Gaussian-Grounded Query Refinement for Feed-Forward 4D Gaussian Splatting

Jingqiao Xiu[1] **Yicong Li**[2*] **Leigang Qu**[1] **Angela Yao**[1]
[1]National University of Singapore
[2]University of Science and Technology of China



Figure 1: We propose GGQR, a feed-forward 4D Gaussian Splatting framework for reconstructing dynamic scenes from casual videos. The resulting 4D Gaussians are inherently continuous across space and time: they can be queried at any timestamp to instantiate time-specific 3D Gaussians, enabling the rendering of appearance, geometry, and motion from any viewpoint.

Abstract

Feed-forward 4D Gaussian Splatting (4DGS) offers an efficient paradigm for reconstructing dynamic scenes from monocular videos, bypassing the need for lengthy per-scene optimization. However, existing methods suffer from three fundamental bottlenecks: redundant frame-wise grid-aligned Gaussian representations, suboptimal single-step 2D-to-4D regression architectures, and restrictive dependencies on 3D/4D signals during training or inference. In response to these limitations, we introduce a novel Gaussian-Grounded Query Refinement (GGQR) framework. To reduce representation redundancy, GGQR learns a compact set of Gaussian queries equipped with a newly designed plateau-shaped temporal kernel, enabling individual primitives to model motion-consistent regions over extended temporal intervals. To alleviate regression ambiguity, these queries are iteratively refined through a Gaussian-grounded attention mechanism operating in 3D space. Specifically, GGQR uses a geometry foundation model to lift 2D observations into a shared 3D space, explicitly grounds the 3D attention in intermediate Gaussian states, and residually updates those Gaussian states. Trained in a fully self-supervised manner on casual videos, GGQR achieves state-of-the-art performance among feed-forward 4DGS methods on dynamic reconstruction benchmarks, excelling in both novel view-time rendering and motion modeling.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Feed-forward 4D Gaussian Splatting (4DGS) is a promising paradigm for reconstructing 4D Gaussians from monocular videos. By avoiding time-consuming per-scene optimization, it holds great potential for scene understanding [39, 40], immersive media [42, 43, 41], and world models [38]. To deliver such dynamic representations, the dominant practice [45, 37, 46] is to augment standard 3D Gaussians with motion attributes that govern their temporal lifespans and spatial transformations.

Existing feed-forward 4DGS frameworks [45, 37, 46], despite their advancements, still encounter fundamental bottlenecks in scene representation, network architecture, and problem formulation. **(1) Redundant Representations:** They instantiate a distinct set of 4D Gaussians for every grid cell (e.g., pixel or patch) in every frame, even for the exact same region undergoing identical motion across time. This repeated instantiation creates massive redundancy that fails to align with the underlying structural coherence of dynamic scenes. **(2) Suboptimal Architectures:** They regress 3D Gaussian motion directly from 2D observations, where actual scene dynamics are inevitably entangled with camera motion. Even with a static camera, identical 2D displacements can originate from vastly different 3D motions due to variations in depth and viewpoint. Consequently, relying on a single-step network to solve this ill-posed 2D-to-4D regression is overly burdensome and highly prone to local optima. **(3) Restrictive Formulations:** They require external conditions that are difficult to satisfy in practice, such as metric-scale camera poses as network inputs [45] or hard-to-acquire 4D annotations like scene flow and 3D point trajectories as supervision signals [46]. Together, these compromise both the real-world applicability and the training scalability of feed-forward 4DGS.

To overcome these limitations, we identify the following desiderata for our framework: **(1) Compact Representation.** An intuitive way to reduce redundancy is to decouple Gaussian prediction from per-frame spatial grids by learning a compact set of Gaussian queries, each representing a primitive capable of independently modeling a motion-consistent surface across frames. However, such sparsification is incompatible with conventional temporal opacity kernels [45], which decay rapidly away from their temporal centers and thereby prevent Gaussians from persisting over time. This necessitates a reformulation to sustain stable opacity across an extended temporal window. **(2) Grounded Architecture.** We argue that Gaussian motion prediction should be performed within a shared 3D space, where scene dynamics are decoupled from camera motion. To further mitigate the ambiguity inherent in 2D-to-4D regression, the Gaussian states represented by queries should not merely be decoded at the final output; rather, they should explicitly ground the intermediate reasoning of the network. This motivates an iterative mechanism, where each layer predicts residual updates to the current Gaussian states, decomposing an otherwise challenging regression problem into progressive Gaussian-grounded corrections. **(3) Unconstrained Formulation.** To unlock scalability and practicality, the model inputs and training pipeline must not rely on 3D/4D ground-truth annotations or pseudo labels from external models.

In light of these considerations, we propose **GGQR**, a **G**aussian-**G**rounded **Q**uery **R**efinement framework. First, we redefine the opacity gate as a differentiable plateau function, enabling each primitive to maintain a consistently high response throughout its lifespan. This formulation allows a single primitive to represent a motion-consistent region with persistent opacity over time, paving the way for a query-based compact representation. Second, we build our model upon Depth Anything 3 (DA3) [19]. Leveraging its camera and depth estimates, we lift 2D observations into a shared 3D space to serve as spatio-temporal references, effectively factoring out camera motion. Simultaneously, we sample a compact set of Gaussian queries from the 3D Gaussians predicted by DA3 and statically initialize their motion attributes. These queries undergo iterative refinement through a stack of attention layers, each of which evolves them to reference times based on their current motion hypotheses. They then gather contextual features from references selected according to their Gaussian densities, and aggregate evidence weighted by their temporal gates. This process explicitly conditions the spatio-temporal attention on the intermediate states, guiding residual updates to the queries' motion parameters and temporal windows. Finally, our framework supports self-supervised training from casual videos using rendering-based reconstruction losses. Extensive experiments on dynamic reconstruction benchmarks demonstrate that our model achieves state-of-the-art reconstruction performance among the evaluated feed-forward 4DGS methods. Beyond high-fidelity novel view-time rendering, our approach yields superior motion modeling, evidenced by significant gains in motion segmentation and cross-frame keypoint correspondence.

In summary, our contributions are threefold:

- We propose GGQR, a query-based framework for feed-forward 4DGS. Equipped with a novel plateau-shaped temporal kernel, GGQR substantially reduces the redundancy in the frame-wise, grid-aligned Gaussian representations of existing methods.
- GGQR introduces a Gaussian-grounded iterative refinement mechanism to mitigate the inherent ambiguities of single-step 2D-to-4D regression. Within this architecture, the attention module operates directly in 3D space, explicitly conditioned on the intermediate Gaussian states.
- Extensive experiments on widely adopted dynamic reconstruction benchmarks demonstrate that GGQR significantly surpasses prior works, not only in novel view-time rendering but also in downstream tasks such as motion segmentation and keypoint correspondence.

2 Related Work

Optimization-based Gaussian Reconstruction. 3D Gaussian Splatting (3DGS) [12] establishes Gaussian primitives as an efficient, explicit representation for real-time novel-view synthesis and has been widely extended to dynamic scenes. One line of research maintains a canonical set of 3D Gaussians and models temporal variations through learned deformations, typically implemented via neural networks [47, 9, 17] or feature planes [36]. A second category adopts more explicit temporal parameterizations, endowing Gaussians with temporal functions [4, 33], motion scaffolds [13], or explicit trajectories and motion bases [21, 15, 20, 28, 30]. A third approach formulates the scene using native 4D Gaussian distributions [48, 6], from which 3D Gaussians at specific timestamps are obtained by conditioning on time for rendering. Despite their representational nuances, these methods invariably require per-scene optimization, which is not only computationally prohibitive but also suffers from severe degradation under sparse views.

Feed-forward Geometry Reconstruction. While classical 3D reconstruction relies on optimization grounded in geometric principles, such as Structure-from-Motion (SfM) [25] and Multi-View Stereo (MVS) [26], recent feed-forward models instead infer geometry directly from images in a data-driven manner. DUSt3R [32] first demonstrates that Transformers can regress pointmaps from unposed, uncalibrated image pairs to jointly recover both scene and camera geometry. Subsequent models such as CUT3R [31], VGGT [29], π^3 [34], and DA3 [19] extend this paradigm to an arbitrary number of views. Concurrently, MonST3R [51] adapts this line of work to dynamic scenes, while St4RTrack [7] further couples scene reconstruction with point tracking. More recent frameworks, such as Any4D [11] and D4RT [50], unify 4D reconstruction tasks within a single architecture. However, these methods reconstruct scenes as discrete point clouds. Unlike Gaussian primitives, they lack a continuous representation for rendering with graphics pipelines.

Feed-forward Gaussian Reconstruction. Early feed-forward Gaussian reconstruction methods [2, 5, 44] target static scenes and regress 3D Gaussians from sparse posed images. With the rise of geometry foundation models [32, 14, 29], recent approaches [27, 49, 53, 10] propose to jointly infer geometry, appearance, and camera parameters from unposed images. For dynamic scenes, frameworks [16, 18] take monocular videos as input to predict discrete 3D Gaussians at specific target times. Feed-forward 4D Gaussian reconstruction advances this paradigm by yielding temporally continuous 4D Gaussians. 4DGT [45] pioneers this task for open-domain monocular videos with known camera poses, whereas subsequent methods like StreamSplat [37] and NeoVerse [46] relax the requirement for camera calibration. Nevertheless, existing pose-free methods remain largely confined to localized motion modeling between adjacent frames, whereas our approach performs holistic spatio-temporal reasoning over a longer window.

3 Methodology

As illustrated in Figure 2, our framework consists of three components. First, a frozen foundation model processes an input sequence of T frames $\mathbf{I} \in \mathbb{R}^{T\times H\times W\times 3}$ and produces four layers of backbone features $\tilde{\mathbf{F}} \in \mathbb{R}^{T\times 4\times h\times w\times C_f}$ and frame-wise, pixel-aligned 3D Gaussians $\tilde{\mathbf{G}} \in \mathbb{R}^{T\times H\times W\times C_s}$. Second, a Dense Prediction Transformer (DPT) [24] progressively upsamples and fuses the multi-scale features $\tilde{\mathbf{F}}$ into high-resolution dense features $\mathbf{F} \in \mathbb{R}^{T\times H\times W\times C_g}$. Third, a spatio-temporal Transformer initializes a compact set of Gaussian queries and iteratively refines them through an attention mechanism grounded in their intermediate Gaussian states. Here, (H, W) and (h, w) are the image and backbone-feature resolutions; C_f, C_s, and C_g are the dimensions of backbone features, Gaussian states, and dense features, respectively.

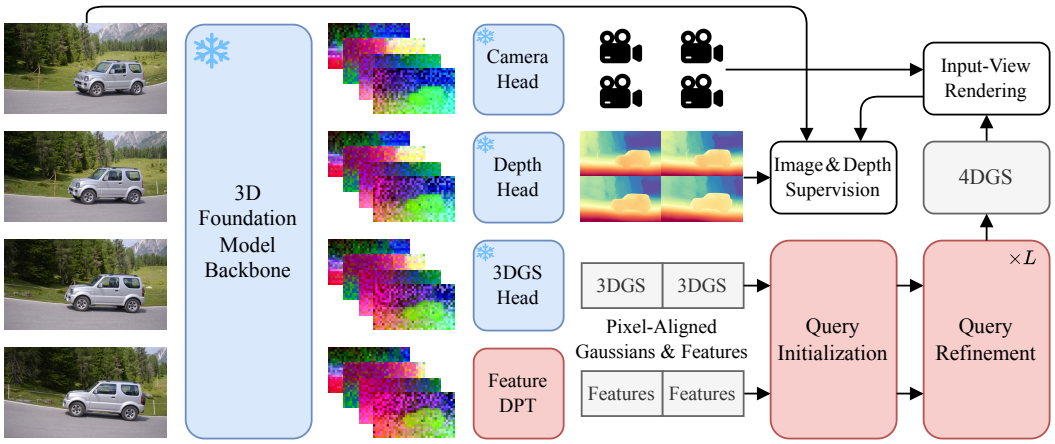

Figure 2: GGQR incorporates a frozen 3D foundation model to extract backbone features, scene geometry, and pixel-aligned 3D Gaussians from the input frames. The reference set combines the predicted scene geometry with dense features produced by DPT. A compact set of Gaussian queries is initialized from the predicted 3D Gaussians and iteratively refined into 4D Gaussians. The framework is trained using self-supervised losses applied to rendered images and depth maps.

3.1 4D Gaussian Representation

We represent a dynamic scene with a set of N 4D Gaussian primitives, denoted as $\mathbf{G} = \{g_i\}_{i=1}^{N}$. Each primitive maintains a static state $(\mu_i, s_i, q_i, c_i, \alpha_i)$, where $\mu_i \in \mathbb{R}^3$ is the spatial center, $s_i \in \mathbb{R}^3$ is the scale, $q_i \in \mathbb{R}^4$ is the rotation quaternion, $c_i \in \mathbb{R}^3$ is the color, and $\alpha_i \in \mathbb{R}$ is the opacity. To extend the primitive continuously across time, we attach a dynamic state $(\tau_i, r_i, \{v_i^d\}_{d=1}^{D_v}, \{\omega_i^d\}_{d=1}^{D_\omega})$, where τ_i and r_i are the temporal center and radius, respectively. The linear and angular motion polynomials have degrees D_v and D_ω, and their degree-d coefficients are $v_i^d \in \mathbb{R}^3$ and $\omega_i^d \in \mathbb{R}^3$.

Given a target time t, the 4D Gaussians evolve into 3D Gaussians. While the scale and appearance remain time-invariant, the position, rotation, and opacity vary continuously with time as follows:

$$\mu_i^t = \mu_i + \sum_{d=1}^{D_v} v_i^d (t-\tau_i)^d, \quad q_i^t = q_i \otimes \mathrm{Exp}\left(\sum_{d=1}^{D_\omega} \omega_i^d (t-\tau_i)^d\right), \quad \alpha_i^t = \alpha_i \,\mathrm{Rec}(t;\tau_i,r_i), \tag{1}$$

where $\otimes$ denotes quaternion multiplication, Exp is the exponential map, and Rec is a temporal gate.

Existing methods typically formulate temporal gates that decay rapidly as time deviates from their centers (e.g., standard Gaussian kernels). This inevitably leads to temporal redundancy: since each Gaussian is sharply localized around its temporal center, the model is forced to spawn numerous Gaussians with staggered temporal centers to cover a long-lived surface. To address this, we define the temporal gate as the following differentiable plateau function:

$$\mathrm{Rec}(t;\tau_i,r_i) = \sigma(\kappa(t-\tau_i+r_i)) - \sigma(\kappa(t-\tau_i-r_i)), \tag{2}$$

where σ is the sigmoid function and $\kappa = 10\ln(9)/r_i$ sets the boundary sharpness. The gate stays near 1 inside the temporal window, then decays steeply from about 0.9 at $0.1r_i$ inside either boundary to about 0.1 at $0.1r_i$ outside it. This allows a surface persisting over an extended interval to be represented by one Gaussian, eliminating the need to activate new Gaussians at neighboring times.

3.2 Grounded Query Refinement

Query Initialization. Given the predicted 3D Gaussians $\tilde{\mathbf{G}}$, we transform them into a unified world coordinate system using the estimated camera parameters. We then aggregate these world-space Gaussians across all input frames and apply Farthest Point Sampling (FPS) over their spatial centers to obtain a set of N initial Gaussian queries. Each query's static state is inherited from the sampled 3D Gaussians. For the dynamic states, the temporal center is set to the normalized frame index, the support radius is defined by the temporal interval between adjacent frames, and all motion coefficients are initialized to zero. Additionally, we associate each query with two learnable features, a static feature f_s and a motion feature f_m, both initialized from the corresponding dense

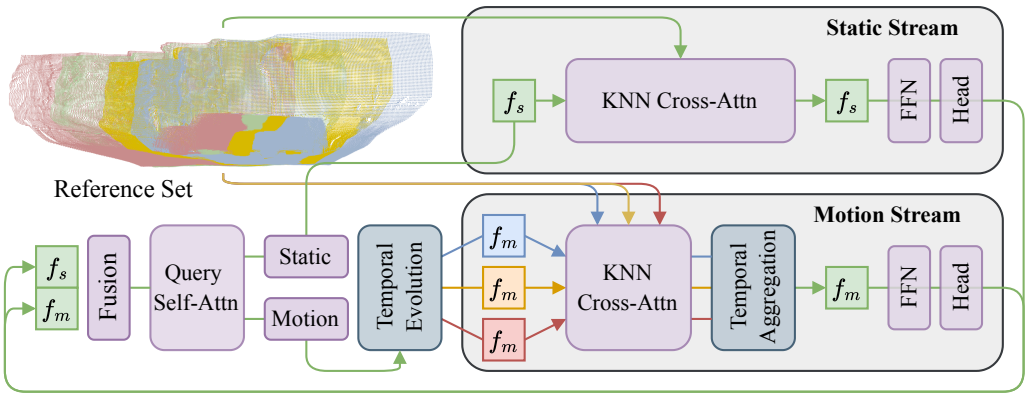

Figure 3: Each query refinement layer first performs self-attention on fused static-motion features, followed by dual-stream cross-attention: the static stream attends to same-time references, while the motion stream evolves queries over time and aggregates cross-time evidence. Residual heads update the respective Gaussian states. Queries and references with the same color indicate the same time.

features. Simultaneously, all pixel-aligned features $\mathbf{F}$ form a reference set that provides a localized representation of the scene anchored to unprojected depths and respective frames.

The initial queries and reference set are passed through a sequence of L query refinement layers. As illustrated in Figure 3, each layer consists of a joint self-attention module, followed by separate cross-attention and MLP modules dedicated to the static and motion streams.

Query Self-Attention. In each refinement layer, we first project the concatenation of static and motion features of each query into a unified feature and apply self-attention to facilitate interaction among different queries. This process is formulated as:

$$f_u = \text{Self-Attn}\left(\text{Proj}_u([f_s; f_m])\right), \tag{3}$$

$$[f_s; f_m] \leftarrow [f_s + \text{Proj}_s(f_u); f_m + \text{Proj}_m(f_u)], \tag{4}$$

where $[\cdot;\cdot]$ denotes channel-wise concatenation. To make the attention Gaussian-aware, we encode the current static and motion states into a query embedding, which is injected into the self-attention computation. Consequently, the interaction is explicitly conditioned on the spatio-temporal center, spatial extent, temporal window, appearance, and motion. This stage enhances global consistency by propagating information across spatially and temporally correlated Gaussians.

Anisotropic KNN Matching. For a Gaussian query g_i and a reference time t, we evolve the query to time t based on its current state hypotheses following Equation (1), yielding $g_i^t = (\mu_i^t, s_i, q_i^t, c_i, \alpha_i^t)$. We then select K reference features according to their Gaussian densities $\mathcal{N}(x_j^t \mid \mu_i^t, \Sigma_i^t)$, where x_j^t denotes the location of the reference feature, $\Sigma_i^t = R(q_i^t)\,\text{diag}(s_i^2)R(q_i^t)^\top$ denotes the covariance matrix, and $R(q_i^t)$ is the rotation matrix derived from the quaternion q_i^t. Unlike standard Euclidean KNN, which assumes an isotropic search space, this selection mechanism accounts for the anisotropic extent and orientation of the Gaussians. This adaptation is crucial for accurately associating features with elongated structures under large motions.

To enlarge the effective receptive field of the KNN matching, we organize the reference set $\mathbf{R}$ into a coarse-to-fine pyramid. Early layers operate on spatially subsampled references, constructed by applying FPS independently to each frame in $\mathbf{R}$. This increases the spatial reach of each neighborhood to capture large displacements. As the query is iteratively refined and moves closer to the target local region, later layers transition to denser references to resolve motion within finer-grained neighborhoods. Furthermore, we introduce a learnable *null feature* to suppress unreliable KNN matches. If the Gaussian density of a selected neighbor falls below a threshold, its feature is replaced with the null feature rather than forcing a weak correspondence. This mechanism provides an explicit 'no reliable match' fallback option and prevents contamination from distant references.

Dual-Stream Cross-Attention. Based on the above KNN matches, we perform cross-attention between the Gaussian query features and the spatio-temporal reference features. Specifically, we employ two cross-attention streams to decouple the updates of static and motion features. The static stream focuses on spatial refinement by attending strictly to reference features from the same time as the query's temporal center τ_i. In contrast, the motion stream targets temporal dynamics by attending

exclusively to reference features from times other than τ_i, using the temporally evolved Gaussian states g_i^t. In the motion stream, the keys and values are augmented with relative spatio-temporal offset embeddings to provide explicit motion cues. This allows the network to compare the current query feature against neighboring reference features and learn the underlying motion. Formally, the static and motion streams are expressed as:

$$f_s \leftarrow f_s + \text{Cross-Attn}\left(f_s, \text{KNN}(\mathbf{R}^{\tau_i})\right), \tag{5}$$

$$f_m \leftarrow f_m + \text{Agg}\left(\left\{\text{Cross-Attn}(f_m, \text{KNN}(\mathbf{R}^t)) \mid t \neq \tau_i\right\}\right), \tag{6}$$

where Agg denotes an aggregation operation over evidence from different timestamps, weighted by the temporal gate response $\text{Rec}(t; \tau_i, r_i)$ for each time t. This mechanism enables the model to emphasize features within the current support window and suppress irrelevant features from outside it. Consequently, the learned radius r_i becomes highly functional: shrinking r_i filters out distracting evidence for short-lived motions, while expanding r_i allows long-lived structures to accumulate consistent evidence over a wider interval.

Residual State Update. Finally, following the decoupled streams, the updated static and motion features are processed by their respective feed-forward networks (FFNs) before being fed into separate prediction heads to predict residual corrections to the Gaussian state of each query. The static head updates the position, scale, rotation, appearance, and opacity, while the motion head refines the temporal radius and the polynomial coefficients for linear and angular motion. These residual updates are applied in the logit space: the current physical values are first converted into logits, updated with the predicted residuals, and then mapped back to physical values via their respective activation functions. By employing zero-initialization for these heads, the refinement process is grounded in the static initialization, requiring each layer to focus solely on predicting corrections to the current state. After L refinement layers, the 4D Gaussians represented by the final set of queries constitute the reconstructed dynamic scene.

3.3 Self-Supervised Training

Our training process is fully self-supervised, eliminating the need for ground-truth annotations or external pseudo-labeling models. Given a training frame sequence $\mathbf{I}$, our model predicts camera parameters, depth maps $\mathbf{D}$, and associated confidence maps $\mathbf{C}$ for each frame. During training, we select a subset of frames as inputs, extract their backbone features and pixel-aligned 3D Gaussians, and pass them to the reconstruction modules. To supervise the 4D Gaussian representation produced at each refinement layer l, we render color images $\hat{\mathbf{I}}^l$ and depth maps $\hat{\mathbf{D}}^l$ at the estimated camera poses and normalized times of all frames in the training sequence, and optimize the following objective:

$$\mathcal{L} = \frac{1}{L}\sum_{l=1}^{L}\left(\mathcal{L}_{\text{color}}(\hat{\mathbf{I}}^l, \mathbf{I}) + \lambda_d\,\mathcal{L}_{\text{depth}}(\hat{\mathbf{D}}^l, \mathbf{D}, \mathbf{C}) + \lambda_m\,\mathcal{L}_{\text{motion}}\right), \tag{7}$$

where the hyperparameters λ_d and λ_m balance the contributions of the respective loss terms.

The color term combines mean squared error (MSE) with an LPIPS perceptual loss [52], with the latter weighted by λ_p:

$$\mathcal{L}_{\text{color}} = \text{MSE}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_p\,\text{LPIPS}(\hat{\mathbf{I}}, \mathbf{I}). \tag{8}$$

The depth term is formulated as a confidence-weighted robust regression objective:

$$\mathcal{L}_{\text{depth}} = \frac{\sum_{t,p}\mathbf{C}_{t,p}|\hat{\mathbf{D}}_{t,p} - \mathbf{D}_{t,p}|}{\sum_{t,p}\mathbf{C}_{t,p}}, \tag{9}$$

where t and p index the frames and pixels, respectively. Since the depth of sky regions is often ill-defined in monocular estimation, we use the sky mask provided by the foundation model backbone to set their confidence to zero, effectively masking out the corresponding depth loss.

While reconstruction-based supervision provides strong geometric constraints, it can lead to unnecessarily complex motion. To favor simpler motion among explanations with similar reconstruction fidelity, we impose a motion penalty on all polynomial coefficients of linear and angular motion:

$$\mathcal{L}_{\text{motion}} = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{d=1}^{D_v}\left\|v_i^d\right\|_2 + \sum_{d=1}^{D_\omega}\left\|\omega_i^d\right\|_2\right). \tag{10}$$

Table 1: Quantitative comparison with baselines on DAVIS [23] and DyCheck [8]. Protocols for motion segmentation ($\mathcal{J}$, $\mathcal{F}$) and point correspondence (PCK, EPE) are detailed in Section 4.3.

Method	DAVIS					DyCheck				
	PSNR ↑	SSIM ↑	LPIPS ↓	$\mathcal{J}$ ↑	$\mathcal{F}$ ↑	PSNR ↑	SSIM ↑	LPIPS ↓	PCK ↑	EPE ↓
4DGT [45]	16.25	0.439	0.574	24.96	14.61	18.53	0.527	0.376	0.167	0.292
NeoVerse [46]	21.43	0.595	0.155	51.74	49.92	21.43	0.596	0.161	0.701	0.058
StreamSplat [37]	23.46	0.651	0.185	N/A	N/A	21.02	0.534	0.227	0.528	0.086
Ours	**26.57**	**0.755**	**0.147**	**68.58**	**64.87**	**26.18**	**0.742**	**0.140**	**0.830**	**0.039**

4 Experiments

Training Datasets. We train our model on OpenVidHD-0.4M [22], comprising 400K high-quality, open-domain videos. Since our framework relies solely on raw videos without annotations, it is readily scalable to larger datasets (e.g., WebVid-10M [1], Panda-70M [3]). During training, we sample 2 to 8 frames from the source videos at intervals of 4 to 8 frames and use them as reconstruction inputs. We employ multi-resolution training with the longer image side fixed at 504 pixels.

Evaluation Protocols. Our evaluation is conducted on two widely adopted benchmarks: DyCheck [8], a dynamic reconstruction benchmark with held-out camera views, and DAVIS [23], an in-the-wild video dataset featuring complex motions and frequent occlusions. For the novel view-time rendering evaluation, we sample 16 frames at intervals of 3 frames as reconstruction inputs and evaluate the predicted 4DGS on the remaining 30 held-out frames (same-camera rendering on DyCheck and DAVIS). We report standard quality metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [35], and Learned Perceptual Image Patch Similarity (LPIPS) [52]. We additionally evaluate cross-camera rendering on DyCheck by reconstructing the 4DGS from the same 16 input frames used in the above same-camera evaluation and rendering all 46 frames from another held-out camera. We report co-visibility-masked PSNR (mPSNR), SSIM (mSSIM), and LPIPS (mLPIPS) for these renderings.

Compared Methods. We compare our method with scene-level feed-forward 4DGS methods: 4DGT [45], NeoVerse [46], and StreamSplat [37]. Since 4DGT requires camera parameters as input, we condition it on ground-truth parameters for DyCheck and DA3-estimated parameters for DAVIS, both of which are also used to render the output Gaussians. The other methods do not require externally supplied camera parameters for reconstruction. For NeoVerse and our method, we reconstruct Gaussians from all input frames and render novel views using the internally estimated camera poses and normalized times of the held-out frames. On the other hand, StreamSplat bypasses pose estimation by absorbing camera motion into the Gaussians. As it only processes frame pairs, we use neighboring input frames to reconstruct Gaussians for the local temporal window and render the held-out frames. To disentangle the architectural contribution from the effects of the enhanced backbone and enlarged training set, we construct a controlled baseline using the same pretrained DA3 backbone and OpenVidHD training videos. This baseline replaces our query refinement module with NeoVerse's motion head and jointly trains that head with DA3's Gaussian head.

Implementation Details. During training, the pretrained DA3 backbone and heads are kept frozen, while all other modules are optimized. We initialize the feature DPT head from the DA3 depth head, and additionally inject image features to provide rich appearance cues. Our model processes $N = 8,192$ Gaussian queries through $L = 6$ refinement layers. The reference set is constructed from $1/4$-resolution depth and feature maps. Within the KNN attention, each query selects $K = 32$ reference features according to their Gaussian densities, using a hierarchical reference set with layer-wise downsampling factors of $(8, 4, 2, 2, 1, 1)$. The orders of the linear and angular motion models are set to $D_v = D_\omega = 2$. The empirical loss weights are set to $\lambda_d = 0.1$, $\lambda_m = 1.0$, and $\lambda_p = 0.05$. Our framework is trained on 8 NVIDIA H200 GPUs for 5 days.

4.1 Main Results

Table 1 reports quantitative comparisons on DyCheck and DAVIS. Higher PSNR and SSIM indicate better reconstruction fidelity, while lower LPIPS indicates better perceptual quality. Table 2 reports cross-camera DyCheck results and efficiency, and Figure 4 provides qualitative comparisons.

Table 2: Cross-camera rendering quality on DyCheck [8], evaluated with co-visibility-masked metrics, and efficiency metrics for a 16-frame sequence at 360×480 on a single NVIDIA H100 GPU.

Method	mPSNR↑	mSSIM↑	mLPIPS↓	Recon. Time↓	Peak Memory↓	Render FPS↑
4DGT [45]	10.09	0.418	0.714	1.41 s	7.42 GB	197
NeoVerse [46]	10.11	0.384	0.572	1.95 s	10.2 GB	231
StreamSplat [37]	11.28	0.404	0.679	**0.74 s**	**3.04 GB**	642
Ours	**11.72**	**0.457**	**0.520**	2.00 s	16.7 GB	**1248**



Figure 4: Qualitative comparison with baselines on DAVIS [23] and DyCheck [8]. For each scenario, we present two input frames and two novel view-time renderings from the reconstructed 4DGS.

Quantitative Results. Our method consistently outperforms all baselines by a significant margin across all metrics on both datasets in Table 1. Baseline methods that directly regress 4D Gaussians from 2D features struggle to faithfully capture motion attributes. Consequently, they introduce severe artifacts in dynamic regions, which substantially degrade image quality and lead to lower PSNR and SSIM scores. The quality advantage persists when the backbone and training data are matched: on DAVIS, the controlled baseline achieves 19.94 PSNR, 0.492 SSIM, and 0.229 LPIPS versus our method's 26.57 PSNR, 0.755 SSIM, and 0.147 LPIPS. This substantial gap underscores the essential role of our architecture. Beyond superior visual fidelity, our 4DGS representation is also highly compact: each video scene is represented by a total of only 8K Gaussian primitives across all frames, a mere fraction of the 100K+ primitives per frame produced by baselines.

Efficiency Metrics. On a single NVIDIA H100 GPU, we measure inference time and peak GPU memory when reconstructing a 4DGS from a 16-frame sequence at 360×480 resolution. After reconstruction, we measure rendering throughput in frames per second (FPS) at the same resolution. As shown in Table 2, our method achieves substantially higher rendering throughput than the evaluated feed-forward 4DGS methods by reducing primitive redundancy in its reconstructed 4DGS.

Qualitative Results. Visual comparisons clearly illustrate the superiority of our approach. In dynamic regions, particularly under concurrent camera motion, baselines frequently suffer from severe blurriness, ghosting effects, and geometric distortion. Conversely, our model robustly mitigates camera interference, preserving sharp boundaries, fine-grained textures, and coherent structures throughout the sequence. These visual improvements validate the efficacy of our compact 4D Gaussian representation in suppressing floaters and redundancy-induced artifacts, as well as our geometry-grounded iterative refinement in reconstructing challenging scenes.

4.2 Ablation Study

We conduct ablation studies to validate the key designs in Section 3. As shown in Table 3, removing any component leads to a noticeable degradation in rendering quality.

Effect of the Plateau Temporal Gate. First, we replace the plateau temporal gate with a standard Gaussian kernel ("w/o PlatGate"). As visualized in Figure 6, the Gaussian gate used by 4DGT [45]

Table 3: Ablation study of three key designs, including plateau temporal gate (PlatGate), anisotropic KNN matching (AniKNN), and dual-stream features (DualFeat), on DAVIS [23] and DyCheck [8].

Variant	DAVIS			DyCheck		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
w/o PlatGate	24.71	0.686	0.163	24.16	0.672	0.155
w/o AniKNN	25.13	0.675	0.169	24.61	0.660	0.161
w/o DualFeat	22.90	0.617	0.186	22.19	0.601	0.177
Full	**26.57**	**0.755**	**0.147**	**26.18**	**0.742**	**0.140**

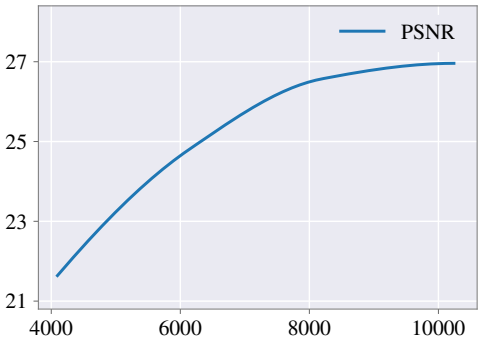

Figure 5: Reconstruction quality (PSNR) versus the number of Gaussian queries on DAVIS [23].

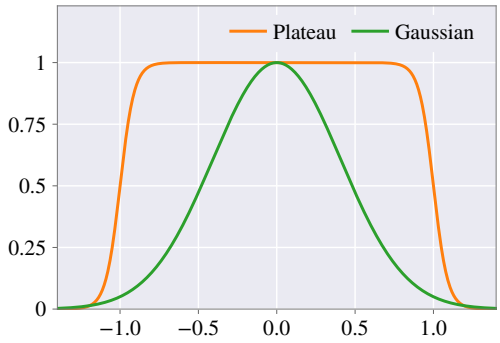

Figure 6: Response versus the normalized time offset $(t-\tau_i)/r_i$ for plateau and Gaussian gates.

decays rapidly away from its temporal center, whereas our plateau gate maintains near-full activation over most of its support interval and decays only near the boundaries. This sustained activation keeps persistent surfaces visible without requiring redundant primitives to compensate for opacity decay.

Effect of the Anisotropic KNN Matching. Second, we replace the anisotropic KNN with a standard Euclidean KNN ("w/o AniKNN"). Relying solely on centroid distances, standard KNN frequently selects spatially nearby reference points that fall outside the anisotropic extent of the temporally evolved query. In contrast, our anisotropic KNN incorporates the Gaussian's scale, rotation, and angular motion into the matching process. This formulation provides a cleaner spatial context and more coherent temporal cues for better-conditioned refinement.

Effect of the Dual-Stream Features. Third, we collapse the dual-stream design into a single feature stream ("w/o DualFeat"). This variant encourages the model to over-rely on strong same-time features, neglecting the cross-time correspondences essential for motion. Our dual-stream design circumvents this shortcut: the static stream focuses on the same-time evidence for geometry and appearance refinement, while the motion stream is explicitly forced to aggregate cross-time evidence for motion estimation. This architectural disentanglement facilitates the learning of Gaussian dynamics.

Effect of the Gaussian Query Count. We vary the number of Gaussian queries on DAVIS to examine the trade-off between representation compactness and reconstruction fidelity. As shown in Figure 5, using only 4K queries degrades reconstruction quality, suggesting insufficient capacity to represent fine structures and textures. Increasing the count to 10K yields modest gains in reconstruction quality over our default 8K, which balances compactness and fidelity.

4.3 In-Depth Analysis

We explore emergent properties of the reconstructed 4DGS, specifically motion segmentation and point correspondence, with results detailed in Table 1 and Figure 7. Notably, StreamSplat is unsuitable for motion segmentation, as its Gaussian motion entangles scene dynamics with camera movement.

Motion Segmentation. To obtain motion masks, we assign a binary motion label to each Gaussian by thresholding the magnitude of its first-order motion coefficients at 0.001. This binary attribute is then splatted onto the image plane via standard rasterization, yielding a soft dynamic occupancy map. Each pixel in this map quantifies the accumulated contribution of dynamic Gaussians, allowing us

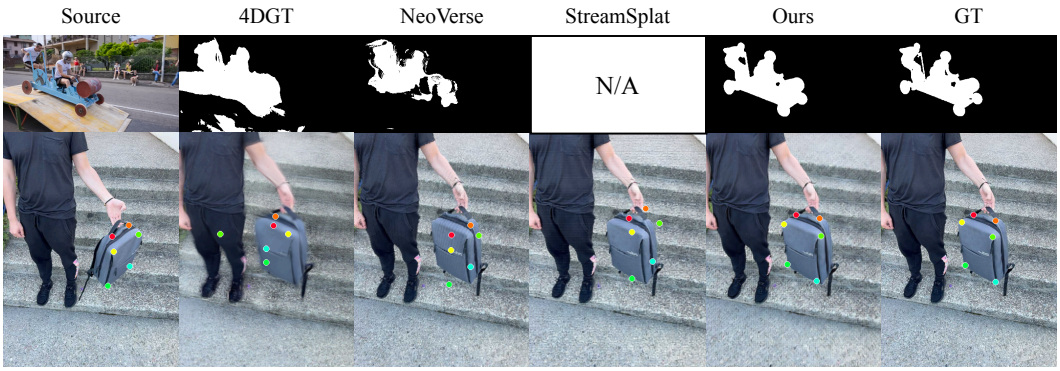

Figure 7: Qualitative comparison with baselines. The top row shows motion segmentation masks on DAVIS [23], and the bottom row shows keypoint correspondences across frames on DyCheck [8].

to derive a binary segmentation mask via thresholding at 0.5. These predicted masks on the DAVIS dataset are evaluated against ground-truth foreground annotations. Following standard evaluation protocols, we report the average region similarity $\mathcal{J}_{\text{mean}}$ (Jaccard Index) and average contour accuracy $\mathcal{F}_{\text{mean}}$ (F-measure). These strong results demonstrate that our reconstructed 4D Gaussians accurately separate dynamic objects from the static background.

Point Correspondence. To extract point correspondences between source and target frames, we leverage the explicitly modeled motion of the reconstructed 4D Gaussians. Specifically, for the 3D Gaussians evaluated at the source time, we rasterize their motion-warped 3D centers, which are evolved to the target time and transformed into the target camera space, onto the source image plane. This yields a coordinate map that assigns each source pixel a corresponding 3D position in the target frame. We then sample this map at each source keypoint location and project the sampled 3D point onto the 2D image plane to obtain the predicted target keypoint location. Due to the temporally sparse keypoint annotations in the DyCheck dataset, we evaluate correspondences between adjacent annotated frames and restrict the analysis to co-visible keypoints. Under this evaluation protocol, we report the Percentage of Correct Keypoints (PCK) at a threshold of $0.05 \max(H, W)$, as well as the 2D End-Point Error (EPE) in normalized image coordinates. These strong results demonstrate that our reconstructed 4D Gaussians effectively capture the scene dynamics.

4.4 Limitations

Our framework still has several limitations. First, our model is built on top of a frozen DA3 model [19]. When the estimated cameras, depths, or 3D Gaussians are inaccurate, the lifted reference set and initial Gaussian queries in world space suffer from noise, which can subsequently destabilize the downstream iterative query refinement. In practice, failure cases are more likely to occur in scenes exhibiting heavy motion blur, severe occlusions, non-Lambertian (reflective or transparent) surfaces, or drastic topological changes. Second, our model faces challenges when scaling to lengthy, high-resolution videos featuring crowded, dynamic scenes with numerous interacting objects.

5 Conclusion

In this paper, we introduce GGQR, a novel Gaussian-grounded query refinement framework that addresses the representational, architectural, and formulation bottlenecks of existing feed-forward 4DGS approaches. Guided by the goals of compact representation, grounded architecture, and unconstrained formulation, GGQR employs a compact set of Gaussian queries equipped with a plateau-shaped temporal kernel to reduce redundancy among frame-wise pixel-aligned Gaussians. It also leverages an iterative Gaussian-grounded refinement mechanism within a shared 3D space estimated by a 3D foundation model to mitigate ambiguities in single-step 2D-to-4D regression. Crucially, GGQR operates without external 3D/4D conditions and requires only casual videos for self-supervised training. Extensive experiments demonstrate that GGQR significantly outperforms prior feed-forward 4DGS methods in novel view-time rendering and motion modeling. We hope this work will inspire further exploration of feed-forward 4DGS and serve as a solid foundation for efficient, accurate, and scalable dynamic scene reconstruction.

Acknowledgments and Disclosure of Funding

We would like to acknowledge that computational work involved in this research is partially supported by NUS IT's Research Computing group using grant number NUSREC-HPC-00001.

References

[1] M. Bain, A. Nagrani, G. Varol, and A. Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[2] D. Charatan, S. L. Li, A. Tagliasacchi, and V. Sitzmann. Pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024.

[3] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024.

[4] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *International Journal of Computer Vision*, 134(3):83, 2026.

[5] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386, 2024.

[6] 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 Conference*, pages 1–11, 2024.

[7] H. Feng, J. Zhang, Q. Wang, Y. Ye, P. Yu, M. J. Black, T. Darrell, and A. Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[8] H. Gao, R. Li, S. Tulsiani, B. Russell, and A. Kanazawa. Monocular dynamic view synthesis: A reality check. In *Conference on Neural Information Processing Systems*, pages 33768–33780, 2022.

[9] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024.

[10] L. Jiang, Y. Mao, L. Xu, T. Lu, K. Ren, Y. Jin, X. Xu, M. Yu, J. Pang, F. Zhao, D. Lin, and B. Dai. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025.

[11] J. Karhade, N. Keetha, Y. Zhang, T. Gupta, A. Sharma, S. Scherer, and D. Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[12] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[13] J. Lei, Y. Weng, A. W. Harley, L. Guibas, and K. Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6177, 2025.

[14] V. Leroy, Y. Cabon, and J. Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91, 2024.

[15] Z. Li, Z. Chen, Z. Li, and Y. Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024.

[16] H. Liang, J. Ren, A. Mirzaei, A. Torralba, Z. Liu, I. Gilitschenski, S. Fidler, C. Oztireli, H. Ling, Z. Gojcic, and J. Huang. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. In *Conference on Neural Information Processing Systems*, 2025.

[17] Y. Liang, N. Khan, Z. Li, T. H. Nguyen-Phuoc, D. Lanman, J. Tompkin, and L. Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2642–2652, 2025.

[18] C. Lin, Y. Lin, P. Pan, Y. Yu, T. Hu, H. Yan, K. Fragkiadaki, and Y. Mu. Movies: Motion-aware 4d dynamic view synthesis in one second. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[19] H. Lin, S. Chen, J. H. Liew, D. Y. Chen, Z. Li, Y. Zhao, S. Peng, H. Guo, X. Zhou, G. Shi, J. Feng, and B. Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026.

[20] Y. Lin, Z. Dai, S. Zhu, and Y. Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024.

[21] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision*, pages 800–809, 2024.

[22] K. Nan, R. Xie, P. Zhou, T. Fan, Z. Yang, Z. Chen, X. Li, J. Yang, and Y. Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations*, 2025.

[23] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[24] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[25] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[26] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2016.

[27] B. Smart, C. Zheng, I. Laina, and V. A. Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

[28] C. Stearns, A. Harley, M. Uy, F. Dubost, F. Tombari, G. Wetzstein, and L. Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *ACM SIGGRAPH Asia Conference*, pages 1–11, 2024.

[29] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5294–5306, 2025.

[30] Q. Wang, V. Ye, H. Gao, W. Zeng, J. Austin, Z. Li, and A. Kanazawa. Shape of motion: 4d reconstruction from a single video. In *IEEE/CVF International Conference on Computer Vision*, pages 9660–9672, 2025.

[31] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[32] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[33] Y. Wang, P. Yang, Z. Xu, J. Sun, Z. Zhang, Y. Chen, H. Bao, S. Peng, and X. Zhou. Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21750–21760, 2025.

[34] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[36] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

[37] Z. Wu, Q. Yan, X. Yi, L. Wang, and R. Liao. Streamsplat: Towards online dynamic 3d reconstruction from uncalibrated video streams. In *International Conference on Learning Representations*, 2026.

[38] J. Xiu, F. Hong, Y. Li, M. Li, W. Wang, S. Han, L. Pan, and Z. Liu. Egotwin: Dreaming body and view in first person. In *International Conference on Learning Representations*, 2026.

[39] J. Xiu, M. Li, W. Ji, J. Chen, H. Zhao, S. Satoh, and R. Zimmermann. Hierarchical debiasing and noisy correction for cross-domain video tube retrieval. In *ACM International Conference on Multimedia*, pages 9271–9280, 2024.

[40] J. Xiu, M. Li, Z. Yang, W. Ji, Y. Yin, and R. Zimmermann. Few-shot incremental learning via foreground aggregation and knowledge transfer for audio-visual semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pages 8788–8796, 2025.

[41] J. Xiu, Y. Li, and A. Yao. Optimizing mesh animation from video via shape flow guidance. In *European Conference on Computer Vision*, pages 190–208, 2026.

[42] J. Xiu, Y. Li, N. Zhao, H. Fang, X. Wang, and A. Yao. Geometric alignment and prior modulation for view-guided point cloud completion on unseen categories. In *IEEE/CVF International Conference on Computer Vision*, pages 27435–27444, 2025.

[43] J. Xiu, C. Wang, and D. Xu. Mllmsplat: A 2d mllm-powered framework for 3d gaussian splatting understanding, generation, and editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 33301–33311, 2026.

[44] H. Xu, S. Peng, F. Wang, H. Blum, D. Barath, A. Geiger, and M. Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16453–16463, 2025.

[45] Z. Xu, Z. Li, Z. Dong, X. Zhou, R. Newcombe, and Z. Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. In *Conference on Neural Information Processing Systems*, 2025.

[46] Y. Yang, L. Fan, Z. Shi, J. Peng, F. Wang, and Z. Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[47] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.

[48] 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*, 2024.

[49] B. Ye, S. Liu, H. Xu, X. Li, M. Pollefeys, M.-H. Yang, and S. Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations*, 2025.

[50] C. Zhang, G. Le Moing, S. Koppula, I. Rocco, L. Momeni, J. Xie, S. Sun, R. Sukthankar, J. K. Barral, R. Hadsell, Z. Ghahramani, A. Zisserman, J. Zhang, and M. S. M. Sajjadi. Efficiently reconstructing dynamic scenes one d4rt at a time. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[51] 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. In *International Conference on Learning Representations*, 2025.

[52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[53] S. Zhang, J. Wang, Y. Xu, N. Xue, C. Rupprecht, X. Zhou, Y. Shen, and G. Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21936–21947, 2025.

A More Baselines

We additionally compare our method with two optimization-based 4DGS methods for casual monocular videos, Shape of Motion (SoM) [30] and MoSca [13]. As shown in Tables 4 and 5, our method achieves the highest PSNR and SSIM on DAVIS and same-camera DyCheck and the highest mPSNR and mSSIM on cross-camera DyCheck, while MoSca and SoM obtain lower LPIPS and mLPIPS, respectively. Compared with these methods, which optimize each scene individually, our method reconstructs scenes substantially faster through feed-forward inference and achieves higher rendering throughput, at the cost of increased peak GPU memory.

Table 4: Comparison with optimization-based 4DGS methods on DAVIS and same-camera DyCheck.

Method	DAVIS			DyCheck		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SoM [30]	19.10	0.466	0.245	20.31	0.536	0.160
MoSca [13]	24.48	0.741	**0.115**	21.05	0.558	**0.109**
Ours	**26.57**	**0.755**	0.147	**26.18**	**0.742**	0.140

Table 5: Cross-camera DyCheck and efficiency comparison with optimization-based 4DGS methods.

Method	mPSNR↑	mSSIM↑	mLPIPS↓	Recon. Time↓	Peak Memory↓	Render FPS↑
SoM [30]	11.69	0.443	**0.511**	71.3 s	**2.33 GB**	403
MoSca [13]	11.53	0.431	0.527	751 s	3.98 GB	156
Ours	**11.72**	**0.457**	0.520	**2.00 s**	16.7 GB	**1248**

B Discussion

Although a single Gaussian with a second-degree polynomial motion model may be insufficient to capture complex motion over the full reconstruction window, our method can approximate such motion using multiple Gaussians. This is because complex motion can be approximated by simpler local motions over shorter temporal intervals, with different Gaussians modeling different intervals. During query refinement, each Gaussian is evolved to other frames according to its current motion model and matched against cross-time reference features. When the motion changes beyond what this model can capture, the evolved Gaussian may no longer match the reference features consistently. This mismatch provides a cue to reduce the Gaussian's temporal radius, suppressing its contribution at times when its motion model is inaccurate. Gaussians with temporal centers closer to those times can model the corresponding motion locally.

C Broader Impacts

This work advances the efficiency of dynamic scene reconstruction from casual monocular videos, offering significant potential for applications in AR/VR, robotic simulation, and content creation. By eliminating the requirement for per-scene optimization, our method substantially lowers the computational barriers to converting real-world video into high-fidelity 4D representations, thereby democratizing 4D synthesis tools for a broader range of researchers and creators. However, we recognize that such capabilities also carry risks of misuse. The ability to reconstruct and re-render dynamic scenes from ordinary video could be exploited to synthesize footage of individuals or private environments without explicit consent. Such unauthorized manipulation poses ethical challenges regarding privacy and digital authenticity. We encourage the development of digital forensic techniques alongside these tools to mitigate the risks of unauthorized re-rendering.

D Licenses

We use third-party assets in compliance with their respective licenses. DAVIS [23] and OpenVidHD-0.4M [22] are licensed under CC BY 4.0, while DyCheck [8] is licensed under Apache 2.0. Depth Anything 3 [19] uses Apache 2.0 for its code and CC BY-NC 4.0 for its checkpoints.

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: The claims made in the abstract and introduction align with the contributions and scope of this paper.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The limitations of this work are discussed in Section 4.4.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] 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: This paper provides all necessary details to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] 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 datasets used in this work are publicly available, and the source code is available at `https://huggingface.co/JingqiaoXiu/GGQR`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper provides the training and test details in the experiments section.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: Following community practices for the evaluated benchmarks, this paper does not report error bars or other statistical significance measures.

   Guidelines:

   - The answer [N/A] 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: This paper provides information on the computer resources used for all experiments in the experiments section.

   Guidelines:

   - The answer [N/A] 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: The research in this paper is conducted in full accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [Yes]

    Justification: This paper discusses both potential positive and negative societal impacts in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not introduce datasets or models with a high risk of misuse.

    Guidelines:

    - The answer [N/A] 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: This paper credits the original creators of all used assets and lists their respective licenses in the appendix.

    Guidelines:

    - The answer [N/A] 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: The released assets are accompanied by usage instructions at `https://huggingface.co/JingqiaoXiu/GGQR`.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing, editing, or formatting; they were not part of the research methodology.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.