the total number of frames. We convert them into patches. For frame i , the patches are denoted as $\{\mathbf{I}_{i,j} \in R^{p \times p \times 3} | j = 1 \dots HW/p^2\}$, $\{\mathbf{T}_{i,j}\}$ and $\{\mathbf{P}_{i,j}\}$, where p is the patch size.

Feature fusion. We use the pretrained DINOv2 image encoder [33] to extract high-level C -dimensional features $\mathbf{F}_{i,j} \in R^C$. These are concatenated with the temporal and spatial encoding $\mathbf{T}_{i,j}$

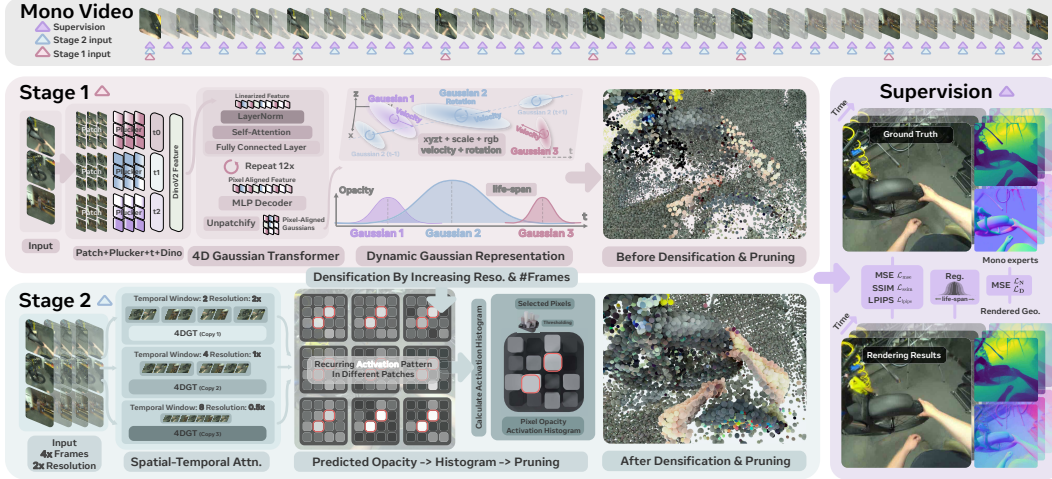


Figure 2: An overview of our method in training and rendering. 4DGT takes a series of monocular frames with poses as input. During training, we subsample the temporal frames at different granularity and use all images in training. We first train 4DGT to predict pixel-aligned Gaussians at coarse resolution in stage one. In stage two training, we pruned a majority of non-activated Gaussians according to the histograms of per-patch activation channels, and densify the Gaussian prediction by increasing the input token samples in both space and time. At inference time, we run the 4DGT network trained after stage two. It can support dense video frames input at high resolution.

and $\mathbf{P}_{i,j}$ as well as the input RGB image $\mathbf{I}_{i,j}$ to form the fused transformer input:

$$\{\mathbf{X}_{i,j}\} = \mathcal{F}(\{\mathbf{I}_{i,j} \oplus \mathbf{T}_{i,j} \oplus \mathbf{P}_{i,j} \oplus \mathbf{F}_{i,j} \mid i = 1 \cdots N, j = 1 \cdots HW/p^2\}), \quad (1)$$

where $\oplus$ denotes the concatenation operation and $\mathcal{F}$ denotes the all-to-all self-attention transformer module. In contrast to ViT, input used static LRMs in that it uses only Plücker rays [69] or DINO feature [14], our transformer takes timestamp-aware Plücker rays with DINO feature together as input, which we found to be beneficial to provide the best prediction in view synthesis as well as geometry prediction.

Dynamic Gaussians. We use a variant of 4DGS [65, 55] to unify the various components in dynamic scene predictions. To better represent geometry, we adopt the 2DGS [15] defined by the center $\mathbf{x} \in R^3$, scale $\mathbf{s} \in R^2$, opacity $\mathbf{o} \in R^1$ and orientation $\mathbf{q} \in R^4$ (quaternion) of the Gaussians. Compared to 3DGS[19] used in previous work [65, 55], 2DGS yields better geometry predictions. To represent motion, we use four temporal attributes, namely the temporal center $\mathbf{c} \in R^1$, life-span $\mathbf{l} \in R^1$, velocity $\mathbf{v} \in R^3$, and angular velocity $\boldsymbol{\omega} \in R^3$ (as angle-axis) for each Gaussian. Given a specific timestamp t_s for rendering a particular dynamic Gaussian point $\mathbf{g} = \{\mathbf{x}, \mathbf{s}, \mathbf{q}, \mathbf{o}, \mathbf{c}, \mathbf{l}, \mathbf{v}, \boldsymbol{\omega}\}$, we first calculate the offset to the opacity, location and orientation of the Gaussian point from the temporal attributes [65]. Specifically, the life-span $\mathbf{l}$ is used to influence the Gaussian opacity $\mathbf{o}$ over time:

$$\sigma = \sqrt{-\frac{1}{2} \cdot \frac{(1/2)^2}{\log(o_{th})}}, \quad \mathbf{o}_{t_s} = \mathbf{o} \cdot e^{-\frac{1}{2} \cdot \frac{(t_s - \mathbf{c})^2}{\sigma^2}}, \quad (2)$$

where o_{th} is the opacity multiplier at the life-span boundary and σ is the standard deviation of the Gaussian distribution in the temporal domain. Intuitively, the Gaussian retains its full opacity at its temporal center and fades in a Gaussian distribution along the temporal axis. At 1/2 time relative to the temporal center $\mathbf{c}$, the opacity of the point is reduced by multiplying a small factor o_{th} , which is set to 0.05 for all experiments. The location and orientation of each Gaussian is adjusted by the velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$ to account for the motion:

$$\mathbf{x}_{t_s} = \mathbf{x} + \mathbf{v} \cdot (t_s - \mathbf{c}), \quad \mathbf{q}_{t_s} = \mathbf{q} \cdot \phi(\boldsymbol{\omega} \cdot (t_s - \mathbf{c})), \quad (3)$$

where ϕ denotes converting the angle-axis representation to a quaternion. For each 2DGS with $\mathbf{x}_{t_s}, \mathbf{s}_{t_s}, \mathbf{q}_{t_s}, \mathbf{o}_{t_s}$ at timestamp t_s , we apply 2DGS rasterizer implemented in [67] to render image.

Dynamic Gaussian decoding. Given the pixel-aligned features $\{\mathbf{X}_{i,j}\}$, we use a transformer to decode the pixel-aligned 4DGS for each frame as

$$\{\mathbf{G}_{i,j}\} = \mathcal{D}_{\mathbf{x},\mathbf{s},\mathbf{q},\mathbf{o},\mathbf{c},\mathbf{l},\mathbf{v},\boldsymbol{\omega}}(\{\mathbf{X}_{i,j}|i = 1 \cdots N, j = 1 \cdots HW/p^2\}), \quad (4)$$

where $\mathcal{D}_{\mathbf{x},\mathbf{s},\mathbf{q},\mathbf{o},\mathbf{c},\mathbf{l},\mathbf{v},\boldsymbol{\omega}}$ denotes the MLP-based decoder head for producing the full suite of dynamic Gaussian parameters $\{\mathbf{G}_{i,j}\}$.

Our proposed 4DGS can unify the prediction of appearance and geometry properties of both static and dynamic elements. For static scenes, the network can learn to predict Gaussians with a long-living lifespan $\mathbf{l} \rightarrow \infty, \mathbf{v} \rightarrow 0, \boldsymbol{\omega} \rightarrow 0$. For complex dynamic motions with occlusions, it predicts transient dynamic objects with short-living Gaussians $\mathbf{l} \rightarrow 0$.

3.2 Multi-level Pixel & Token Density Control

While pixel-aligned Gaussian has been a standard choice in prior work [69, 72], it has a severe limitation in representing video frames that require dense, long-term sampling to capture motion effectively. Naively sampling frames spatial-temporally would result in two key issues. First, the increasing number of aligned Gaussians degrades optimization and rendering performance, leading to suboptimal training and blurry details in dynamic regions. Second, the growing number of input tokens significantly increases computational cost, resulting in under-trained models.

Two-stage training. We introduce a two-stage approach to address these challenges. First, we train on coarsely sampled low-resolution images from scratch until convergence. In the second stage, inspired by 3DGS [19], we propose to filter pixel-aligned Gaussians by pruning low-opacity predictions per patch based on the histograms and increasing the token count to predict more Gaussians across space-time. Additionally, we introduce a multi-level spatiotemporal attention mechanism to further reduce the computational cost of self-attention layers.

Pruning. After the initial training stage at coarse resolution, we compute a histogram of activated Gaussians per patch and observe that only a few channels are activated. A similar pattern emerges in other pixel-aligned Gaussian methods [69], motivating us to decode only a small set of Gaussians using the activated channels. Formally, for each patch of Gaussian parameters $\mathbf{G}_{i,j}$, we consider the standard deviation of their opacity values $\mathbf{o}_{i,j} = \{\mathbf{o}_{i,j,k}|k = 1 \cdots p^2\}$:

$$\mu(\mathbf{o}_{i,j}) = \frac{1}{p^2} \sum_{k=1}^{p^2} \mathbf{o}_{i,j,k}, \quad \sigma(\mathbf{o}_{i,j}) = \sqrt{\mu(\mathbf{o}_{i,j}^2) - \mu(\mathbf{o}_{i,j})^2}, \quad (5)$$

where $\mu(\mathbf{o}_{i,j})$ is the mean of the opacity values and $\sigma(\mathbf{o}_{i,j})$ is the standard deviation. A particular pixel k is considered activated if it has a value larger than 1 unit of the standard deviation:

$$\mathbf{m}_{i,j,k} = \begin{cases} 1, & \mathbf{o}_{i,j,k} > \mu(\mathbf{o}_{i,j}) + \sigma(\mathbf{o}_{i,j}), \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

where k is the index of the pixel in the patch and $\mathbf{m}_{i,j,k}$ indicates whether the pixel is activated. A histogram $\mathbf{h}_{i,j}$ of all activation mask $\mathbf{m}_{i,j} = \{\mathbf{m}_{i,j,k}|k = 1 \cdots p^2\}$ for the patch output is:

$$\mathbf{H} = \sum_{i=1, j=1}^{N, HW/p^2} \mathbf{M}_{i,j}, \quad (7)$$

where $\mathbf{M}_{i,j} \in \mathbb{N}^{p^2}$ is the activation mask for patch (i, j) , $\mathbf{H} \in \mathbb{N}^{p^2}$ is the aggregated histogram of all activation masks, and p is the patch size. We select S channel from the histogram $\mathbf{h}_{i,j}$ for the patch output for all patches onward in training. This effectively implements an S/p^2 times reduction in the number of Gaussians for each patch, mimicking the pruning strategy of 3DGS [19]. We provide more in-depth analysis for this histogram-based pruning strategy compared to alternatives with visualizations in the appendix.

Densification. The predicted Gaussian number can naturally increase with more space-time token inputs, either in resolution per frame or temporal frame numbers. The initially trained model provides a good scaffolding for pixel-aligned Gaussians when we increase the input token number in space and time. In the second stage of training, we increase the spatial and temporal resolution by a factor

of R_s and R_t , respectively. Combining the densification process and pruning strategy, this would result in $R_s^2 \cdot R_t \cdot S/p^2$ times of the Gaussians compared to the first stage. We select $R_s = 2$, $R_t = 4$, $S = 10$, and $p = 14$ for all experiments, leading to only 80% of the original number of Gaussians while greatly increasing the sampling rate of space-time by 16 times.

Multi-level spatial-temporal attention. The number of patches participating in the self-attention module $\mathcal{F}$ increases by a factor of $R_s^2 \cdot R_t$, which will slow down optimization and inference significantly. To mitigate this, we propose a temporal level-of-detail attention mechanism to reduce the computational cost. We propose to divide the N input frames into M equal trunks in the highest level. This division limits the attention mechanism in the temporal dimension, but reduces the computation of calculating n total tokens to $O(\frac{n^2}{M})$. To balance spatial-temporal samples, we construct a temporal level-of-detail structure by alternating the temporal range and spatial resolution, achieving a much smaller overhead while maintaining the ability to handle long temporal windows. For each level l , we reduce the spatial resolution by a factor of 2^l and increase temporal samples by 2. Empirically, we use level $L = 3$ and $M = 4$, which leads to an approximately 2 times reduction in the computational cost.

3.3 Training

Loss and regularization. We train 4DGT using segments of $W = 128$ consecutive frames from the monocular video and subsample every 8 frames as input, resulting in $N = 16$ input frames. Notably, for the second stage training where we apply techniques mentioned in section 3.2 and section 3.2, we increase the number of input frames to $N = 64$. After obtaining all Gaussian parameters $\{\mathbf{G}_{i,j}\}$ from each of the N input frames, we render them to all $W = 128$ images for self-supervision and compute the MSE loss. Additionally, we add the perceptual LPIPS loss [17] $\mathcal{L}_{\text{lpiips}}$ and SSIM loss [56] $\mathcal{L}_{\text{ssim}}$ for better perceptual quality.

$$\mathcal{L}_{\text{mse}} = \sum_{i=1}^W \frac{\|\mathbf{I}_i - \mathbf{I}'_i\|_2}{W}, \quad \mathcal{L}_{\text{lpiips}} = \sum_{i=1}^W \frac{\|\psi(\mathbf{I}_i) - \psi(\mathbf{I}'_i)\|_1}{W}, \quad \mathcal{L}_{\text{ssim}} = \sum_{i=1}^W \frac{\text{SSIM}(\mathbf{I}_i, \mathbf{I}'_i)}{W}, \quad (8)$$

where $\mathbf{I}_i$ denotes the input image and $\mathbf{I}'_i$ is the rendered image, ψ is the pre-trained layers AlexNet [20] and SSIM is the SSIM function [56]. To better regularize the training, we encourage the points to be static and have a long lifespan using:

$$\mathcal{L}_{\mathbf{v}} = \sum_{i=1, j=1, k=1}^{N, HW/p^2, p^2} \frac{\|\mathbf{v}_{i,j,k}\|_1}{NHW}, \quad \mathcal{L}_{\omega} = \sum_{i=1, j=1, k=1}^{N, HW/p^2, p^2} \frac{\|\omega_{i,j,k}\|_1}{NHW}, \quad \mathcal{L}_1 = \sum_{i=1, j=1, k=1}^{N, HW/p^2, p^2} \frac{\|\frac{1}{\mathbf{l}_{i,j,k}}\|_1}{NHW}, \quad (9)$$

where $\mathbf{v}_{i,j,k}$ is the velocity of the Gaussian point, $\omega_{i,j,k}$ is the angular velocity of the Gaussian point and $\mathbf{l}_{i,j,k}$ is the life-span of the Gaussian point.

Expert guidance. We observe that training can benefit from leveraging monocular export models in geometry prediction. We extract the depth map $\mathbf{D}_i$ and normal map $\mathbf{N}_i$ from all W frames using DepthAnythingV2 [64] and StableNormal [66] and use them as a pseudo-supervision signal:

$$\mathcal{L}_{\mathbf{D}} = \sum_{i=1}^W \frac{\|\mathbf{D}_i - \mathbf{D}'_i\|_2}{W}, \quad \mathcal{L}_{\mathbf{N}} = \sum_{i=1}^W \frac{\|\mathbf{N}_i - \mathbf{N}'_i\|_2}{W}, \quad (10)$$

where $\mathbf{D}'_i$ and $\mathbf{N}'_i$ are the predicted depth and normal map rendered using the 2DGS rasterizer [15]. The final loss function for training the feed-forward prediction pipeline is:

$$\mathcal{L} = \mathcal{L}_{\text{mse}} + \lambda_{\text{lpiips}} \mathcal{L}_{\text{lpiips}} + \lambda_{\text{ssim}} \mathcal{L}_{\text{ssim}} + \lambda_{\mathbf{v}} \mathcal{L}_{\mathbf{v}} + \lambda_{\omega} \mathcal{L}_{\omega} + \lambda_1 \mathcal{L}_1 + \lambda_{\mathbf{D}} \mathcal{L}_{\mathbf{D}} + \lambda_{\mathbf{N}} \mathcal{L}_{\mathbf{N}}, \quad (11)$$

where λ_{lpiips} , λ_{ssim} , $\lambda_{\mathbf{v}}$, λ_{ω} , λ_1 , $\lambda_{\mathbf{D}}$ and $\lambda_{\mathbf{N}}$ are the weights for the corresponding loss functions. We set $\lambda_{\text{lpiips}} = 2.0$, $\lambda_{\text{ssim}} = 0.2$, $\lambda_{\mathbf{v}} = 1.0$, $\lambda_{\omega} = 1.0$, $\lambda_1 = 1.0$, $\lambda_{\mathbf{D}} = 0.1$ and $\lambda_{\mathbf{N}} = 0.01$ for all experiments. All weights for the regularization losses are warmed up linearly from 0 to their final values during the first 2500 iterations of training.

4 Implementation Detail

Architecture. We use a modified ViT architecture [7] for our fusion network $\mathcal{F}$. Specifically, we use 12 layers of all-to-all self-attention with 16 heads, each head having a hidden dimension of 96,

and the fully connected layers have a $4\times$ wider hidden channel size. Since the Plücker coordinates [18] $\mathbf{P}$ and timestamps $\mathbf{T}$ already provide the 4D position of each pixel, we do not use additional embedding for the positional information. For the second stage training, where we enable the multi-level spatial-temporal attention module, we copy the weights of the first-stage transformer L times and train them independently. The l -th transformer is responsible for the l -th level of spatial-temporal attention, with 1 classification token for passing information between different levels. For the MLP decoders $\mathcal{D}$, we use 2 fully connected layers with a hidden dimension of 256 for each channel. Both the transformer modules $\mathcal{F}$ and $\mathcal{G}$ use GELU [13] as the activation function and layer normalization [1] as the normalization function. We also disable the bias parameters for all the layers.

Training & Inference. We implement 4DGT in PyTorch framework [38]. We employ FlashAttentionV3 [45] and the GSplat Rasterizer [67] for efficient attention and Gaussian optimization respectively. For optimization, we use the AdamW optimizer [27] with a learning rate of $5e^{-4}$ and a weight decay of 0.05. For the second stage training, the learning rate is set to $1e^{-5}$. Additionally, we linearly warm-up the learning rate of each stage in the first 2500 steps and then apply the cosine decaying schedule [26] for the remaining steps. During the second stage training, we additionally augment the input and output to the network by varying the aspect ratio and field of view of the images. Specifically, we randomly sample an aspect ratio from the uniform distribution on $[\frac{1}{3}, \frac{3}{1}]$ and a field of view ratio on the original image on $[30\%, 100\%]$. We train our reconstruction model 100k iterations for the first stage and 30k iterations for the second stage, using a total batch size of 64. With 64 Nvidia H100 GPUs, the first stage training takes roughly 9 days and the second stage training takes roughly 6 days. For all other experiments on inference speed, we use a single 80 GB A100 GPU.

5 Experiments

Training Datasets. We use the following real-world monocular videos with high-quality calibrations:

- Project Aria datasets with closed-loop trajectories: the EgoExo4D [12], Nymeria [29], Hot3D [2] and Aria Everyday Activities (AEA) [28].
- Video data with COLMAP [44] camera parameters: Epic-Fields [50, 5] and Cop3D [46].
- Phone videos with ARKit camera poses: ARKitTrack [71].

Evaluation datasets. We use the synthetic rendering provided in ADT [34] datasets, which provides metric ground truth depth. To evaluate cross-domain generalization, we use DyCheck [11] (DyC) datasets and the dynamic scene in TUM-SLAM [48] (TUM) to evaluate novel view synthesis. We further hold out a test split from EgoExo4D, AEA, and Hot3D, which we refer to as the Aria test set.

Metrics. For appearance evaluation, we compare the PSNR and LPIPS [70] metrics on novel view and time rendering results. For geometry evaluation, we compare against the depth RMSE [64] and normal angle error [66]. We additionally provide qualitative comparisons of motion rendered in 2D as optical flow and motion segmentation. All comparison experiments are conducted on 128-frame subsequences of the monocular videos, with 64 frames used as input and the remaining 64 frames used for testing, with images resized to 504×504 resolution or a similar pixel number for controlled comparison unless specified otherwise. For the DyCheck [11] dataset, we additionally compare the rendering results on the provided test view cameras, which show signals on extreme view synthesis. We provide more details about evaluation implementations in the appendix.

Baselines. We consider the following baselines as the most relevant work for evaluation.

1. **L4GM [43]:** It is the closest prior 4D Gaussian model that generalizes to real-world videos. Different from ours trained using real-world data only, they trained on a synthetic dataset and leveraged additional multi-view diffusion priors from ImageDream [52].
2. **Static-LRM:** We trained a static scene LRM following [69] on the same real world data as our 4DGT. We use 2DGS instead of 3DGS as the representation that shows more similarity to our approach, except that we further model the dynamic content.
3. **Expert monocular models:** We compared each individual expert model we used during training in the same setting, including DepthAnythingV2 [64] aligned with the metric scale of UniDepth [39] and normals provided by StableNormal [66]. For novel view evaluation, we unproject the image using the nearest depth and normal frame.



Figure 3: From left-to-right, we show the novel space-time view comparisons on ADT [34], EgoExo4D [12], DyCheck [11] and the DyCheck test-view (rightmost). We render the depth (upper right) and normal (below right) next to each synthesized novel view. For ground truth depth and normal on EgoExo4D and DyCheck, we use predictions from the expert models from the ground truth image for reference. Please refer to the appendix for more visual comparisons.

4. **MonST3R [68]**: We compare to the dynamic point based representation [68] which highlights the representation difference in using 4DGS. We use ground truth camera poses as input to their model using the official implementation and using PyTorch3D [42] for normal estimation.
5. **Shape of Motion (SoM) [53]**: We use SoM to represent the top-tier per-scene optimization method as a reference for best dynamic reconstruction quality. We follow SoM’s instructions to manually segment the dynamic part. It requires running expert models as input, including mask, depth, and tracking, which we do not use. We include the preprocessing time for time comparisons.

We do not make comparisons with CAT4D [58], BulletTimer [25] and Pred. 3D Repr. [41] since they provide neither the source code nor the pre-trained models.

Comparisons to baselines. Table 1 and Figure 3 present our comparisons to baselines. Compared to L4GM, which also predicts dynamic Gaussians from a trained transformer, 4DGT shows much better generalization across real scenes. We found that a static-LRM can provide a strong baseline for static scenes but will fail when dynamic motion is present, while 4DGT can do well in both. It is further validated in Table 2c on dynamic regions. The geometry predicted from 4DGT is more consistent with the world coordinate compared to expert models when evaluated in metric scale. Compared to the optimization-based method SoM, 4DGT can offer on-par quality in view synthesis as well

Table 1: Comparisons to baselines. We mark SoM in grey as a reference for optimization-based methods, and rank the other baselines with ours as comparisons in learning based approaches.

Method	PSNR (Render) $\uparrow$					LPIPS (Render) $\downarrow$					RMSE (Depth) $\downarrow$			Deg. $\downarrow$	Recon. Time $\downarrow$
	ADT	TUM	DyC	Aria	Avg	ADT	TUM	DyC	Aria	Avg	ADT	TUM	Avg	ADT	
SoM [53]	± 1.640	± 0.266	± 2.597	± 3.701	23.63	± 0.024	± 0.069	± 0.060	± 0.067	0.320	± 3.000	± 1.354	± 1.633	± 3.607	60000 ms / f
L4GM [43]	± 2.713	± 0.112	± 0.629	± 1.666	8.490	± 0.051	± 0.013	± 0.062	± 0.052	0.661	± 0.660	± 0.791	± 0.386	± 4.181	200 ms / f
MonST3R [68, 66]	± 1.651	± 1.673	± 1.673	± 3.355	19.24	± 0.011	± 0.023	± 0.045	± 0.135	0.318	± 0.842	± 0.243	± 0.543	± 2.699	4500 ms / f
Experts [40, 66]	± 3.195	± 3.117	± 1.258	± 2.419	18.96	± 0.082	± 0.073	± 0.058	± 0.046	0.319	± 0.621	± 0.074	± 0.348	± 0.748	350 ms / f
Ours	± 1.508	± 0.048	± 2.034	± 1.591	23.20	± 0.021	± 0.009	± 0.067	± 0.019	0.308	± 0.463	± 0.048	± 0.255	± 1.831	25 ms / f

Table 2: Ablation study on our method components using ADT and DyCheck (DyC).

(a) Ablation on dynamic Gaussian in stage one training.

Method	PSNR $\uparrow$			LPIPS $\downarrow$			RMSE ADT $\downarrow$	Deg. ADT $\downarrow$
	ADT	DyC	Avg	ADT	DyC	Avg		
Naive	15.49	13.95	14.72	0.612	0.517	0.564	1.156	42.25
+ EgoExo4D [12]	22.72	15.27	19.00	0.229	0.385	0.307	1.278	41.11
Static LRM [69]	19.29	14.21	16.75	0.399	0.463	0.431	0.830	31.96
Per-frame [43]	10.07	12.10	11.08	0.748	0.714	0.731	3.117	61.77
+ $\mathcal{L}_{N,D,l,\omega,v}$	26.45	15.86	21.15	0.170	0.399	0.284	0.773	21.59

(b) Ablation on stage two training.

Method	PSNR $\uparrow$			LPIPS $\downarrow$			RMSE ADT $\downarrow$	Deg. ADT $\downarrow$
	ADT	DyC	Avg	ADT	DyC	Avg		
Naive	Out of Memory							
Random.	25.79	14.97	20.38	0.333	0.480	0.406	0.750	25.84
+ D&P	28.79	15.52	22.16	0.242	0.434	0.338	0.722	25.95
+ Multi-level	28.71	15.34	22.03	0.242	0.439	0.341	0.783	27.30
+ Mix. (Ours)	28.31	16.12	22.22	0.243	0.408	0.326	0.934	25.92

(c) Evaluation on the dynamic foreground.

Method	PSNR $\uparrow$			LPIPS $\downarrow$			RMSE ADT $\downarrow$	Deg. ADT $\downarrow$
	ADT	DyC	Avg	ADT	DyC	Avg		
Static LRM [69] (masked)	17.30	13.56	16.76	0.059	0.220	0.102	0.613	49.35
Ours (masked)	27.29	14.93	22.86	0.030	0.195	0.075	0.388	33.32

(d) Motion segmentation results.

Method	w/o $\mathcal{L}_{v,\omega,l}$	MegaSaM [23]	Ours
mIoU $\times 100 \uparrow$	9.4 ± 4.1	77.4 ± 4.0	81.2 ± 1.8

as geometry prediction while being 3 orders of magnitude faster in runtime, which makes it more favorable to process long-time videos in practice.

Ablation study in stage one training. In Table 2a for stage one training, we start from a *Naive* training baseline at coarse resolution using the image rendering losses in Eq. 8 trained only using the AEA dataset with only 7 hours of data. After further scaling to using the EgoExo4D dataset (300 hours), we find that increasing the scale of the dataset can significantly improve the performance. We also compare to a static LRM [69] and per-frame LRM [43] counterpart in the same setting. We can already see the benefits over the baselines at this stage by a large margin. We further include the full training loss in Eq. 9 and Eq. 11, and we can see significant improvements in the quality across all metrics. The regularization terms $\mathcal{L}_{N,D,l,\omega,v}$ can be categorized into two groups: (1) $\mathcal{L}_{v,\omega,l}$ regularizes the motion of the dynamic Gaussian predictions. Without this term, although the quantitative results are not greatly affected, the model would fall into the trivial local minima of making every Gaussian transient and dynamic, not correctly modeling the scene’s static or slow-moving parts. This would result in a purely-white motion mask. In table 2d, we evaluate the quality of the motion mask on the ADT dataset [34] of our method without this term. Thanks to the explicit modeling of the motion parameters in our representation, our model can produce comparable motion segmentation against MegaSaM [23], which has explicit flow supervision, while being 200 $\times$ faster (1.5s v.s. 300s). (2) $\mathcal{L}_{N,D}$ provides expert guidance for the geometry of the dynamic Gaussian prediction. These terms can greatly improve the quality of the reconstructed geometry. Please refer to the appendix for visual comparisons.

Ablation study in stage two training. Table. 2b shows the ablation study of key design in stage two training. Starting from a *Naive* model without density control to prune and densify (*D&P*) Gaussians and the *multi-level* attention proposed in 3.2, naively scaling up resolution and temporal samples will run out of memory in training. Compared with variants using a *Random* sampled Gaussian from predicted patches and further densifying, using the proposed *D&P* strategy to decode sparse activated Gaussian will lead to better results while being efficient in training. Adding the proposed multi-level attention can further speed up training with only a minor sacrifice in quality, but can speed up training two times faster. Finally, we mixed all the proposed datasets in training at stage two. Compared to model training only using EgoExo4D, mixing datasets improves generalization across domains.

Table 3: Comparison with a more comprehensive upper-bound on the ADT dataset [34].

Method	PSNR $\uparrow$	LPIPS $\downarrow$	RMSE $\downarrow$	Degree $\downarrow$	Recon. Time $\downarrow$
SoM [53]	30.30 $\pm$ 1.64	0.242 $\pm$ 0.024	4.16 $\pm$ 3.00	34.07 $\pm$ 3.61	60,000 ms/f
SoM* [53, 19]	28.40 $\pm$ 2.60	0.281 $\pm$ 0.027	4.18 $\pm$ 2.98	18.46 $\pm$ 1.76	60,000 ms/f
Ours	28.31 $\pm$ 1.51	0.243 $\pm$ 0.021	0.93 $\pm$ 0.46	25.94 $\pm$ 1.84	25 ms/f
Ours_{tune10s}	31.98 $\pm$ 1.01	0.220 $\pm$ 0.012	0.85 $\pm$ 0.45	19.25 $\pm$ 2.65	25 + 150 ms/f

Initializing optimization-based methods with 4DGT. Table 3 presents a quantitative comparison on the ADT dataset [34] with stronger baselines and 4DGT-initialized optimization-based method. *SoM-2DGS-Geometry* augments SoM [53] with 2DGS [19] and employs the same normal regularization as ours to provide a stronger geometry upper-bound. This results in improved normal quality, but our feed-forward 4DGT prediction method still achieves comparable—or superior—results while being orders of magnitude faster. After finetuning the feed-forward prediction for only 10 seconds (100 iterations, 150 ms per frame, denoted as *Ours_{tune10s}*), performance further improves beyond all optimization-based baselines. While SoM and its 2DGS-augmented variant require 30,000 optimization steps per scene (and a sophisticated tracking expert like TAPIR [6]), our finetuned feed-forward prediction achieves a $350\times$ speedup, and pure feed-forward performance (*Ours*) is $2,400\times$ faster. More finetuning further improves reconstruction quality.

6 Conclusion

We introduced 4DGT, a novel dynamic scene reconstruction method that predicts 4DGS from an input posed video frame in a feed-forward manner. The representation power of 4DGT enables it to handle general dynamic scenes using 4DGS with varying lifespans, and support it to handle complex dynamics in long videos. Different from prior work that heavily depends on multi-view supervision from synthetic datasets, 4DGT is trained only using real-world monocular videos. We demonstrate that 4DGT can generalize well to videos recorded from similar devices, and the ability of generalization can improve when mixing datasets for training in scale.

Limitations and future work. We do not claim 4DGT can generalize to all videos in the wild. We assume the availability of a reliable calibration to train 4DGT and deploy it for inference. For this requirement, the training datasets have been limited to data sources from a few egocentric devices and phone captures. We observe that the quality may degrade in videos recorded by an unseen type of device due to the inaccurate metric scale calibrations. We believe this can be significantly improved by further scaling up the method using more diverse datasets recorded by different devices with curated calibrations. Similar to most monocular reconstruction methods, we still observe significant artifacts when viewing Gaussians from extreme view angles, departing far from the input trajectory. Future directions can propose better representations to address it or learn to distill more priors from multi-view expert models, such as generative video models.

Acknowledgments and Disclosure of Funding

We thank Aljaz Bozic for the insightful discussions. We thank the Project Aria team for their open-source and dataset contributions. This work was also partially supported by NSFC (No. U24B20154).

References

- [1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [2] P. Banerjee, S. Shkodrani, P. Moulon, S. Hampali, S. Han, F. Zhang, L. Zhang, J. Fountain, E. Miller, S. Basol, et al. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. *arXiv preprint arXiv:2411.19167*, 2024.
- [3] A. Bozic, M. Zollhofer, C. Theobalt, and M. Niessner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [4] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. Duvall, J. Dourgarian, J. Busch, M. Whalen, and P. Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020.
- [5] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018.
- [6] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023.
- [7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [8] Y. Duan, F. Wei, Q. Dai, Y. He, W. Chen, and B. Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.
- [9] J. Engel, K. Somasundaram, M. Goesele, A. Sun, A. Gamino, A. Turner, A. Talattof, A. Yuan, B. Souti, B. Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.
- [10] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023.
- [11] H. Gao, R. Li, S. Tulsiani, B. Russell, and A. Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022.
- [12] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.
- [13] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [14] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.
- [15] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.
- [16] Y. Jiang, L. Zhang, J. Gao, W. Hu, and Y. Yao. Consistent4d: Consistent 360 {deg} dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848*, 2023.
- [17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
- [18] P. Julius. On a new geometry of space. *Julius Plückers Gesammelte wissenschaftliche Abhandlungen, op. cit.*(24), 1:462–545, 1865.
- [19] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [21] J. Lei, Y. Weng, A. Harley, L. Guibas, and K. Daniilidis. MoSca: Dynamic gaussian fusion from casual videos via 4D motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024.

- [22] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022.
- [23] Z. Li, R. Tucker, F. Cole, Q. Wang, L. Jin, V. Ye, A. Kanazawa, A. Holynski, and N. Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv preprint arXiv:2412.04463*, 2024.
- [24] Z. Li, D. Wang, K. Chen, Z. Lv, T. Nguyen-Phuoc, M. Lee, J.-B. Huang, L. Xiao, C. Zhang, Y. Zhu, et al. Lirm: Large inverse rendering model for progressive reconstruction of shape, materials and view-dependent radiance fields. *arXiv preprint arXiv:2504.20026*, 2025.
- [25] H. Liang, J. Ren, A. Mirzaei, A. Torralba, Z. Liu, I. Gilitshenski, S. Fidler, C. Oztireli, H. Ling, Z. Gojcic, et al. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *arXiv preprint arXiv:2412.03526*, 2024.
- [26] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [27] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [28] Z. Lv, N. Charron, P. Moulon, A. Gamino, C. Peng, C. Sweeney, E. Miller, H. Tang, J. Meissner, J. Dong, et al. Aria everyday activities dataset. *arXiv preprint arXiv:2402.13349*, 2024.
- [29] L. Ma, Y. Ye, F. Hong, V. Guzov, Y. Jiang, R. Postyeni, L. Pesqueira, A. Gamino, V. Baiyya, H. J. Kim, et al. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *European Conference on Computer Vision*, pages 445–465. Springer, 2024.
- [30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [31] R. Murai, E. Dexheimer, and A. J. Davison. MAST3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *CVPR*, June 2025.
- [32] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015.
- [33] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [34] X. Pan, N. Charron, Y. Yang, S. Peters, T. Whelan, C. Kong, O. Parkhi, R. Newcombe, and Y. C. Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20133–20143, 2023.
- [35] Z. Pan, Z. Yang, X. Zhu, and L. Zhang. Efficient4d: Fast dynamic 3d object generation from a single-view video. *arXiv preprint arXiv:2401.08742*, 2024.
- [36] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.
- [37] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.
- [38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- [39] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024.
- [40] L. Piccinelli, C. Sakaridis, Y.-H. Yang, M. Segu, S. Li, W. Abbeloos, and L. Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025.
- [41] D. Qi, T. Yang, B. Wang, X. Zhang, and W. Zhang. Predicting 3d representations for dynamic scenes. *arXiv preprint arXiv:2501.16617*, 2025.
- [42] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.
- [43] J. Ren, C. Xie, A. Mirzaei, K. Kreis, Z. Liu, A. Torralba, S. Fidler, S. W. Kim, H. Ling, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024.

- [44] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [45] J. Shah, G. Bikshandi, Y. Zhang, V. Thakkar, P. Ramani, and T. Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37: 68658–68685, 2024.
- [46] S. Sinha, R. Shapovalov, J. Reizenstein, I. Rocco, N. Neverova, A. Vedaldi, and D. Novotny. Common pets in 3d: Dynamic new-view synthesis of real-life deformable categories. *CVPR*, 2023.
- [47] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *CVPR*, July 2017.
- [48] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.
- [49] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.
- [50] V. Tschernezki, A. Darkhalil, Z. Zhu, D. Fouhey, I. Laina, D. Larlus, D. Damen, and A. Vedaldi. Epic fields: Marrying 3d geometry and video understanding. *Advances in Neural Information Processing Systems*, 36: 26485–26500, 2023.
- [51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [52] P. Wang and Y. Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.
- [53] Q. Wang, V. Ye, H. Gao, J. Austin, Z. Li, and A. Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.
- [54] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state, 2025.
- [55] Y. Wang, P. Yang, Z. Xu, J. Sun, Z. Zhang, C. Yuan, H. Bao, S. Peng, and X. Zhou. Freetimegs: Free gaussians at anywhere for dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.
- [56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004.
- [57] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and W. Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023.
- [58] R. Wu, R. Gao, B. Poole, A. Trevithick, C. Zheng, J. T. Barron, and A. Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. *arXiv preprint arXiv:2411.18613*, 2024.
- [59] D. Xie, S. Bi, Z. Shu, K. Zhang, Z. Xu, Y. Zhou, S. Pirk, A. Kaufman, X. Sun, and H. Tan. Lrm-zero: Training large reconstruction models with synthesized data. *arXiv preprint arXiv:2406.09371*, 2024.
- [60] Y. Xie, C.-H. Yao, V. Voleti, H. Jiang, and V. Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024.
- [61] Y. Xu, Z. Shi, W. Yifan, H. Chen, C. Yang, S. Peng, Y. Shen, and G. Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024.
- [62] Z. Xu, Y. Xu, Z. Yu, S. Peng, J. Sun, H. Bao, and X. Zhou. Representing long volumetric video with temporal gaussian hierarchy. *ACM Transactions on Graphics*, 43(6), November 2024. URL <https://zju3dv.github.io/longvolcap>.
- [63] J. Yang, J. Huang, Y. Chen, Y. Wang, B. Li, Y. You, M. Igl, A. Sharma, P. Karkus, D. Xu, B. Ivanovic, Y. Wang, and M. Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2025.
- [64] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.
- [65] Z. Yang, H. Yang, Z. Pan, and L. Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2024.
- [66] C. Ye, L. Qiu, X. Gu, Q. Zuo, Y. Wu, Z. Dong, L. Bo, Y. Xiu, and X. Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024.
- [67] V. Ye, R. Li, J. Kerr, M. Turkulainen, B. Yi, Z. Pan, O. Seiskari, J. Ye, J. Hu, M. Tancik, et al. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

- [68] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.
- [69] K. Zhang, S. Bi, H. Tan, Y. Xiangli, N. Zhao, K. Sunkavalli, and Z. Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *European Conference on Computer Vision*, 2024.
- [70] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [71] H. Zhao, J. Chen, L. Wang, and H. Lu. Arkittrack: A new diverse dataset for tracking using mobile rgb-d data. *CVPR*, 2023.
- [72] C. Ziwen, H. Tan, K. Zhang, S. Bi, F. Luan, Y. Hong, L. Fuxin, and Z. Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint arXiv:2410.12781*, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Our abstract and introduction has clearly stated the contributions and claims we made. We have articulated the technical challenges we have addressed compared to prior work, and stated the novelty of our methods both in results end-to-end, and analyzed via careful ablations.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We have included the limitations of our current methods in the conclusion section. Overall we observe two main limitations for the current methods. First, we observe the generalization capabilities of our method is related to the training data. The real world videos are captured from different devices and have various characteristics. We observed that the generalization capability of the model can improve drastic to videos captured by the devices when there are training data collected by the same device used in training. In our experiments, we have extensively used data from Project Aria, Iphone and GoPro, which however is not representative enough for general in the wild videos. We see domain gap due to this reason and we believe it can be resolved in the future to further scale our training to videos with larger varieties. Second, we have further highlighted the limitation of our predicted 4DGS when viewed from extreme view angles. This has been a persistent challenge in monocular 4D reconstruction and remain as a challenge.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided details of our methods in implementation with the hyper-parameters. We have full disclosed our training datasets and training settings. We also provided more details in training and evaluation that cannot be sufficiently described in the main paper due to page limit.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

- (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
- (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
- (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: No

Justification: We cannot provide the code implementation at the submission time due to legal process. We will intend to release our model upon the acceptance of the paper and legal review.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We have provided details of training datasets, evaluation datasets, our implementation details and hyperparameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provided the full evaluations with error bars in standard deviation for our evaluation against baselines.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided details of our compute resources in reproducing the model in implementation details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have fully reviewed the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: We include a short discussion of social impacts in the supplementary materials. As an early research efforts, we have not foreseen our method will significantly impact the social topics at this stage.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Although we intend to release model if upon approval, we do not foresee misuse of our model in sensitive topics. Our models can only reconstruct 4D from input an user provider and will not generate or hallucinate sensitive content. Our training data also has strong compliance to privacy.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly cited all the work, including data, model and code implementation that influence this work. All datasets, models and code have public research license to be used safely.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Contents

1	Introduction	2
2	Related Work	3
3	Method	3
3.1	Feed-Forward Dynamic Gaussian Prediction	3
3.2	Multi-level Pixel & Token Density Control	5
3.3	Training	6
4	Implementation Detail	6
5	Experiments	7
6	Conclusion	10
A	Additional Details	23
A.1	Additional Details on the Multi-Level Attention Module	23
A.2	Additional Details on the Densification & Pruning of the Dynamic Gaussian	23
A.3	Additional Details on Datasets and Baselines	24
A.4	Additional Details on the Number of Gaussians	24
B	Additional Results	25
B.1	Qualitative Results of the Ablation Study	25
B.2	Results and Discussion on Pruning Pattern Selection	25
B.3	Supplementary Videos	26
C	Additional Disucssions	26
C.1	Additional Discussions on Expert Models	26
C.2	Additional Discussions on the Choice of the Dynamic Gaussian Representation . .	26
C.3	Additional Discussions on Metric Scale Cameras	27
C.4	Additional Discussions on Limitations	27
D	Social Impact	27

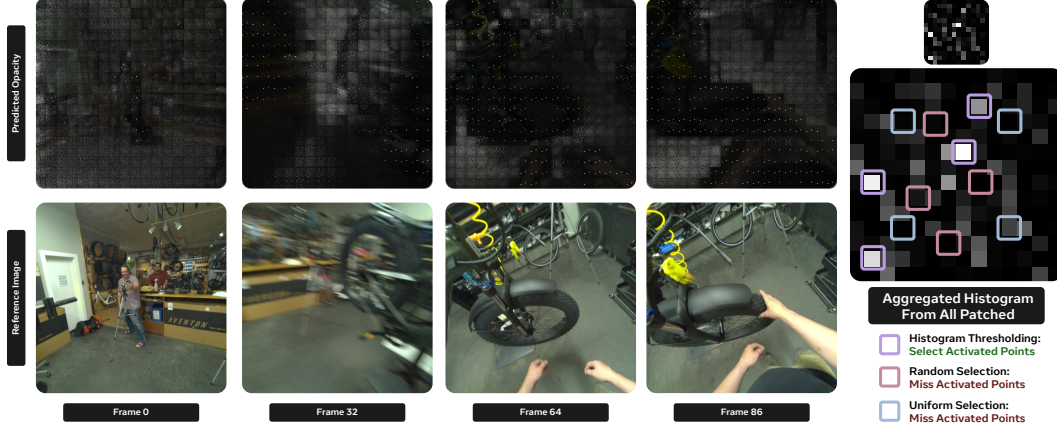


Figure 4: The predicted opacity map ($\in R^{N \times H \times W}$) of the pixel-aligned dynamic Gaussians from 4DGT and the computed histogram ($\in R^{p \times p}$) of the activation distribution. The right section shows the difference between histogram thresholding (Ours) and other filtering methods (randomly or uniformly selecting the Gaussians to keep) for reducing the number of Gaussians.

A Additional Details

A.1 Additional Details on the Multi-Level Attention Module

As mentioned in the Method section of the main paper, aside from the number of Gaussians, another efficiency-limiting factor is the number of tokens in the large number of high-resolution images. In the second stage of training, we increase the spatial and temporal resolution by a factor of R_s and R_t , respectively. The number of patches participating in the self-attention module $\mathcal{F}$ increases by a factor of $R_s^2 \cdot R_t$, which will slow down optimization and inference significantly. To mitigate this, we propose a temporal level-of-detail attention mechanism to reduce the computational cost. Specifically, noticing that the computational complexity of the self-attention module is $O(n^2)$ (simplified from $O(n^2 + n)$) where n is the number of tokens [51], We propose to divide the N input frames into M equal trunks in the highest level. This division limits the attention mechanism in the temporal dimension, but reduces the computation of calculating n total tokens to $O(\frac{n^2}{M})$. To balance spatial-temporal samples, we construct a temporal level-of-detail structure by alternating the temporal range and spatial resolution, achieving a much smaller overhead while maintaining the ability to handle long temporal windows. For each level l , we reduce the spatial resolution by a factor of 2^l and increase temporal samples by 2. This results in a computational complexity of:

$$O(\frac{n^2}{M} + \dots + \frac{n^2}{M \cdot 2^{L-1}}) = O(\frac{n^2}{M \cdot 2^{L-1}} \cdot \sum_{l=0}^{L-1} 2^l) = O(n^2 \cdot \frac{2^L - 1}{M \cdot 2^{L-1}}) \approx O(\frac{2n^2}{M}). \quad (12)$$

Empirically, we use level $L = 3$ and $M = 4$, which leads to an approximately 2 times reduction in the computational cost.

A.2 Additional Details on the Densification & Pruning of the Dynamic Gaussian

In appendix A.2, we visualize the predicted opacity map of the pixel-aligned dynamic Gaussians. It shows clear patterns of the activation of the pixel inside each patch, especially for the dynamic regions. Notably, randomly or uniformly selecting the Gaussians to keep will lead to a significant number of active Gaussians being pruned, effectively removing the ability of the model to model the dynamic parts, while our histogram thresholding scheme can effectively keep the Gaussians that are contributing. These strategies blend the densification and pruning strategies of Gaussian representations [19] and the multi-stage training strategy of ViT models [54], effectively introducing a density control scheme for the feed-forward prediction pipeline.

A.3 Additional Details on Datasets and Baselines

For each dataset used in training [28, 12, 2, 71, 29, 46, 50], we select 99.15% of the sequences as the training set and hold out the rest. For the datasets used in evaluation:

- **ADT** [34]: We select 4 subsequences for validating the reconstruction performance:
 - *Apartment_release_multiuser_cook_seq141_M1292*
 - *Apartment_release_multiskeleton_party_seq114_M1292*
 - *Apartment_release_meal_skeleton_seq135_M1292*
 - *Apartment_release_work_skeleton_seq137_M1292*
- **DyCheck** [11]: We use all 6 sequences with 3 views, and follow [53, 11] to apply the covisibility mask before computing metrics on novel views:
 - *apple, block, space-out, spin, paper-windmill, teddy*
- **TUM** [48]: We select 3 subsequences for evaluation:
 - *rgb_dataset_freiburg2_desk_with_person*
 - *rgb_dataset_freiburg3_walking_halfsphere*
 - *rgb_dataset_freiburg3_sitting_halfsphere*
- **EgoExo4D** [12]: We select 3 subsequences from the hold-out sequences:
 - *cmu_bike01_2, sfu_cooking015_2, uniandes_bouldering_003_10*
- **Nymeria** [29]: We select 2 sequences from the hold-out set:
 - *20230607_s0_james_johnson_act1_7xwm28*
 - *20230612_s1_christina_jones_act0_u2r0z8*
- **AEA** [28]: We select the *loc5_script5_seq7_rec1* sequence from the hold-out set.
- **Hot3D** [2]: We select the *P0020_ff537251* sequence from the hold-out set.

The testing sequences from EgoExo4D, AEA, and Hot3D are denoted as *Aria* in all comparisons.

Note that we do not make comparisons with CAT4D [58], BulletTimer [25] since they provide neither the source code nor the pre-trained models as of writing.

A.4 Additional Details on the Number of Gaussians

The number of dynamic Gaussians predicted by the first stage is pixel-aligned, and can be computed as (derived from eq. (4)):

$$N_g = N \times H \times W. \quad (13)$$

For a resolution of 252×252 and 16 images (half spatial resolution and $1/4 \times$ temporal resolution of the second stage), this results in 508,032 (0.5M) Gaussians.

In the second stage, the resolution is increased to 504×504 and 64 images. With the proposed patch-based pruning strategy, the number of Gaussians can be computed as (derived from eq. (4)):

$$N_g = N \times H \times W \times \frac{S}{p^2}, \quad (14)$$

which results in a total of 829,440 Gaussians.

Finally, the proposed multi-level spatial attention mechanism introduces two additional downsampled outputs with $1/4 \times$ and $1/16 \times$ the spatial resolution, respectively. This leads to a final Gaussian count of:

$$N_g = 829,440 \times \left(1 + \frac{1}{4} + \frac{1}{16}\right) = 1,088,640 \text{ (1M)}. \quad (15)$$

for the second stage.

Thanks to our proposed selective activation pruning strategy, the number of Gaussians only increases by $1 \times$ while the space-time resolution increases $15 \times$.

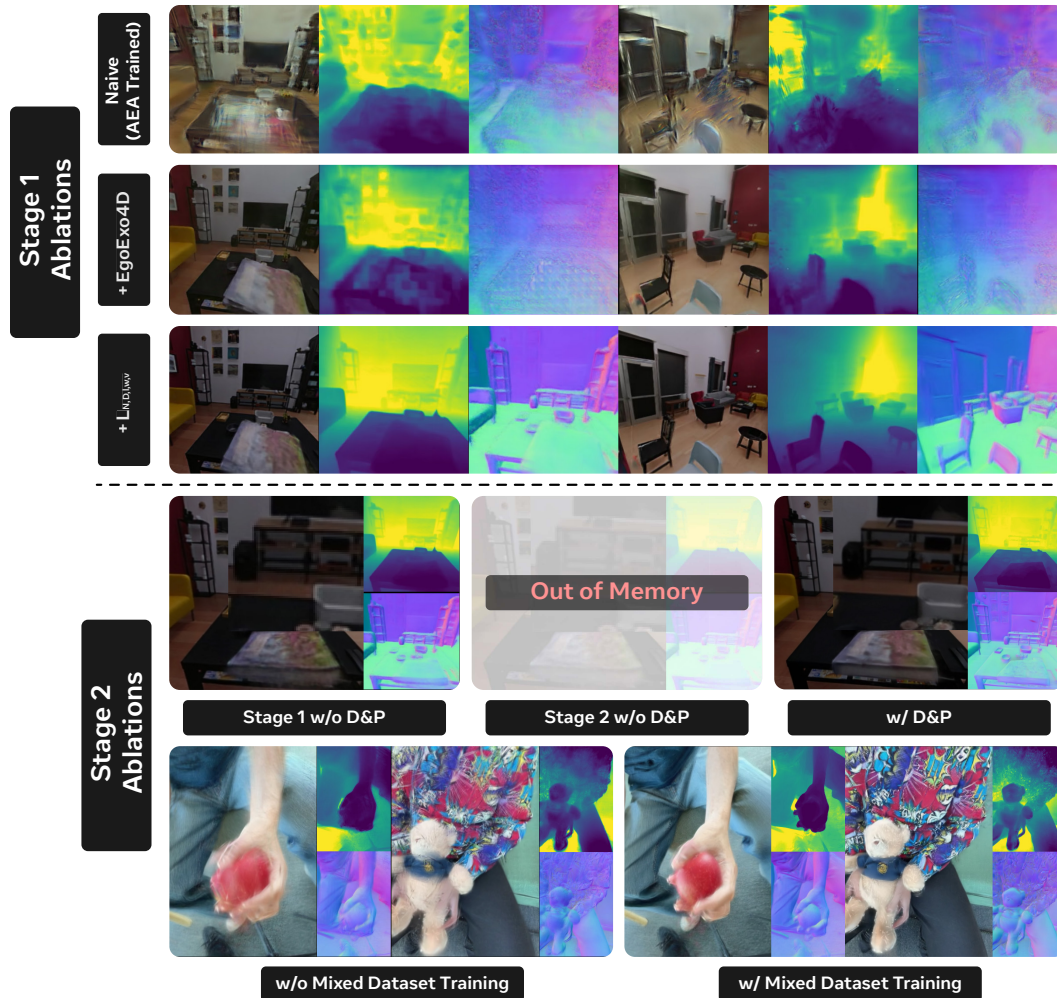


Figure 5: Ablation study on proposed components.

B Additional Results

B.1 Qualitative Results of the Ablation Study

In fig. 5, we show the ablation study results for the first and second stage training, respectively. As shown in the table and figure, our proposed loss and representation effectively model the dynamic regions and improve the reconstruction quality. Moreover, the proposed density control scheme effectively regularized the number of Gaussians with increased input count and input resolution, greatly improving details and avoiding using too much memory. Adding larger-scale datasets [12, 71] helps generalization for both in-domain and out-of-domain datasets.

B.2 Results and Discussion on Pruning Pattern Selection

In eq. (7), after computing $\mathbf{H}$ once following the first-stage training, the top S entries are selected to define a shared pruning pattern for the second-stage training. This approach effectively shares the same pattern across all patches.

The motivation for this design is twofold:

- **Empirical activation consistency:** Across the p^2 pixels within each patch, the model consistently favors similar pixels for activation and subsequent use in Gaussian rendering across all patches. This aggregation leads to a clear shared activation pattern, as visualized

in appendix A.2 (right), where the predicted opacity maps exhibit this pattern consistently. Quantitative results (table 4) show that the shared pruning pattern achieves on-par or better performance compared to recalculating the pattern for each patch on-the-fly.

- **Implementation efficiency:** Using a shared pruning pattern enables efficient implementation by discarding unused rows in the weight matrix of the final fully-connected layer of the decoder heads. This is considerably less resource-intensive regarding both memory usage and computation time, compared to dynamically sorting the opacity values for every patch during runtime.

Table 4: **Comparison of pruning strategies on the ADT [34] dataset.** “On-the-fly” computes a unique pruning pattern for each patch. “Shared” (ours) uses a single pattern for all patches.

Method	PSNR↑	LPIPS↓	RMSE↓	Degree↓	Speed Overhead
On-the-fly	28.36	0.241	0.78	25.95	Yes
Shared (Ours)	28.79	0.242	0.72	25.84	No

B.3 Supplementary Videos

We attach additional video results in the supplementary video material. The supplementary video is structured as follows:

- 00:00:00–00:00:15: Brief introduction to the input & output setting and goal of the paper.
- 00:00:15–00:00:45: Reconstruction and novel view rendering results for the depth, normal, optical flow, dynamic mask, and appearance inferred in a rolling window fashion over a long video.
- 00:00:45–00:01:05: More qualitative video results from other datasets.
- 00:01:05–00:01:35: Comparison with baseline methods StaticLRM [69], L4GM [43] and Shape-of-Motion [53].
- 00:01:35–00:02:00: Ablation study of the proposed components.

C Additional Disucssions

C.1 Additional Discussions on Expert Models

Aside from the expert normal and depth guidance, one natural way to improve the outputs’ temporal consistency is to incorporate a flow expert’s guidance [49]. It seems an optical flow expert like RAFT [49] can easily be plugged into our pipeline, but it’s non-trivial in practice. There are two reasons we did not adopt such a flow model for guidance:

- In our preliminary experiments, we discovered that the estimated optical flow exhibits strong inconsistency, often reaching more than 10 pixels in cycle consistency errors. This in turn makes the training of the 4DGT model unstable and leads to NaN values in the prediction.
- A tracking expert model like TAPIR [6] would produce much more consistent results for guiding the prediction of dynamic Gaussians, as shown by Shape-of-Motion [53]. However, the computation of such dense all-to-all tracking is extremely time-consuming (a few hours for a 128-frame clip), making it impractical for our large-scale training setup (1000 hours of video data).

Due to these reasons, we leave the addition of the flow expert model’s guidance to improve the temporal consistency to future work.

C.2 Additional Discussions on the Choice of the Dynamic Gaussian Representation

The main purpose for our choice of dynamic Gaussian representation is to enable seamless integration to a feed-forward prediction pipeline, which can be used for self-supervised training on general

dynamic videos. Compared to per-frame 3DGS, static 3DGS, flow vector field or decomposed motion bases, we found the explicit modeling of the motion terms of dynamic Gaussians (adapted from FTGS [65], originally proposed in 4DGS [19] (L124, L129)) to be a better fit for this purpose.

- Compared to a per-frame 3DGS [65] representation (denoted as the **per-frame** variant in the ablation studies of the paper), our representation enables the integration of space-time information as a 4D Gaussian with a non-zero life-span, automatically encodes information across multiple frames, making it possible to train the 4DGT model in a self-supervised manner on monocular videos. In the most extreme case with infinite life-span, the representation is reduced to a purely static 3DGS (denoted as the **Static-LRM** baseline in the paper) and would only work on static scenes. Compared to per-frame 3DGS and static 3DGS, our representation can freely encode the different levels of motion speed (from 0 to ∞) of the dynamic scene. Comparison against the **StaticLRM** baseline and **per-frame 3DGS** variant can be found in Table 2(a) of the main paper.
- Compared to a 3D flow vector field representation, like DynamicGaussians [65], our representation can be easily integrated into the pixel-aligned feed-forward prediction pipeline for patch-based vision transformers. However, their flow vector field, which is typically encoded by an MLP, would be much harder to predict in a feed-forward manner. Similar problems exist for the rigid motion representation used in Shape-of-Motion [53], since it's extremely ill-posed to accurately predict their motion bases and coefficients without complicated initialization. In comparison, our representation does not require such careful initialization. In practice, we simply set all $\mathbf{v}$, ω to zero, t to the timestamp of the corresponding frame, and l to a large value (50s).
- Compared to other implicit network-based methods like NeRF [30] (as used in Pred. 3D Repr. [41]), a Gaussian-based representation would enable much more efficient rendering and training.

C.3 Additional Discussions on Metric Scale Cameras

We empirically find the model works best when trained and inferred with metric-scale cameras due to the ambiguity in depth-scale estimation. Notably, the model doesn't rely on fully accurate scaling to perform well, as shown by experiments on the COP3D and EPIC-FIELDS datasets [46, 50], showing the ability to handle slight deviation from metric-scale calibrations. By introducing such non-metric datasets in training, we force the model to reason from the relative relation of the input cameras and the input images. However, the model would fail to predict coherent results when there exists an order-of-magnitude scale error.

C.4 Additional Discussions on Limitations

Due to the ill-posed nature of monocular reconstruction (e.g., limited viewpoint coverage and low frame rates), our method, like other monocular approaches, can still exhibit some blurriness and artifacts, especially during sudden or very fast movements and when visualizing reconstructions from challenging viewpoints. These issues are largely inherent to current monocular reconstruction paradigms. Notably, however, as also pointed out by reviewers, our approach already surpasses prior state-of-the-art baselines in terms of sharpness and artifact reduction, and demonstrates results comparable to optimization-based methods. Further improvements, such as scaling up the training datasets and introducing additional expert or supervisory signals to the 4DGT model, are promising directions for alleviating these remaining limitations.

D Social Impact

As an early-stage research on 4D reconstruction, we do not foresee any immediate social impact from this work. However, it's worth noting that such a feed-forward pipeline could be used to synthesize more convincing fake videos by introducing novel views.