Instant4D: 4D Gaussian Splatting in Minutes

Zhanpeng Luo
University of Pittsburgh
ZhanpengLuo@pitt.edu

Haoxi Ran*
Carnegie Mellon University
ranhaoxi@cmu.edu

Li Lu
Sichuan Univeristy
luli@scu.edu.cn

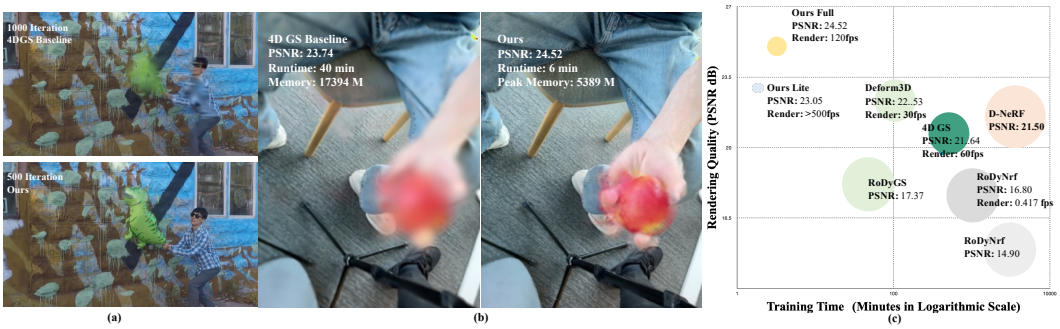

Figure 1: **Part (a):** INSTANT4D achieves better rendering performance with fewer training iterations against the original 4D Gaussian Splatting (4DGS) [37]. **Part (b):** Visualization on detailed dynamic object like a "spinning" apple. After 40-minute optimization, the rendering result of 4DGS remains blurry, while our method achieves better visual quality by 0.8 dB PSNR, faster optimization for convergence by 85%, and lower GPU memory by 69%. **Part (c):** Bubble chart comparing with most recent art. Note that the bubble size indicates the size of an optimized model.

Abstract

Dynamic view synthesis has seen significant advances, yet reconstructing scenes from uncalibrated, casual video remains challenging due to slow optimization and complex parameter estimation. In this work, we present INSTANT4D, a monocular reconstruction system that leverages native 4D representation to efficiently process casual video sequences within minutes, without calibrated cameras or depth sensors. Our method begins with geometric recovery through deep visual SLAM, followed by grid pruning to optimize scene representation. Our design significantly reduces redundancy while maintaining geometric integrity, cutting model size to under **10%** of its original footprint. To handle temporal dynamics efficiently, we introduce a streamlined 4D Gaussian representation, achieving a **30×** speed-up and reducing training time to within two minutes, while maintaining competitive performance across several benchmarks. We further apply our model to in-the-wild videos, showcasing its generalizability. Our project website will be published at `https://instant4d.github.io/Instant4D/`.

1 Introduction

Reconstructing dynamic 3D scenes from casually captured, uncalibrated video is a fundamental challenge in computer vision, critical for applications such as augmented reality (AR), virtual reality (VR), and immersive content creation. While static 3D scene modeling has seen remarkable progress [14, 1, 15], extending these techniques to dynamic scene remains challenging, especially

*Project Lead

when handling moving objects with monocular camera only. This process often requires time-consuming optimization [13, 3] to recover scene geometry and accurate motion. Furthermore, occlusion, deformation, and irregular camera paths add complexity, making efficient and coherent modeling difficult in uncalibrated settings.

Recent approaches leverage optical flow [13], depth [10], point-tracking [27], and pose prior [5] to solve this challenging task. Nevertheless, reconstructing from a short, causal video still requires hours of optimization. Inspired by recent advances in deep visual SLAM [12] and the real-time rendering [6, 37] we propose INSTANT4D, a reconstruction system for dynamic scene reconstruction in only minutes. We employ deep visual SLAM to estimate camera trajectories and refine the monocular depth into video consistent depth. These depth maps are then back-projected into a dense 3D point cloud as 4DGS optimization. Furthermore, we propose a grid pruning strategy, which efficiently reduces redundancy while preserving occlusion structures, reduces the model size to less than **10%** of its original footprint and significantly accelerates the optimization process, achieving **30×** acceleration compared with recent works of art.

Then, we model the attributes of dynamic scenes with the native 4D Gaussian primitive [36, 37] that captures motion without rigidly segmenting the scene into static and dynamic parts. Unlike approaches [13, 5, 10, 27, 8] that rigidly split the scene into static and dynamic parts, our method allows subtle intrinsic motions from the background (e.g., from the wind) to be captured naturally. However, modeling sparse and temporally inconsistent observations make the 4D Gaussian overfit and prematurely disappear in poorly observed regions. We address this through a carefully crafted initialization scheme and a motion-aware 4D covariance model.

INSTANT4D demonstrates short training time, low peak memory, fast rendering speed, and high rendering quality, as shown in Figure 1. Specifically, we reconstruct scene in NVIDIA dataset [39] in average 2 minutes and on Dycheck [3] dataset in average 7.2 minutes. We achieve 30× speed-up in reconstruction time. 90% reduction in memory and demonstrate competitive performance on several benchmarks. Our primary contributions are summarized as follows.

- We propose INSTANT4D, a modern and fully automated pipeline that reconstructs casual monocular videos in few minutes, achieving **30×** speed up.
- We introduce a grid pruning strategy that reduces the number of Gaussians by **92%**, preserving the occlusion structures and enabling scalability to long video sequences.
- We present a novel design for 4DGS in monocular setup, which achieves **29%** better than current state-of-the-art methods on the Dycheck Datast.

2 Related Work

2.1 Dynamic Novel View Synthesis (NVS)

NeRF-based NVS Earlier methods like [39] used single view and multiview stereo depth to synthesize novel views of dynamic scenes from a single video using explicit depth-based 3D warping. A recent line of work [22, 13, 18] extends NeRF [16] to handle a dynamic scene by adding a time dimension. Notably, RoDynRF [13] septate the scene into static and dynamic parts and used the static radiance field [16] to estimate the camera poses only, which were robustly reconstruct from unposed RGB video. However, limited by Neural Radiance Fields' rendering speed and numerous iteration demands, usually RoDynRF [13] takes over 2 days to reconstruct a casual video.

Gaussian-based NVS Approaches to modeling motion with Gaussian Splatting can be broadly categorized into three types: *deformation-based*, *trajectory-based*, and *4D-Gaussian-based* methods. Deformation-based methods [31, 38, 32, 7] employ multi-layer perceptrons (MLPs) or low-rank K-planes to dynamically adjust the parameters of Gaussians over time. While expressive, these methods typically suffer from slower training and rendering due to the added complexity of learning continuous deformations. Trajectory-based methods [27, 10, 26] explicitly model the motion of Gaussians by precomputing trajectories, often derived from external motion estimators. Although this enables precise tracking of object movement, it demands substantial preprocessing. For example, a 3-second video can generate trajectory files exceeding 100 GB [27]. Furthermore, these methods rigidly partition the scene into static and dynamic components, neglecting subtle background motion

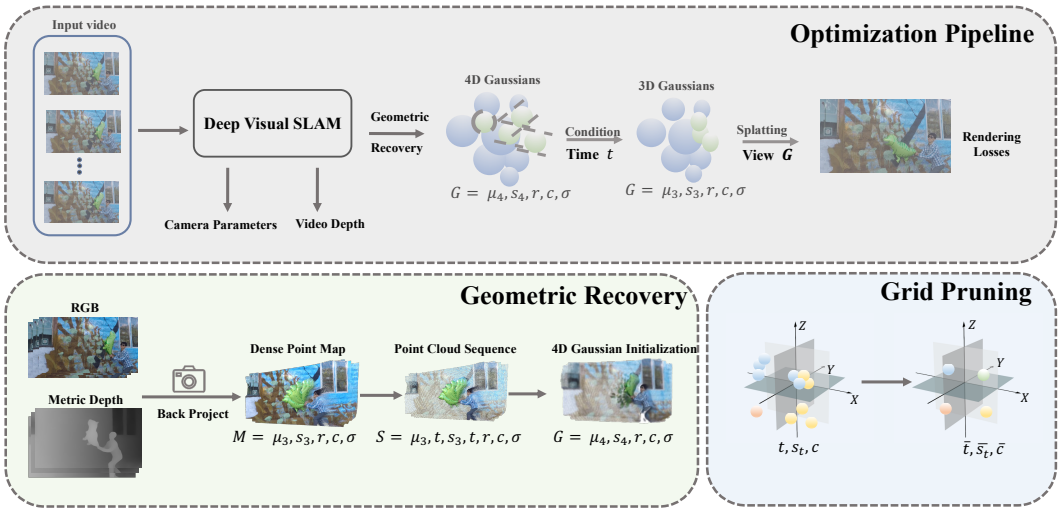

Figure 2: Pipeline of INSTANT4D. We use Deep Visual SLAM model and Unidepth [21] to obtain camera parameters, and metric depth. The metrics depth would be further optimized to consistent video depth. After that we back project from consistent depth to get dense point cloud, further voxel filtered to sparse point cloud, as discuss in Section 3.2. Based on the 4d Gaussians Initialization, we can reconstruct a scene in 2 minutes. More details about optimization are described in Section 3.3.

that may still be perceptible. This hard segmentation introduces artifacts when background elements exhibit slight temporal shifts. Finally, 4D-Gaussian-based methods [34, 37, 33] extend the standard 3D Gaussian splatting by adding a temporal dimension directly into the representation. At given timestamp, the 4D Gaussian is conditioned to a 3D distribution. However, it is prone to overfitting in monocular settings where certain regions are visible only briefly; without careful temporal management, 4D Gaussians tend to vanish prematurely as they are underconstrained in time.

2.2 Visual SLAM and SfM

Classical structure-from-motion (SfM) and Simultaneous Localization and Mapping (SLAM) pipelines recover camera poses and sparse geometry by minimizing re-projection or photometric errors through bundle adjustment [4, 23]. Deep visual SLAM systems, such as DROID-SLAM [25] replace handcrafted heuristics with differentiable bundle adjustment layers and data-driven priors, resulting in improved robustness in texture-poor scenes and mild dynamics. Recently, MegaSAM [12] has extended the differentiable bundle adjustment to dynamic scenes. MegaSAM leverages data learned before camera and flow supervision, robustly recovers camera parameters, and generates consistent video depth. Another notable recent data-driven method, DUST3R [29], powered by CroCo encoder [30], learned from vast pretrained data, reconstructs camera poses quickly and robustly. Follow efforts such as MONST3R [40] extend DUST3R onto dynamic scene with additional dynamic supervision. CUT3R [28] also apply the CroCo [30] encoder and apply continuous learning for static and dynamic reconstruction.

3 Method

Given an unconstrained video sequence $\mathcal{V} = \{I_i\}_{i=1}^N$ with resolution $H \times W$, our goal is to estimate camera extrinsics $\hat{\mathbf{G}}_i \in \mathrm{SE}(3)$, intrinsics $K \in \mathbb{R}^{3\times3}$, and temporally consistent depth maps $\hat{D} = \{\hat{D}_i\}_{i=1}^N$. Leveraging these estimates, we reconstruct a dynamic 4D scene representation and render novel views in real-time from arbitrary viewpoints $\mathbf{G}^*$ at given timestamps t^* using Gaussian Splatting. Our pipeline is illustrated in Figure 2 We first briefly summarize relevant background on deep visual SLAM and Gaussian Splatting (Section 3.1), before detailing our geometric initialization pipeline (Section 3.2) and optimization strategy (Section 3.3).

3.1 Preliminary

MegaSAM [12] extends DROID-SLAM [25]'s differentiable bundle adjustment framework to handle dynamic monocular videos. Specifically, (initialized from DepthAnything [35]), camera poses $\hat{\mathbf{G}}_i \in \mathrm{SE}(3)$, and camera intrinsics represented by focal length f (inialialized from Unidepth [21]).

During optimization, MegaSAM jointly refines these parameters by iteratively minimizing the weighted reprojection residuals between predicted optical flow and rigidly computed flow from current estimates:

$$\mathbf{u}_{ij} = \pi\left(\hat{\mathbf{G}}_{ij} \circ \pi^{-1}(\mathbf{p}_i, \hat{\mathbf{d}}_i, K^{-1}), K\right), \tag{1}$$

where $\hat{\mathbf{G}}_{ij}$ denotes the relative transformation from frame i to frame j, and $\pi(\cdot)$ denotes the camera projection operation.

MegaSAM optimizes these parameters using the Levenberg–Marquardt (LM) algorithm:

$$\left(\mathbf{J}^\top \mathbf{W} \mathbf{J} + \lambda \,\mathrm{diag}(\mathbf{J}^\top \mathbf{W} \mathbf{J})\right) \Delta = \mathbf{J}^\top \mathbf{W} \mathbf{r}, \tag{2}$$

where $\Delta = (\Delta\mathbf{G}, \Delta\mathbf{d}, \Delta f)^\top$ is the parameter update, $\mathbf{J}$ is the Jacobian of reprojection residuals $\mathbf{r}$ with respect to the parameters, and $\mathbf{W}$ is a diagonal weighting matrix derived from each frame pair. The damping factor λ is adaptively predicted by the network during each iteration to stabilize optimization.

4D Gaussian Splatting [37] extends the explicit 3D Gaussian Splatting [6] to dynamic scenes by incorporating temporal dynamics into scene modeling. Specifically, a standard 3D Gaussian is parameterized by its mean position $\boldsymbol{\mu} \in \mathbb{R}^3$, covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$, and opacity $\alpha \in \mathbb{R}$ as follows:

$$G(\mathbf{p}, \boldsymbol{\mu}, \Sigma, \alpha) = \alpha \exp\left(-\frac{1}{2}(\mathbf{p} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{p} - \boldsymbol{\mu})\right). \tag{3}$$

To represent view- and time-dependent appearance, 4DGS employs a set of 4D sphericylindrical harmonics (SCH), constructed by combining 3D spherical harmonics (SH) with temporal Fourier basis functions:

$$Z_{nl}^m(t, \theta, \phi) = \cos\left(\frac{2\pi n}{T} t\right) Y_l^m(\theta, \phi), \tag{4}$$

where Y_l^m are the standard 3D spherical harmonics indexed by degree $l \geq 0$ and order m with $-l \leq m \leq l$, n is the temporal frequency index, and T denotes the temporal period.

3.2 Geometric Recovery

Back Projection on Consistent Depth Given the input image sequence $\mathcal{V}$, we first apply MegaSAM [12] to obtain estimates for camera extrinsics $\hat{\mathbf{G}}_i \in \mathrm{SE}(3)$ and intrinsics $K \in \mathbb{R}^{3\times3}$. We then refine the initial monocular depth estimates to achieve temporally consistent depth maps $\hat{D} = \{\hat{D}_i\}_{i=1}^N$ through an additional first-order optimization. Using these refined depths, we back-project each pixel coordinate $\mathbf{p}_i$ from image space into 3D world coordinates $\mathbf{X}_i \in \mathbb{R}^3$:

$$\mathbf{X}_i = \hat{\mathbf{G}}_i \circ \pi^{-1}(\mathbf{p}_i, \hat{D}_i, K^{-1}), \tag{5}$$

where π^{-1} denotes the back-projection of pixel $\mathbf{p}_i$ (in homogeneous coordinates $\tilde{\mathbf{p}}_i$) and depth $\hat{D}_i$ into the camera frame, and $\hat{\mathbf{G}}_i$ transforms the 3D point into the world coordinate frame. This procedure yields a dense colored point cloud representing the scene geometry. To handle variations in depth scale, particularly in outdoor scenes with unbounded regions (e.g., skies), we adaptively increase the voxel size S_v during subsequent grid pruning.

Motion Probability Estimation Separating dynamic foreground objects from the static background remains beneficial, as it allows us to efficiently allocate computational resources by representing the static background sparsely while preserving detailed granularity in dynamic regions. To this end, we leverage intermediate predictions from the deep visual SLAM pipeline's low resolution motion probability map $\hat{m} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}}$. to interpolate to a per-pixel motion probabilities. We then employ Otsu's thresholding method [17] on these probability maps to generate binary masks distinguishing

static from dynamic scene elements. Empirically, we observed that in sequences with large temporal sampling intervals, motion estimation can fail to reliably identify moving objects at the sequence boundaries (i.e., first and last frames). To mitigate this issue, we introduce synthetic pseudo-frames at both ends of the sequence, thereby improving motion consistency. Additional visualizations of our motion estimation procedure are provided in the supplementary materials.

Grid Pruning Back-projecting depth maps for a 512×512 video sequence of four seconds (30 FPS) yields $\sim$30M raw 3D points. To eliminate redundancy and resolve self-occlusions, we partition the world space into a regular voxel grid and retain only the centroid of points within each occupied voxel. The edge length is adapted to the scene scale,

$$S_v = \lambda_s \cdot \frac{1}{N} \sum_{i=1}^{N} \frac{\hat{D}_i}{\hat{f}}, \tag{6}$$

where $\hat{D}_i$ is the mean depth of frame i, $\hat{f}$ the estimated focal length, N the number of frames, and λ_s a user-defined scale factor. Each 3D point is mapped to a voxel by integer division with S_v; points in the same cell are aggregated via a hash-map and replaced by their centroid, while other attributes other than position, such as color, timestamp, scale in time s_t, and motion probability $\hat{m}$ are averaged. Voxels with insufficient support are discarded as outlier to suppress noise.

On the NVIIDA Dynamic Scene benchmark [39], this pruning reduces the model's memory footprint from 10.7 GB to 0.83 GB (92%), lowers training time from 181s to 42s (4× speed-up), and boosts rendering throughput from 154 FPS to 981 FPS (see Table 2). With the resulting compact point prior we are able to bypass the densification stage conventionally used in 3D Gaussian Splatting [6].

3.3 Optimization

Motion Modeling Once 3D positions and RGB colors are acquired, we optimize the remaining Gaussian attributes such as rotation r, scaling s, opacity o, and motion m with a lightweight 4D Gaussian formulation. Each Gaussian is described by a 4D mean $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_z, \mu_t)^\top \in \mathbb{R}^4$, a diagonal scale vector $\mathbf{s} = (s_{xyz},\ s_t)^\top$, a scalar opacity α, and a rotation matrix $R \in \mathbb{R}^{4\times4}$. Unlike prior 4DGS [37] work that relies on two entangled quaternions and high-order SCH, we model the Gaussian's appearance with simple RGB value, rather than high-order spherical harmonious function. The simple design cuts the per-Gaussian parameter count by over 60 % and empirically lessen over-fitting in monocular settings.

A render-time 3D Gaussian at timestamp t is obtained by conditioning the multivariate Gaussian on the temporal dimension:

$$\boldsymbol{\mu}_{xyz|t} = \boldsymbol{\mu}_{1:3} + \Sigma_{1:3,4}\, \Sigma_{4,4}^{-1}\, (t - \mu_4), \tag{7}$$

$$\Sigma_{xyz|t} = \Sigma_{1:3,1:3} - \Sigma_{1:3,4}\, \Sigma_{4,4}^{-1}\, \Sigma_{4,1:3}, \tag{8}$$

where $\Sigma \in \mathbb{R}^{4\times4}$ is the full covariance matrix. This conditioned form encodes continuous motion without explicit trajectory storage.

Isotropic Gaussian Although anisotropic Gaussians can model fine-grained shape details, their additional degrees of freedom frequently destabilise optimisation in monocular scenarios. Inspired by Gaussian Marbles [24], we therefore adopt an *isotropic* variant: the orientation matrix is fixed to the identity ($R = I$), and the covariance is parameterized by two scalars, one shared spatial scale s_{xyz} and one temporal scale s_t. This compact parameterization improves numerical stability, reduces memory usage, and acts as an implicit regularizer. As evidenced in Table 4, the isotropic model delivers higher robustness without sacrificing rendering quality.

Motion-Aware Gaussian In a monocular 4DGS primitive modeling, static background primitives can vanish once they leave the camera frustum unless they are explicitly distinguished from moving objects. We apply the mask previously get them from 3.2.

To make our 4D primitive aware of underlying motion in the monocular dynamic scene. Considering the opacity $o_t = o \times \mathcal{N}(t, \mu_4, \Sigma_{4,4})$ and Equation 7, we can find that in the case of our isotropic

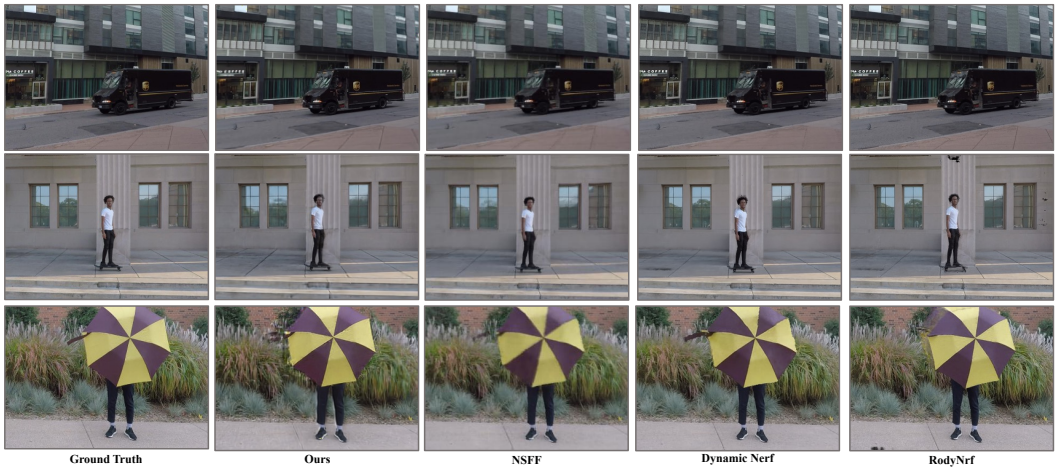

Figure 3: Visual comparison on the NVIDIA dataset. [39]

Method	Calibration	Runtime ↓	Rendering FPS ↑		PSNR ↑
			480 × 270	860 × 480	
HyperNeRF [19]	COLMAP	64	0.40	-	17.60
DynamicNeRF [3]	COLMAP	74	0.05	-	**26.10**
RoDynRF [13]	COLMAP	28	0.42	0.13	25.89
4DGS [31]	COLMAP	1.2	43	29	21.45
Casual-FVS [9]	Video-Depth-Pose	0.25	48	27	24.57
InstantSplat* [2]	Visual SLAM	0.15	117	-	22.56
4DGS* [37]	Visual SLAM	0.16	98	-	18.34
Ours	Visual SLAM	**0.02**	**822**	**676**	23.99

Table 1: Quantitative comparison of efficiency and visual quality on NVIDIA dataset following [13]. $*$: Our implementation by replacing the calibration method (COLMAP) from the original paper with Visual SLAM for fair comparison.

Gaussians, the temporal scaling s_t would be the only term in the covariance affect the Gaussian attribute related with the time.

$$\Sigma_{4,4} = s_t \times s_t \tag{9}$$

Therefore, through explicitly set s_t larger for static region, those Gaussians that should stay in the 4D space will not disappear. Those dynamic Gaussians, will change there position and scaling according to the deviation of timestamp t. During rendering, Gaussians farther away from the timestamp t will be culled if their opacity $o_t = o\mathcal{N}(t; \mu_4, \Sigma_{4,4})$ falls below a threshold.

4 Experiments

4.1 Training and Inference Detail

Implementation Detail On the Dycheck iPhone dataset [3], we followed the evaluation protocol established by Jeong et al [5]. We set the maximum optimization iterations to 5,000 and adopted the standard 3DGS[6] hyperparameters for loss weights and learning rates, with the exception of reducing the position learning rate to 1e-5 and extending the learning rate scheduler to 5,000 steps. Our initialization strategy differed between model variants. For the *Lite* model, we initialized 4D Gaussians with a voxel size of λ_s = 4 for static regions and λ_d = 4 for dynamic regions. In our *Full* model, we set λ_s = 1 but omitted the grid pruning step for dynamic regions to preserve detail and alleviate some potential underfit caused without densification. Temporal scaling was set to $s_t = \frac{2}{fps}$ for dynamic regions, while static regions used a constant scale equal to the entire video length ($s_t = l_{video}$).

Figure 4: Visual Comparison on the Dycheck dataset.[3]

Component	SH	Filter	Densification	PSNR ↑	Runtime (sec)	Memory (MB)	Rendering FPS
Ours	✓	✓		**23.99**	**42**	**832**	**981**
SCH		✓		23.83	59	2806	588
W.o. Voxel Filter			✓	23.38	181	10676	154

Table 2: Ablation study on key components of INSTANT4D's influence on speed and memory. We analyze the effect of spherical harmonics (SH), grid filtering, and Gaussian densification on rendering quality, training runtime, memory usage, and rendering frame rate (FPS). Removing higher-order SCH 3.1 slightly reduces computational cost with minimal impact on PSNR. The grid filtering significantly reduces both memory footprint and runtime while maintaining rendering quality, highlighting its role as an effective regularizer against overfitting.

For the NVIDIA Dynamic Scene dataset [39], we reduced the maximum optimization iterations to 1,500 while maintaining the same hyperparameters as our implementation in Dycheck [3], adjusting only the learning rate scheduler's maximum step to match the shorter optimization cycle. The grid pruning and initialization parameters remained consistent across both datasets.

Runtime and Memory Computational requirements for our method scale with input video length, as the SLAM system must track additional depth maps. Peak memory usage occurs during consistent video depth optimization, while 4DGS optimization maintains relatively stable runtime regardless of sequence length. Testing on a single NVIDIA A6000 GPU, our Lite model completes the full training pipeline in 96 seconds with peak memory usage of 988 MB on the shortest sequence (235-frame "paper-windmill"), and 131 seconds with peak memory of 1,147 MB on the longest sequence (379-frame "apple"). For our Full model, geometric recovery processes at approximately 0.8 seconds per frame, requiring about 5 minutes total for depth estimation, video depth consistency optimization, and camera tracking on the "apple" sequence. Inference runs at over 400 Hz, and the voxel pruning stage completes in under 5 seconds per scene.

4.2 Evaluation on NVIIDA & Dycheck Benchmarks

Evaluation on NVIIDA We evaluate INSTANT4D against several baseline methods on the NVIDIA Dynamic dataset following the [13] protocol. This dataset consists of seven scenes, each with 12 frames captured from 12 camera viewpoints for training, with testing performed from fixed viewpoints at consecutive timestamps. The visualization is shown in Figure 3.

To isolate the contributions of our approach, we developed two comparative baselines with similar training time constraints. The first **InstantSplat [2] style baseline** , adapt Fan et al.'s approach [2], with covisible global geometry initialization and joint camera pose optimization. For the prior part, we tested with counterparts against MAST3R [11] such as CUT3R [28] and MONST3R [40], but these models struggle with frequently shifting point clouds when processing a long sequence. Therefore, we still use MegaSAM [12] as the visual SLAM model. As shown in Table 1, our model achieves

PSNR(↑)	Apple	Block	Paper	Spin	Teddy	Average	Runtime(h)	Mem(GB)
D-NeRF [22]	24.23	21.80	21.85	22.15	19.46	21.50	> 24	12
RoDynRF [13]	17.38	15.99	20.71	16.66	13.28	16.80	22	15
4DGS [31]	23.24	22.05	21.03	22.99	18.89	21.64	1.2	21
Deform3D [38]	24.82	23.26	20.62	23.51	20.93	22.63	-	-
RoDynRF [13] (w.o. pose)	14.50	14.73	17.94	15.75	11.56	14.90	22	15
RoDyGS [5]	16.79	17.67	19.20	18.47	14.69	17.37	1.0	-
Ours (Lite)	24.9	23.48	23.18	23.60	19.96	23.02	**0.03**	**1.1**
Ours (Full)	**26.84**	**23.98**	**24.77**	**25.25**	**21.78**	**24.52**	0.12	8

Table 3: DyCheck iPhone benchmark [3]. Methods above the mid-rule are trained with ground-truth camera; those below operate without calibrated poses. *Runtime* denotes the mean training time per scene and *Mem* the peak GPU memory during optimization. Runtime for RoDyGS, RoDynRF, and D-NeRF is provided by the authors of [5].

a higher rendering quality while maintaining significantly faster training times, demonstrating the effectiveness of our grid pruning and initialization strategy.

Furthermore, we introduce a 4DGS [37] baseline, This baseline isolates the contribution of our 4D Gaussian representation by implementing a standard 4DGS approach without our isotropic and motion-aware Gaussian strategies. As shown in Table 1, our full model achieves superior rendering quality while maintaining significantly faster training times, demonstrating the effectiveness of our grid pruning and initialization strategy. The ablation results in Table 4 further confirm that omitting motion-aware Gaussians substantially degrades rendering quality across both datasets. This degradation likely stems from overlapping dynamic elements in world space, where improperly timestamped Gaussians occlude each other and impede optimization.

While some prior methods such as [13, 3] achieve higher PSNR values, they typically incorporate additional regularization techniques to compensate for the limited information available in the 12-frame NVIDIA dataset. Nevertheless, our method offers a compelling trade-off, delivering competitive quality with reconstruction speeds that dramatically outpace previous approaches.

Evaluation on DyCheck The DyCheck iPhone benchmark [3] presents severe motion and parallax, making it a stringent test for dynamic reconstruction. Following the RoDyGS [5] evaluation protocol, we report per-scene PSNR together with training time and peak memory (Table 3. Our *Lite* variant already surpasses all baselines methods that do not require ground truth camera pose as input, achieving an average **23.02 dB** in just **0.03 h** with a 1.1 GB footprint. The *Full* configuration lifts performance to **24.52 dB**, outperforming the concurrent RoDyGS by **7.15 dB** and exceeding Deform3D[38] (which uses ground-truth poses) by **1.89 dB**, yet still trains in only 7.2 minutes.

Both variants sustain real-time rendering (> 500 FPS), confirming that our voxel-initialized, motion-aware 4D Gaussian representation delivers state-of-the-art quality at a fraction of previous computational cost.

From the visual comparison Figure 4, we can see that compared to the baseline 4DGS model, our method preserves significantly better for static background as well as for the dynamic object.

Figure 5: Visualization on the DAVIS Dataset.

4.3 Evaluation on in-the-wild video

To assess performance on in-the-wild video, we conduct qualitative experiments on DAVIS Dataset [20]. Figure 5 shows renderings from both novel viewpoints and novel timestamps; complete video results are available on our project website. Our reconstructions exhibit crisp object boundaries and temporally coherent appearance. For example, both the *Bear* sequence (82 frames) and *Breakdance* sequence (68 frames) require 2 min for SLAM calibration and 2 min for 4D reconstruction.

We also discussed failure cases. In the low-texture *Kite-surf* sequence, the ocean dominates the field of view, leading to inaccurate visual-SLAM poses; consequently, the surfer occasionally disappears in

the rendered output. Addressing such degenerate scenarios will be our future work on texture-robust pose initialization.

4.4 Ablation and Analysis

We first experiments on the NVIDIA [39] dataset to evaluate each component's influence on the training runtime and memory as seen in the Table 2. The grid pruning significantly reduces both memory footprint and runtime while maintaining rendering quality, the simple RGB value we used show a beneficial trade-off on both performance and training speed.

To assess the impact of each design choice, we conduct an ablation study on the DyCheck iPhone dataset [3] at $2\times$ resolution (Table 4). Starting from our *Full* model, we disable individual components to evaluate their influence on rendering quality and temporal consistency. First, we replace the per-Gaussian RGB coefficient with the original time-varying 4D sphericylindrical harmonic basis as used in 4DGS. While this configuration increases the parameter count, it provides no benefit in rendering quality and in fact decreases PSNR by **1.0 dB.** This suggests that the simplified RGB representation is not only more efficient but also better suited for monocular, unconstrained video.

Component	**PSNR** ↑	**SSIM** ↑
Motion Aware Gaussian	21.11	0.721
Isotropic Gaussian	23.00	0.661
SH	23.52	0.755
Ours (Full)	**24.52**	**0.834**

Table 4: Ablation study on each component's effective on performance.

Additionally, we evaluate the impact of removing our isotropic Gaussian design. Instead of using a fixed identity orientation and a scalar for 3d scaling, this variant employs an anisotropic covariance. We find that the added flexibility introduces instability, resulting in a **1.25 dB** drop in PSNR. Finally, we investigate the effect of disabling the motion-aware Gaussian strategy, setting the temporal scale uniformly for all Gaussians. This configuration fails to differentiate between static and dynamic regions, leading to motion blur and a substantial reduction in PSNR by **3.4 dB**. The results demonstrate that each component of INSTANT4D, including the compact RGB representation, isotropic Gaussian formulation, and motion-aware temporal scaling, plays a crucial role in maintaining visual fidelity and temporal stability.

5 Discussion & Conclusion

Discussion While INSTANT4D achieves state-of-the-art efficiency and reconstruction quality, it is currently limited in its scalability to long-duration video sequences. The visual SLAM component retains depth maps for each frame, leading to a linear increase in memory consumption as the sequence length grows. This constraint hinders the application of our method to extended captures, such as multi-minute scenes or continuous video streams. Addressing this bottleneck requires innovations in hierarchical memory management and online depth-map compression, which we consider promising avenues for future research. Furthermore, handling scenes with highly reflective or transparent surfaces remains a challenge, as depth estimation becomes less stable under such conditions.

Conclusion We present INSTANT4D, a novel system for fast 4D reconstruction from casual, uncalibrated monocular video. Our approach leverages grid pruning, motion-aware Gaussian Splatting, and efficient 4D representation to achieve real-time rendering speed with limited memory overhead. Experiments on several benchmarks demonstrate its superior rendering visual quality and computational efficiency compared to existing methods. Our future work will focus on extending our framework to video sequences with arbitrary length, improving scalability through hierarchical SLAM representations and efficient memory management, while also addressing limitations in highly reflective and low-texture scenes.

References

[1] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.

[2] Zhiwen Fan, Kairun Wen, Wenyan Cong, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Sparse-view sfm-free gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309*, 2024.

[3] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Dynamic novel-view synthesis: A reality check. In *NeurIPS*, 2022.

[4] Richard Hartley. *Multiple view geometry in computer vision*, volume 665. Cambridge university press, 2003.

[5] Yoonwoo Jeong, Junmyeong Lee, Hoseung Choi, and Minsu Cho. Rodygs: Robust dynamic gaussian splatting for casual videos. *arXiv preprint arXiv:2412.03077*, 2024.

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[7] Mijeong Kim, Jongwoo Lim, and Bohyung Han. 4d gaussian splatting in the wild with uncertainty-aware regularization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 129209–129226. Curran Associates, Inc., 2024.

[8] Junoh Lee, Changyeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 5384–5409. Curran Associates, Inc., 2024.

[9] Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. Fast view synthesis of casual videos with soup-of-planes. In *European Conference on Computer Vision*, pages 278–296. Springer, 2024.

[10] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024.

[11] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[12] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv preprint arXiv:2412.04463*, 2024.

[13] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023.

[14] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[15] Andreas Meuleman, Ishaan Shah, Alexandre Lanvin, Bernhard Kerbl, and George Drettakis. On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics*, 44(4), 2025.

[16] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[17] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[18] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5865–5874, 2021.

[19] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[20] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[21] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024.

[22] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021.

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

[24] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

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

[26] Diwen Wan, Yuxiang Wang, Ruijie Lu, and Gang Zeng. Template-free articulated gaussian splatting for real-time reposable dynamic view synthesis. *arXiv preprint arXiv:2412.05570*, 2024.

[27] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.

[28] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025.

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

[30] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

[31] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024.

[32] Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 123787–123811. Curran Associates, Inc., 2024.

[33] Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 4k4d: Real-time 4d view synthesis at 4k resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20029–20040, 2024.

[34] Zhen Xu, Yinghao Xu, Zhiyuan Yu, Sida Peng, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Representing long volumetric video with temporal gaussian hierarchy. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024.

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

[36] Zeyu Yang, Zijie Pan, Xiatian Zhu, Li Zhang, Yu-Gang Jiang, and Philip HS Torr. 4d gaussian splatting: Modeling dynamic scenes with native 4d primitives. *arXiv preprint arXiv:2412.20720*, 2024.

[37] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023.

[38] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20331–20341, 2024.

[39] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020.

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

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: We made our main claim, core method and major advantage in the abstract and introduction.

   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 discuss the limitation of our work in the section 5.

   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: [Yes]

Justification: For our motion-aware, isotropic Gaussian, we give detailed assumption and proof around the equation 7.

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 listed all the necessary details to reproduce the experiment. We will also release our code if accpeted.

   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: [Yes]

Justification: The data sets are available by [39] and [3]. We will provide sufficient instructions to faithfully reproduce the main experiments.

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 provided detailed information about the implementation detail in the section 4.1.

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 report PSNR, SSIM as metric reflecting rendering quality and we also provide comparison on rendering FPS, memory usage and training time.

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 reported detailed information in the runtime and memory as in the section 4.1.

   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 faithfully observe the NeurIPS 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 talk about the application of our method in section 1 5 while there might not be direct social impacts.

    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 can 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 can 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 can enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that can arise when the technology is being used as intended and functioning correctly, harms that can 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 can 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: The paper poses no such risks.

    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 can 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: Yes, we properly cite and credit these assets.

    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: [Yes]

    Justification: We will provide details of our code and model.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The core method development in this research does not666 involve LLMs as any important, original, or non-standard components

    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.