ONE QUERY, REAL MOTION: LOCAL-FLOW DISTILLATION FOR DYNAMIC GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic 3D Gaussian Splatting methods usually represents a scene as canonical Gaussians deformed by a time-conditioned network. Such direct-query models enable real-time rendering, but formulate motion as a timestamp-to-state mapping rather than a state-to-state transition cause fast or complex motion is poorly constrained, which often produces motion blur and loss of fine spatial detail. To address this issue, we decompose motion into an exactly composable low-frequency positional semigroup and a state-dependent local flow that captures high-frequency residual dynamics. The local flow is learned through training-time rollouts supervised on keyframes and intermediate frames, and a transition-consistency objective distills the learned rollout dynamics into the direct decoder. At inference, only the direct decoder is used, so rendering remains a single-query operation, while the learned temporal dynamics improve alignment and detail. Experiments on HyperNeRF and N3DV demonstrate consistent improvements over baselines.

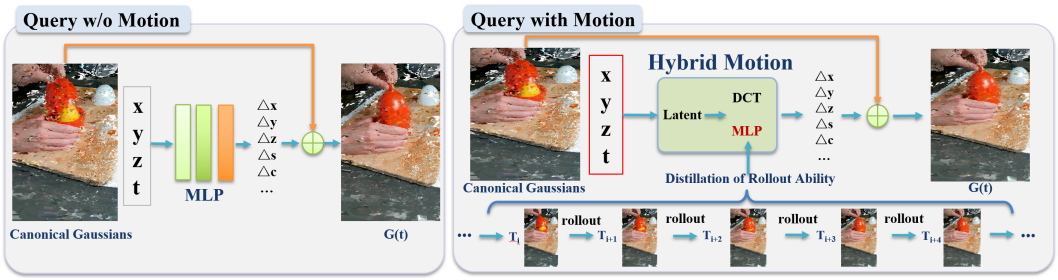

Figure 1: **Comparison between our method and other methods.** Compared to the methods based on deformation field, we decompose motion into DCT and MLP parts, and then use MLP to distill and learn local flow transfer to make it have real motion modeling ability in one query, which is not available in the query w/o motion method.

1 INTRODUCTION

Explicit 3D Gaussian Splatting (3DGS) achieves both real-time performance and high fidelity for static novel-view synthesis via anisotropic Gaussian primitives and differentiable rasterization (Kerbl et al., 2023). Nevertheless, its representation shows lack of the temporal dimension and cannot inherently characterize deformation, occlusions, and object appearance and disappearance. Dynamic radiance-field methods put forward an initial solution to temporal modeling. Represented by D-NeRF, 4D-NeRF dynamic approaches maintain temporal consistency through a canonical radiance field paired with a time-dependent deformation field (Pumarola et al., 2021), yet suffer from the inefficiency of ray-by-ray volume rendering. Dynamic Gaussian Splatting migrates this paradigm to explicit Gaussian primitives, giving rise to two major research branches. The first branch maintains Gaussians within a continuous spacetime domain, for example, 4DGS (Wu et al., 2024) and 4DRotorGS (Duan et al., 2024). The second branch fixes Gaussians in canonical space and maps them to individual observation timestamps via deformation fields, with Deformable-GS (Yang et al., 2024b) as a representative work. Subsequent advances along both lines continue to improve representation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

stability (Lu et al., 2024; Xu et al., 2024; Liu et al., 2025; Yan et al., 2024; Wang et al., 2025; Oh et al., 2025; Shi et al., 2026; Jiao et al., 2026b; He et al., 2026), motion parameterization (Huang et al., 2024; Wan et al., 2024; Yu et al., 2024; Park et al., 2025; Hou et al., 2025; Jiao et al., 2026a; Kwak et al., 2026; Lee et al., 2026; Yeom et al., 2026; Kim et al., 2026), and dynamic-perspective modeling (Qin et al., 2025; Asiimwe & Vondrick, 2025; Wang et al., 2026; Quan et al., 2026; Deng et al., 2025; Hong et al., 2026; Sato et al., 2026). These methods are capable of handling challenging scenarios including large movements, complex topologies, monocular / sparse inputs, and long sequences.

Current dynamic Gaussian Splatting methods follow a unified inference paradigm: a single direct decoding at the query time produces the deformation or Gaussian attributes at that time. Though efficient, this conventional formulation models motion as a pure time-to-state mapping, without embedding explicit state-to-state transition operator.This fundamental design omission leads to two overlooked drawbacks. First, it lacks temporal composability: a direct query returns isolated states without explicit relations between queries, so long sequences tend to be fit segment by segment, and fast local deformations appear as motion blur and loss of detail. Second, the learning signal degenerates. If a transition is defined as the difference between two direct queries, multi-step rollout is algebraically equivalent to one-step querying and provides no new dynamical information. Furthermore, If rollout and direct query share the same photometric path, supervision at intermediate frames is absorbed by direct decoding, and the state-dependent local flow receives no independent observation.

To address this, we propose a dynamic Gaussian representation centered on transition distillation, which decouples direct querying and state transition into two complementary models: a state-free Direct model for test-time deployment, and a state-dependent Rollout model that learns the local dynamics. We split motion into two parts: a DCT-based temporal evolution defined on a time-augmented state space, whose evolution operators form a commutative monoid under function composition and model only low-frequency, exactly composable positional motion, and a nonlinear local flow conditioned on a shared Triplane field that models local positional residuals and changes in scale, rotation, opacity, and color. The two paths share the DCT positional basis and the Triplane condition but use independent nonlinear networks, preventing distillation from degenerating into self-consistency of a single model. During training, the timestamps are split into keyframes and non-keyframes: the keyframes supervise the Direct query photometrically, while the non-keyframes supervise the Rollout model exclusively, with the rollout source state taken from the Direct query at the nearest keyframe and stop-gradiented. Finally, a multi-step transition-consistency loss requires Direct to match the rollout states at every intermediate time, writing the learned dynamics back into a single query. Our contributions are threefold:

- We decompose motion into two complementary components, which respectively account for fundamental, exactly composable positional motion and high-frequency local positional residuals as well as variations in the remaining attributes. The two paths share the DCT and the Triplane but use independent nonlinear networks, thereby ensuring non-degenerate transitions.
- We design a state-dependent local flow as the rollout model, which takes the current local state as input and predicts its subsequent state. This nonlinear local flow is combined with the DCT-based temporal evolution operator, whose evolution operators form a semigroup under function composition, to define the overall state transition. Through a transition-distillation, the Direct model is aligned with the states reached by the local flow.
- We build a unified framework for 4DGS training, distillation, and inference that couples state-transition learning with single-query inference through joint training and online distillation. It alleviates motion blur and detail loss under fast and complex motion, improves long-horizon temporal consistency and dynamic reconstruction quality, and remains compatible with existing deformation-based 4DGS pipelines.

2 Related Work

3DGS for 4D reconstruction Static 3DGS establishes the representational foundation built upon explicit Gaussians to accelerate differentiable rasterization (Kerbl et al., 2023). Dynamic NeRF methods validate the canonical-plus-deformation-field paradigm for modeling temporal consistency across timestamps (Pumarola et al., 2021). Building on these advances, dynamic Gaussian Splatting

has formed two distinct families of representations. The canonical-deformation branch keeps Gaussians in a reference space and models temporal variation through deformation fields. Deformable-GS, GauFRe and MoDGS progressively improve deformation modeling through shared offsets, static/dynamic separation and geometry-aware regularization (Yang et al., 2024b; Liang et al., 2025; Liu et al., 2025). Subsequent studies improve the stability, encoding efficiency, initialization, and optimization of canonical representations under challenging observations (Lu et al., 2024; Xu et al., 2024; Shi et al., 2026; Jiao et al., 2026b; He et al., 2026). In parallel, the direct spatio-temporal branch models Gaussians directly in continuous time. Yang et al. (Yang et al., 2024a) introduce 4D Gaussian Splatting to represent dynamic scenes as a collection of 4D primitives with explicit geometry and appearance, enabling real-time rendering of complex motions. 4DGS and 4DRotorGS introduce explicit spatio-temporal Gaussian representations, while SaRO-GS and FreeTimeGS improve robustness to large motion, topology changes and unconstrained temporal emergence (Wu et al., 2024; Duan et al., 2024; Yan et al., 2024; Wang et al., 2025). Hybrid 3D-4DGS further reduces redundancy in static regions by converting time-invariant Gaussians to 3D representations (Oh et al., 2025). Collectively, these methods deliver stable, compact, real-time-renderable representations for dynamic scenes. Nevertheless, their outputs remain time-conditioned query results without real motion.

Motion modeling in 4DGS Existing approaches improve Gaussian motion modeling mainly through structured motion primitives, trajectory representations and adaptive capacity allocation. Kwak et al. (2025) decompose dynamic Gaussian motion into a global component and a local residual component. SC-GS, SP-GS and CoGS introduce sparse control structures or grouped motion representations to improve efficiency, rigidity and editability (Huang et al., 2024; Wan et al., 2024; Yu et al., 2024). SplineGS and TRiGS further model continuous trajectories and preserve cross-frame Gaussian correspondence for long-term motion consistency (Park et al., 2025; Yeom et al., 2026). CTRL-GS, MAPo and MoRel instead allocate motion capacity hierarchically or through local anchors to accommodate heterogeneous and long-duration dynamics (Hou et al., 2025; Jiao et al., 2026a; Kwak et al., 2026). SPIN-4DGS and MoRGS target large inter-frame displacement and online streaming, respectively, by directly predicting spatio-temporal attributes or exploiting explicit optical-flow cues (Kim et al., 2026; Lee et al., 2026). Though these works greatly enhance trajectory expressiveness, long-term consistency, and motion is still formulated as time-indexed look-ups or time-conditioned mapping.

The dynamical-systems perspective on 4DGS These methods move beyond per-timestamp localization and treat motion as a time-evolving system. NeHaD (Qin et al., 2025) embeds Hamiltonian networks for energy conservation, while PIDG (Hong et al., 2026) and LagrangeGS (Sato et al., 2026) apply continuum-mechanical or Lagrangian constraints. EvoGS (Asiimwe & Vondrick, 2025), ODE-GS (Wang et al., 2026), and ParticleGS (Quan et al., 2026) utilize Neural ODEs for continuous state evolution, and WassersteinGS (Deng et al., 2025) employs state-space regularization.These works achieve temporal consistency by explicitly integrating dynamical systems or imposing structured physical/geometric priors, tightly coupling inference to specific conservation laws or integration paths. Consequently, they operate under a distinct problem setting that requires strong physical priors and inference-time rollout.

3 METHOD

Given a multi-view spatiotemporal sequence, we learn a time-varying Gaussian scene $\mathcal{G}(t)$ whose rendering matches the observations at novel views and times. We separate absolute-state querying from state transition: a lightweight *Direct* model predicts the deformation state with one state-free query, while a higher-capacity *Rollout* model learns state-dependent transitions during training.

3.1 OPERATOR DECOMPOSITION

Each canonical Gaussian $\mathcal{G}_i^0 = \{x_i, s_i, q_i, o_i, c_i\}$ provides the shared geometry and appearance basis. To ensure that the two nonlinear paths share the same motion identity and spatial context, we use a shared Triplane field queried at a spatial position x:

$$z_{latent}(x) = \Phi\big(P_{xy}(x,y), P_{yz}(y,z), P_{zx}(z,x)\big), \tag{1}$$

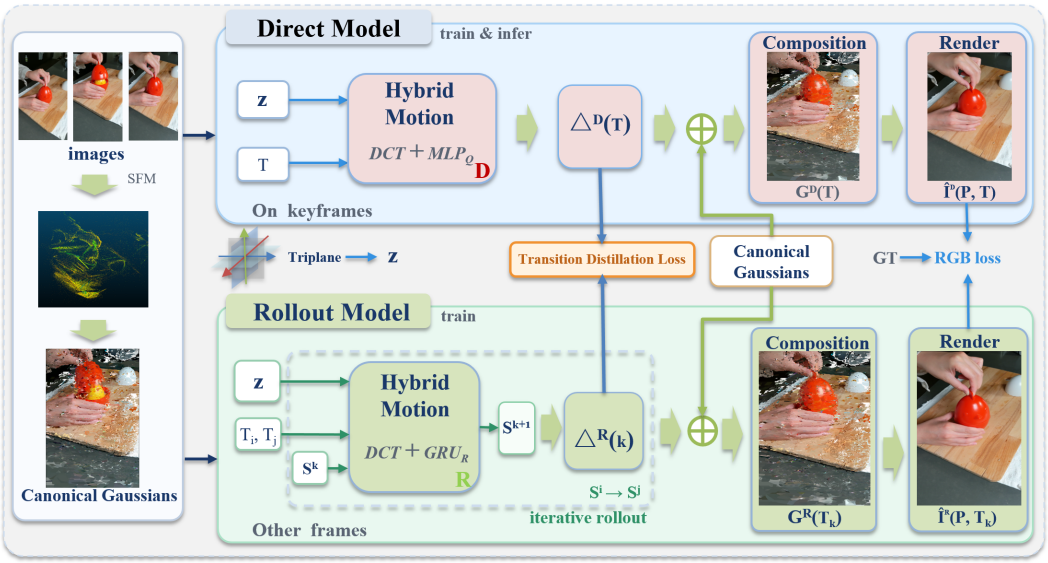

Figure 2: **Overview of the proposed method.** We learn state-dependent local dynamics through training-time rollouts, while retaining a single-query Direct branch for efficient inference. Keyframes supervise direct reconstruction, intermediate frames supervise rollout transitions, and transition consistency distills the learned dynamics back into the Direct decoder. At inference, only the distilled Direct branch is used.

where P_{xy}, P_{xz}, and P_{yz} are learnable plane features and Φ is the fusion and projection. The Direct path reads $z_{latent}(x_i)$ at the canonical position, while the Rollout path reads the same field at the deformed position.

To exploit the explicit low-frequency structure of positional motion without imposing the same linear-composition assumption on the remaining attributes, we restrict the discrete cosine transform (DCT) time-augmented semigroup to position. The positional DCT coefficients are predicted by a shared MLP g_θ from $z_{latent}(x_i)$:

$$A_i^x = g_\theta\big(z_{latent}(x_i)\big), \tag{2}$$

and the low-frequency positional increment over an interval is:

$$L_i^x(t_a, t_b) = A_i^x\big[\psi(t_b) - \psi(t_a)\big], \tag{3}$$

where ψ is a truncated DCT basis. To ensure that a single Direct query and a segmented Rollout rollout represent the same positional motion, we require the interval increment to be exactly composable: applying $L_i^x(t_a, t_b)$ and then $L_i^x(t_b, t_c)$ must equal $L_i^x(t_a, t_c)$. The DCT increment in Eq. equation 3 satisfies this property by construction:

$$L_i^x(t_a, t_c) = L_i^x(t_a, t_b) + L_i^x(t_b, t_c). \tag{4}$$

Equivalently, the interval maps $\Phi_{a,b}(x) = x + L_i^x(a, b)$ satisfy

$$\Phi_{b,c} \circ \Phi_{a,b} = \Phi_{a,c}, \qquad \Phi_{a,a} = \mathrm{Id}. \tag{5}$$

We therefore refer to the time-augmented state space of DCT positional component as an exactly composable positional semigroup, more precisely a monoid of interval maps under interval concatenation. It is not a classical one-parameter semigroup, since the DCT basis is not additive in time; the precise algebraic characterization is given in Appendix B. This property guarantees that Direct single-query evaluation and segmented accumulation agree by construction. To produce the complete deformation in one query while separating composable positional motion from the other attributes, we define the Direct residual as:

$$\begin{aligned} \Delta x_i^D(t) &= L_i^x(t) + Q_\phi^x\big(z_{latent}(x_i), \varphi(t)\big), \\ \Delta r_i^D(t) &= Q_\phi^r\big(z_{latent}(x_i), \varphi(t)\big), \quad r \in \{s, q, o, c\}, \end{aligned} \tag{6}$$

Q_ϕ^x and Q_ϕ^r are the Direct output heads, $\varphi(t)$ is the time encoding, and r indexes scale, rotation, opacity, and color. The Direct model is state-free.

The Rollout path learns how the current state evolves to the next state. It maintains the arrived state S^k and a GRU hidden state h^k. To make the transition aware of the actual reached location, we evaluate the local spacetime feature at the deformed position:

$$\gamma_i(S_i^k, t^k) = W_\gamma\big(x_i + S_i^{k,x},\, t^k - t_0\big), \tag{7}$$

where $S_i^{k,x}$ is the displacement component and W_γ is the encoding projection. To retain short-term motion patterns that cannot be recovered from the instantaneous state, the GRU updates the hidden state as:

$$h^{k+1} = \mathrm{GRU}\big(h^k,\, [z_{latent}(x_i + S_i^{k,x}), S_i^k, \varphi(\delta t^k)]\big), \tag{8}$$

where $\delta t^k = t^{k+1} - t^k$. The state-dependent nonlinear increment is:

$$F_i^R = \overline{\delta t}^{\,k}\, H_\theta\big(z_{latent}(x_i + S_i^{k,x}), S_i^k, h^k, \varphi(t^k), \gamma_i(S_i^k, t^k)\big), \tag{9}$$

where $\overline{\delta t}^{\,k}$ is the clipped step and H_θ is the state-dependent transition network. Positional states combine the DCT and nonlinear increments, while the remaining attributes use only the nonlinear increment:

$$S_i^{k+1,x} = S_i^{k,x} + L_i^x(t^k, t^{k+1}) + F_i^{R,x}, \tag{10}$$

$$S_i^{k+1,r} = S_i^{k,r} + F_i^{R,r}, \quad r \in \{s, q, o, c\}. \tag{11}$$

We denote this one-step operator by T_θ. To prevent the rollout source state from being modified by the distillation target, multi-step rollout starts from the detached Direct state:

$$S_i^{(0)} = \mathrm{sg}\big[\Delta_i^D(t^{(0)})\big], \qquad \big(S_i^{(k+1)}, h^{k+1}\big) = T_\theta\big(S_i^k, h^k, t^k\big). \tag{12}$$

Since H_θ depends on the arrived state and hidden state, the same time increment produces different outputs from different arrived states, so the rollout cannot collapse to a difference of absolute states.

For a rollout segment from source time t_{src} to target time t_{tgt}, the target state is written as the accumulated sum of per-step increments:

$$\begin{aligned} S_i^{(K),x} &= \mathrm{sg}\big[\Delta x_i^D(t_{\mathrm{src}})\big] + \sum_{k=0}^{K-1} \Big[L_i^x(t^k, t^{k+1}) + F_i^{R,x,(k+1)}\Big], \\ S_i^{(K),r} &= \mathrm{sg}\big[\Delta r_i^D(t_{\mathrm{src}})\big] + \sum_{k=0}^{K-1} F_i^{R,r,(k+1)}, \quad r \in \{s, q, o, c\}, \end{aligned} \tag{13}$$

where $t^{(0)} = t_{\mathrm{src}}$, $t^{(K)} = t_{\mathrm{tgt}}$, and $F_i^{R,x,(k+1)}$ and $F_i^{R,r,(k+1)}$ are the nonlinear increments at step $k+1$ from Eq. equation 9.

3.2 Supervision Separation

We partition the training timestamps into a keyframe set $\mathcal{K}$ and its complement. Keyframes supervise the Direct query with an image loss,

$$\mathcal{L}_D^{\mathrm{img}} = \ell_{\mathrm{img}}\big(\hat{I}^D(t), I(t)\big), t \in \mathcal{K} \tag{14}$$

while non-keyframes never supervise the Direct query and are used for Rollout training only. Keyframes are selected at a fixed stride and always include the first and last timestamps; each rollout restarts from the nearest keyframe using the Direct query at that time, which bounds the integration window. If the rollout always advanced from the previous timestamp, long-sequence errors would accumulate and the training signal would concentrate on increasingly unreliable deep states. Conversely, if the Direct query absorbed the photometric supervision at non-keyframes as well, the transition operator would receive only an indirect residual signal and its transition would not be identifiable. We therefore supervise the Rollout model at every non-keyframe transition step:

$$\mathcal{L}_R^{\mathrm{img}} = \sum_{k=1}^{K} \ell_{\mathrm{img}}\big(\hat{I}^R(t^{(k)}), I^{(k)}\big), \tag{15}$$

ℓ_{img} is the L1 and SSIM photometric loss. The rollout source state is obtained from the Direct query and is stop-gradiented, so the Rollout training signal does not update the Direct source state.

3.3 TRANSITION-CONSISTENCY DISTILLATION

Matching only the final rollout state is insufficient to transfer long-horizon composition. We therefore align the Direct query with the rollout state at every intermediate time:

$$\mathcal{L}_{\text{trans}} = \sum_{k=1}^{K} \frac{1}{K} \frac{1}{N} \sum_{i} \sum_{r} \left\| \Delta_{i,r}^{D}(t^{(k)}) - \text{sg}\big[S_{i,r}^{(k)}\big] \right\|_1, \tag{16}$$

where K is the rollout length, N is the number of Gaussians, i and r index primitives and residual channels, and sg denotes stop-gradient. The stop-gradient fixes the knowledge direction from Rollout to Direct, so the rollout model is not pulled backward by the direct.

3.4 TRAINING AND INFERENCE

The full objective is:

$$\mathcal{L} = \mathcal{L}_{D}^{\text{img}} + \lambda_R \mathcal{L}_{R}^{\text{img}} + \lambda_{\text{trans}} \mathcal{L}_{\text{trans}} + \lambda_s \mathcal{L}_{\text{smooth}}, \tag{17}$$

where the auxiliary terms regularize cross-time appearance consistency and motion smoothness. Training first establishes the canonical geometry and the positional DCT time-augmented semigroup, then increases the rollout span to learn reliable transitions, and finally increases λ_{trans} for distillation. At inference, the Rollout model and GRU are removed, and only:

$$\Delta^{D}(t) = L^{x}(t) + Q_{\phi}\big(z_{latent}(x), \varphi(t)\big) \tag{18}$$

is evaluated. The cost is independent of the temporal span and rollout depth, while the single query retains the multi-step dynamics through distillation.

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

We implement the proposed method based on 4DGS, retaining most hyperparameters. For the newly introduced modules, We use 32 DCT bases for the low-frequency positional semigroup, a 128^3 triplane feature field, a 3-layer Direct MLP with width 128, and a GRU-based Rollout module with hidden size 256 and 4 layers. Training is performed in stages, starting with a 3K-iteration coarse stage, and followed by fine stage with 20K-40K iterations. Transition-consistency distillation is activated only after both Direct and Rollout branches have converged and the Rollout loss falls below the Direct loss.

4.2 DATASETS AND METRICS

We evaluate on three widely used benchmarks with complementary characteristics. The D-NeRF(Pumarola et al., 2021) dataset is a synthetic monocular benchmark. HyperNeRF(Park et al., 2021) contains real-world monocular videos. N3DV (Neural 3D Video)Li et al. (2022) is a multi-view dynamic benchmark captured with a dense camera array. We use PSNR, SSIM, and LPIPS as evaluation metrics, while $\uparrow$ indicates higher is better, while $\downarrow$ indicates lower is better.

4.3 BASELINES

We compare against five representative 4D Gaussian splatting baselines that cover the main paradigms of dynamic scene reconstruction. We select 4DGS, Deformable 3DGS , SC-GS, MoDec-GS, and WassersteinGS as baselines. All these baselines operate under the same problem setting as our method. Dynamics-based approaches (e.g., NeHaD, EvoGS, ODE-GS, ParticleGS) are not included in the quantitative comparison because they rely on distinct problem settings involving strong physical constraints and/or iterative rollout at inference, making them orthogonal to our prior-free, single-step formulation. We therefore focus on the above five baselines for a fair and controlled evaluation.

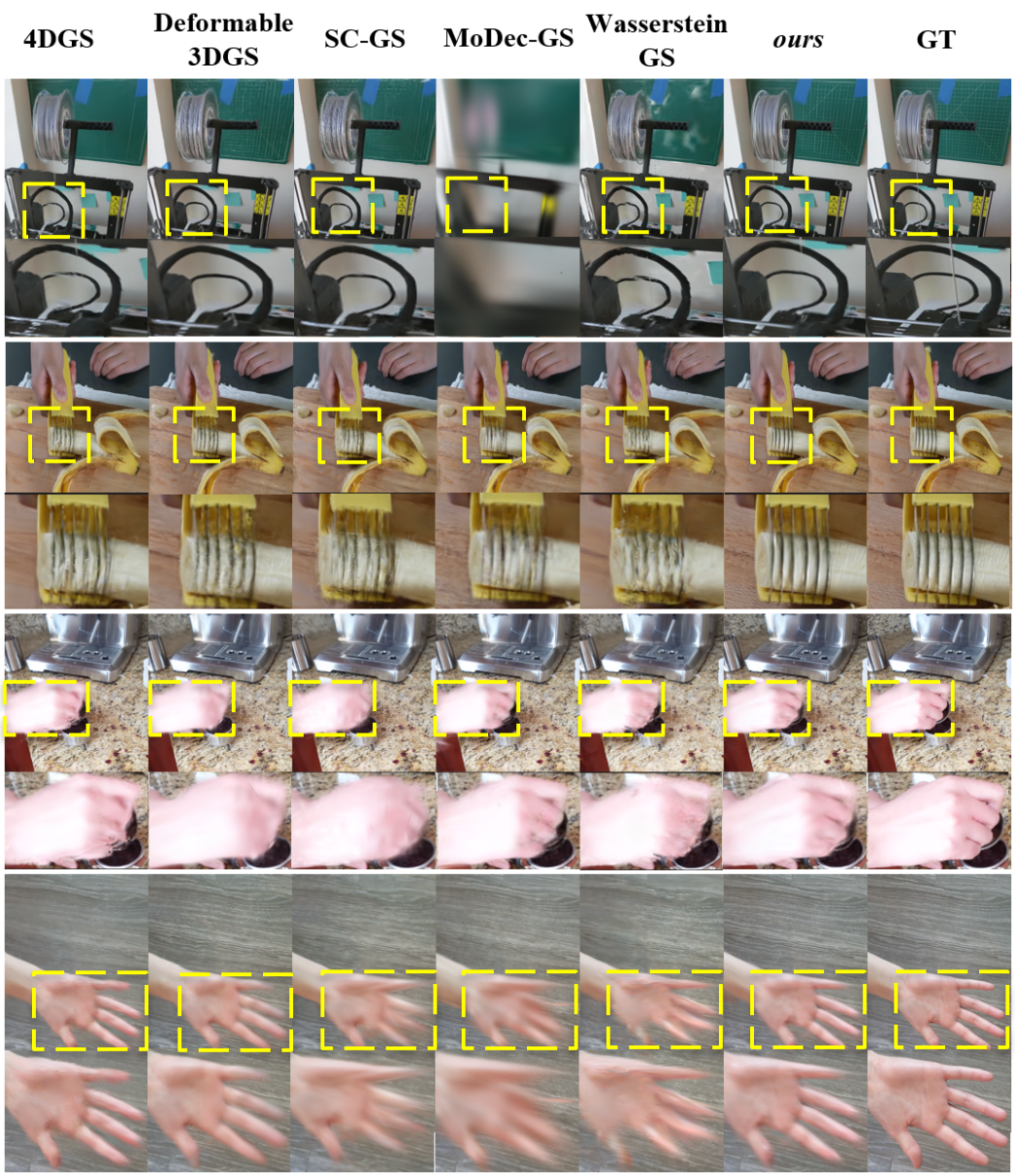

Figure 3: Qualitative results comparison on HyperNeRF dataset.

Table 1: Quantitative comparison on HyperNeRF.

Methods	HyperNeRF–interp			HyperNeRF–misc			HyperNeRF–vrig		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
4DGS(Wu et al., 2024)	28.26	0.825	0.258	**28.79**	0.876	0.202	24.81	0.686	0.344
Deformable-3DGS(Yang et al., 2024b)	25.94	0.773	0.306	27.34	0.859	0.227	22.12	0.591	0.391
SC-GS(Huang et al., 2024)	25.42	0.820	0.373	26.64	0.888	0.339	22.03	0.633	0.387
MoDec-GS(Kwak et al., 2025)	26.70	0.824	0.274	26.51	0.833	0.237	23.68	0.669	0.382
WassersteinGS(Deng et al., 2025)	24.49	0.738	0.356	26.04	0.821	0.289	22.38	0.639	0.385
Ours	**28.35**	**0.872**	**0.190**	28.68	**0.889**	**0.169**	**25.26**	**0.731**	**0.266**

4.4 Experimental Results

Tables 1, and 2 report quantitative comparisons on HyperNeRF, and N3DV. Figures 3 and 4 report the comparison of visualization results of our method and baseline.In addition to HyperNeRF and N3DV, we report the D-NeRF results in the supplementary material.

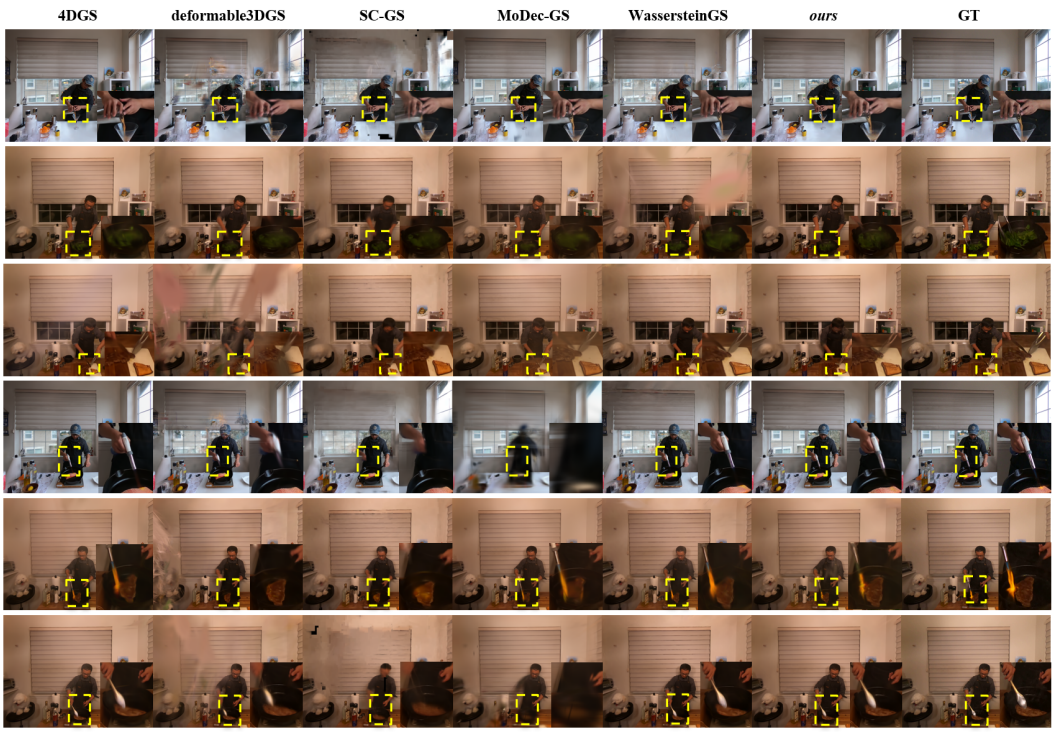

Figure 4: Qualitative results comparison on N3DV dataset. The comparisons in the yellow dashed regions show that the proposed method better preserves local motion, effectively alleviating motion blur and improving the visual quality of 4D reconstruction.

Results on HyperNeRF. Fig3 shows the visual comparison on the HyperNeRF dataset. For fine structural scenes such as vrig-3dprinter, slice-banana, tamping,and so on, MoDec-GS suffers from severe structural degradation and motion blur. Other baselines all produce blurred edges and noticeable splatting artifacts. In contrast, our method recovers sharp, complete contours, texture details and motion structure, which shows that our method is effective in reducing motion blur by modeling more realistic motion.

Results on N3DV. On the real-world N3DV dataset with natural human motions and realistic illumination, most baselines struggle with accurate motion alignment in real dynamic sequences, resulting in obvious motion blur, texture misalignment. In particular, fast-moving body parts appear heavily blurred and distorted in results of Deformable-3DGS, SC-GS and MoDec-GS. 4DGS produces relatively better results but still loses fine-grained textures and exhibits edge softening. Our method achieves the best visual fidelity on this dataset: it accurately tracks dynamic motions to minimize motion blur, preserves rich fine textures including flame and all moving objects, and maintains clean and sharp object contours. The reconstructed results fully validate the practical advantage of our method in real-world dynamic view synthesis.

Table 2: Quantitative comparison on the N3DV dataset.

Methods	PSNR↑	SSIM↑	LPIPS↓
4DGS(Wu et al., 2024)	28.76	0.922	0.125
Deformable-3DGS(Yang et al., 2024b)	25.23	0.899	0.202
SC-GS(Huang et al., 2024)	26.80	0.885	0.289
MoDec-GS(Kwak et al., 2025)	26.51	0.894	0.145
WassersteinGS(Deng et al., 2025)	26.85	0.891	0.189
Ours	**29.36**	**0.936**	**0.098**

Table 3: Comparison of ablation results on HyperNeRF-misc.

Components	PSNR↑	SSIM↑	LPIPS↓
Freely learnable z	27.39	0.818	0.254
DCT-only	24.35	0.776	0.330
MLP-only	27.06	0.861	0.223
w/o distillation	28.52	0.875	0.198
All	**28.68**	**0.889**	**0.169**

4.5 Ablation Study and Analysis

We conduct ablations on HyperNeRF-misc to evaluate the contribution of each component. We first compare different motion representations, including a freely learnable z (instead of querying the feature z from the Triplane), DCT-only, and MLP-only variants. We then remove the proposed distillation objective to evaluate its contribution to the final Direct model. The quantitative results are reported in Table 3.

Ablation on Triplane Conditioning and Freely Learnable z. The freely learnable z underperforms the shared Triplane conditioning across all metrics. While a free latent can fit partial motion, it lacks stable shared spatial context, degrading local geometry and perceptual quality. Conversely, Triplane conditioning better preserves structural consistency and details, validating that spatial queries over a shared field serve as a more effective motion representation for both paths.

Ablation on motion representation. Table 3 shows that DCT-only gets the worst performance, which indicates that a purely low-frequency, composable DCT representation is insufficient for nonlinear local motion. Although MLP-only outperforms DCT-only, it still lags behind the full model. It shows that strong nonlinear fitting capacity alone lacks the explicit structural constraint. The full model achieves the best results across all metrics, demonstrating that DCT and MLP are highly complementary in motion modeling.

Ablation on Transition-Consistency Distillation Removing distillation degrades all metrics, confirming that rollout-learned state transitions are internalized by the direct model rather than serving only as an auxiliary training branch. The reconstruction gains are moderate but consistent; the main benefit is expected in long-horizon stability and temporal consistency under fast motion.

5 Conclusion

We presented a dynamic Gaussian splatting method that separates absolute-state query from state-conditioned transition learning: a lightweight direct model predicts deformation in one query, while a higher-capacity rollout model learns local transitions under independent multi-step supervision. A transition-consistency loss distills rollout dynamics into the direct model, keeping inference as a single constant-cost query. Experiments on D-NeRF, HyperNeRF, and N3DV validate the design. Our method does not model dynamics extrapolation and only interpolates within the observed time range. We also omit quantitative comparison with dynamics-based methods requiring physical priors or inference-time rollout, since they target a different setting. Future work will address extrapolation, unified benchmarks for fair comparison, and more complex scenes.

AI use statement

In this work, we have not used generative AI tools for any tasks with required disclosure. Specifically, we did not use them to generate synthetic data sets, develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean or reformat datasets, support qualitative and thematic data analysis, or interpret results. We have not used generative AI tools for these tasks, and the remaining required disclosure tasks (e.g., assisting in the writing of proofs) are not applicable to this work.

Additionally, we used generative AI tools for recommended disclosure tasks, specifically to edit this research paper to improve readability and to polish the language, grammar, and flow of the manuscript. No other recommended disclosure tasks were performed with AI assistance.

We have reviewed all AI-assisted work. All AI-polished text was meticulously reviewed and manually edited by the authors to ensure that the scientific meaning, technical accuracy, and original arguments were strictly preserved. We verified that no citations, claims, or technical details were altered or hallucinated during the polishing process. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Arnold Caleb Asiimwe and Carl Vondrick. EvoGS: 4D Gaussian Splatting as a Learned Dynamical System, 2025.

Junli Deng, Ping Shi, Yihao Luo, and Qipei Li. Gaussians on their Way: Wasserstein-Constrained 4D Gaussian Splatting with State-Space Modeling. *Computer Graphics Forum*, 44(7), 2025. doi: 10.1111/cgf.70271.

Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes. In *ACM SIGGRAPH 2024 Conference Papers*. ACM, 2024. doi: 10.1145/3641519.3657463.

Jiahao He, Yihua Shao, Zhengkai Zhao, Pan Gao, Fei Ma, Jingcai Guo, Hao Tang, Nicu Sebe, and Qi Tian. GrainGS: Gradient-Decoupled Gaussian Splatting for Efficient Dynamic Novel View Synthesis, 2026.

Haoqin Hong, Ding Fan, Fubin Dou, Zhili Zhou, Haoran Sun, Congcong Zhu, and Jingrun Chen. Physics-Informed Deformable Gaussian Splatting: Towards Unified Constitutive Laws for Time-Evolving Material Field. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4725–4733, 2026. doi: 10.1609/aaai.v40i6.42474.

Karly Hou, Wanhua Li, and Hanspeter Pfister. CTRL-GS: Cascaded Temporal Residue Learning for 4D Gaussian Splatting, booktitle = 4D Vision Workshop at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR Workshops). 2025.

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: Sparse-Controlled Gaussian Splatting for Editable Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4220–4230, 2024.

Han Jiao, Jiakai Sun, Yexing Xu, Lei Zhao, Wei Xing, and Huaizhong Lin. MAPo: Motion-Aware Partitioning of Deformable 3D Gaussian Splatting for High-Fidelity Dynamic Scene Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11767–11776, 2026a.

Han Jiao, Jiakai Sun, Lei Zhao, Zhanjie Zhang, Wei Xing, and Huaizhong Lin. Fast and Robust Deformable 3D Gaussian Splatting. *IEEE Transactions on Visualization and Computer Graphics*, 2026b. doi: 10.1109/TVCG.2026.3675272. early access.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Seung-gyeom Kim, Areum Kim, Yongjae Yoo, and Sukmin Yun. Implicit 4D Gaussian Splatting for Fast Motion with Large Inter-Frame Displacements. In *International Conference on Learning Representations*, 2026.

Sangwoon Kwak, Joonsoo Kim, Jun Young Jeong, Won-Sik Cheong, Jihyong Oh, and Munchurl Kim. MoDec-GS: Global-to-Local Motion Decomposition and Temporal Interval Adjustment for Compact Dynamic 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11338–11348, 2025. doi: 10.1109/CVPR52734.2025.01059.

Sangwoon Kwak, Weeyoung Kwon, Jun Young Jeong, Geonho Kim, Won-Sik Cheong, and Jihyong Oh. MoRel: Long-Range Flicker-Free 4D Motion Modeling via Anchor Relay-based Bidirectional Blending with Hierarchical Densification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37370–37379, 2026.

Wonjoon Lee, Sungmin Woo, Donghyeong Kim, Jungho Lee, Sangheon Park, and Sangyoun Lee. MoRGS: Efficient Per-Gaussian Motion Reasoning for Streamable Dynamic 3D Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41044–41053, 2026.

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D Video Synthesis from Multi-view Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5511–5521, 2022. doi: 10.1109/CVPR52688.2022.00544.

Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. GauFRe: Gaussian Deformation Fields for Real-Time Dynamic Novel View Synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2642–2652, 2025.

Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lyu, Peng Wang, Wenping Wang, and Junhui Hou. MoDGS: Dynamic Gaussian Splatting from Casually-captured Monocular Videos with Depth Priors. In *International Conference on Learning Representations*, 2025.

Jiahao Lu, Jiacheng Deng, Ruijie Zhu, Yanzhe Liang, Wenfei Yang, Tianzhu Zhang, and Xu Zhou. DN-4DGS: Denoised Deformable Network with Temporal-Spatial Aggregation for Dynamic Scene Rendering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Seungjun Oh, Younggeun Lee, Hyejin Jeon, and Eunbyung Park. Hybrid 3D-4D Gaussian Splatting for Fast Dynamic Scene Representation, 2025.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jaesung Oh, and Munchurl Kim. SplineGS: Robust Motion-Adaptive Spline for Real-Time Dynamic 3D Gaussians from Monocular Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26866–26875, 2025.

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. *ACM Transactions on Graphics*, 40 (6):1–12, 2021. doi: 10.1145/3478513.3480487.

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 (CVPR)*, pp. 10318–10327, 2021.

Hai-Long Qin, Sixian Wang, Guo Lu, and Jincheng Dai. Neural Hamiltonian Deformation Fields for Dynamic Scene Rendering. In *ACM SIGGRAPH Asia 2025 Conference Papers*. ACM, 2025. doi: 10.1145/3757377.3764000.

Jinsheng Quan, Qiaowei Miao, Yichao Xu, Zizhuo Lin, Ying Li, Wei Yang, Zhihui Li, and Yawei Luo. ParticleGS: Learning Neural Gaussian Particle Dynamics from Videos for Prior-free Physical Motion Extrapolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8331–8341, 2026.

Shogo Sato, Takuhiro Kaneko, Shoichiro Takeda, Tomoyasu Shimada, Riku Inoue, Kazuhiko Murasaki, and Ryuichi Tanida. LagrangeGS: Non-Conservative Lagrangian System on Dynamic 3D Gaussian Splatting, 2026.

Changyue Shi, Chuxiao Yang, Xinyuan Hu, Minghao Chen, Wenwen Pan, Yan Yang, Jiajun Ding, Zhou Yu, and Jun Yu. Sparse4DGS: 4D Gaussian Splatting for Sparse-Frame Dynamic Scene Reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8933–8941, 2026. doi: 10.1609/aaai.v40i11.37848.

Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint Gaussian Splatting for Real-Time High-Fidelity Dynamic Scene Reconstruction. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 49957–49972, 2024.

Daniel Wang, Patrick Rim, Tian Tian, Dong Lao, Alex Wong, and Ganesh Sundaramoorthi. ODE-GS: Latent ODEs for Dynamic Scene Extrapolation with 3D Gaussian Splatting. In *International Conference on Learning Representations*, 2026.

Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. FreeTimeGS: Free Gaussian Primitives at Anytime Anywhere for Dynamic Scene Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21750–21760, 2025.

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 (CVPR)*, pp. 20310–20320, 2024.

Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4D: 4D Decomposed Hash Encoding for High-Fidelity Dynamic Gaussian Splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 4D Gaussian Splatting with Scale-aware Residual Field and Adaptive Optimization for Real-time Rendering of Temporally Complex Dynamic Scenes. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM)*. ACM, 2024. doi: 10.1145/3664647.3681463.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-Time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. In *International Conference on Learning Representations*, 2024a. URL https://mlanthology.org/iclr/2024/yang2024iclr-realtime/.

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 (CVPR)*, pp. 20331–20341, 2024b.

Suwoong Yeom, Joonsik Nam, Seunggyu Choi, Lucas Yunkyu Lee, Sangmin Kim, Jaesik Park, Joonsoo Kim, Kugjin Yun, Kyeongbo Kong, and Sukju Kang. TRiGS: Temporal Rigid-Body Motion for Scalable 4D Gaussian Splatting, 2026.

Heng Yu, Joel Julin, Zoltán Á. Milacski, Koichiro Niinuma, and László A. Jeni. CoGS: Controllable Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21624–21633, 2024.

A Results on D-NeRF

Results on D-NeRF. SC-GS and Deformable-3DGS achieve the highest numerical accuracy on this benchmark. This performance gain stems from their explicit deformation modeling tailored for simple smooth motions, which enables high-fidelity fitting on low-complexity synthetic scenes. Our method ranks third, which significantly outperforming others mmethods. Notably, although our method does not reach the top numerical precision on this synthetic benchmark, it delivers perceptual quality comparable to Deformable-3DGS. We attribute the gap on D-NeRF to the absence of high-frequency nonlinear motion, where the proposed local-flow distillation provides limited additional benefit. Our method is designed for real-world dynamic scenes with complex, fast motion. More importantly, as demonstrated in subsequent experiments, it exhibits far stronger generalization when transferred to more complex real-world scenarios.

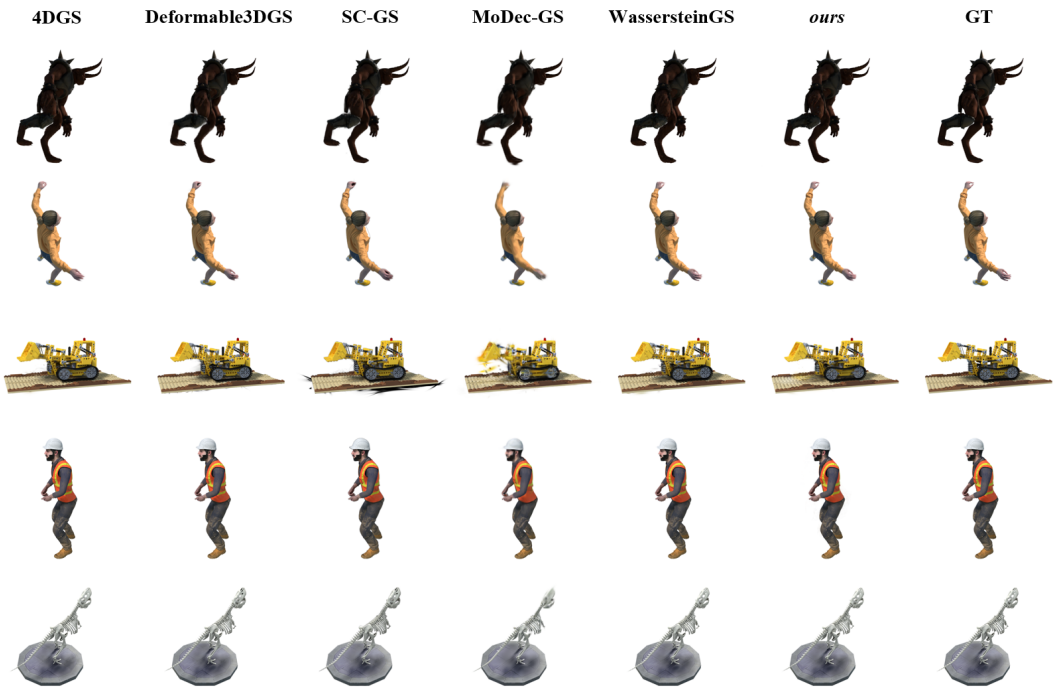

Figure 5: Qualitative results comparison on D-NeRF dataset.

Table 4: Quantitative comparison on the D-NeRF dataset.

Method	PSNR↑	SSIM↑	LPIPS↓
4DGS(Wu et al., 2024)	34.59	0.981	0.023
Deformable-3DGS(Yang et al., 2024b)	39.18	0.989	0.014
SC-GS(Huang et al., 2024)	**40.67**	**0.995**	**0.010**
MoDec-GS(Kwak et al., 2025)	29.63	0.961	0.043
WassersteinGS(Deng et al., 2025)	34.67	0.983	0.019
Ours	37.84	0.986	0.014

B Proof of the DCT Positional Semigroup

B.1 Definitions

Let $t \in \mathbb{R}$ denote the normalized continuous time variable. The interval $[0, 1]$ is used as the normalized observation or training time domain, while the DCT basis functions are naturally defined for all $t \in \mathbb{R}$. Let the truncated DCT basis be

$$\psi(t) = \big[\cos(\pi t), \cos(2\pi t), \ldots, \cos(K\pi t)\big]^\top \in \mathbb{R}^K, \tag{19}$$

where K is the number of retained DCT basis functions.

For a Gaussian with positional DCT coefficient matrix $A \in \mathbb{R}^{3\times K}$, define the corresponding continuous positional representation by

$$p(t) = A\psi(t) \in \mathbb{R}^3. \tag{20}$$

For two time instants $a, b \in \mathbb{R}$, the positional increment is defined as

$$L(a, b) = p(b) - p(a) = A\big[\psi(b) - \psi(a)\big] \in \mathbb{R}^3. \tag{21}$$

Instead of defining an interval transformation indexed by both its start and end times, we augment the positional state with the absolute time. The augmented state space is therefore

$$\mathcal{X} = \mathbb{R}^3 \times \mathbb{R}, \tag{22}$$

and an augmented state is written as

$$X = (x, t) \in \mathcal{X}. \tag{23}$$

For an elapsed time $s \geq 0$, define the evolution operator $\Psi_s : \mathcal{X} \to \mathcal{X}$ by

$$\Psi_s(x, t) = \big(x + A\big[\psi(t + s) - \psi(t)\big],\ t + s\big)\,. \tag{24}$$

Let

$$\mathcal{S} = \{\Psi_s \mid s \geq 0\} \tag{25}$$

be the family of all non-negative-time evolution operators, equipped with the usual composition of mappings.

The purpose of this appendix is to prove that $(\mathcal{S}, \circ)$ forms a one-parameter monoid, and hence a one-parameter semigroup with identity. In particular, the proof establishes that direct evaluation over a time interval is exactly equivalent to successive application of arbitrary temporal increments.

B.2 Temporal Increment Additivity

The key property underlying the construction is the additivity of positional increments.

Lemma 1 (Temporal increment additivity)**.** *For any $a, b, c \in \mathbb{R}$,*

$$L(a, c) = L(a, b) + L(b, c). \tag{26}$$

Proof. By the definition of the positional increment,

$$L(a, b) = A\big[\psi(b) - \psi(a)\big], \tag{27}$$

and

$$L(b, c) = A\big[\psi(c) - \psi(b)\big]. \tag{28}$$

Therefore,

$$\begin{aligned} L(a, b) + L(b, c) &= A\big[\psi(b) - \psi(a)\big] + A\big[\psi(c) - \psi(b)\big] \\ &= A\big[\psi(b) - \psi(a) + \psi(c) - \psi(b)\big] \\ &= A\big[\psi(c) - \psi(a)\big] \\ &= L(a, c). \end{aligned} \tag{29}$$

Thus, the positional increment is additive under temporal concatenation.

B.3 Identity Operator

Lemma 2 (Identity)**.** *The operator corresponding to zero elapsed time is the identity mapping:*

$$\Psi_0 = \mathrm{Id}_{\mathcal{X}}. \tag{30}$$

Proof. For any augmented state $(x, t) \in \mathcal{X}$, by Eq. equation 24,

$$\begin{aligned} \Psi_0(x, t) &= (x + A[\psi(t + 0) - \psi(t)],\ t + 0) \\ &= (x + A[\psi(t) - \psi(t)],\ t) \\ &= (x, t). \end{aligned} \tag{31}$$

Hence,

$$\Psi_0(x, t) = \mathrm{Id}_{\mathcal{X}}(x, t) \tag{32}$$

for every $(x, t) \in \mathcal{X}$, which proves $\Psi_0 = \mathrm{Id}_{\mathcal{X}}$.

B.4 Composition Law

The central result is that temporal evolution operators compose according to addition of their elapsed times.

Lemma 3 (Semigroup composition law)**.** *For any $r, s \geq 0$,*

$$\Psi_r \circ \Psi_s = \Psi_{r+s}. \tag{33}$$

Proof. Let $(x, t) \in \mathcal{X}$ be arbitrary. Applying Ψ_s first gives

$$\Psi_s(x, t) = (x + A[\psi(t + s) - \psi(t)],\ t + s)\,. \tag{34}$$

Applying Ψ_r to the resulting state yields

$$\begin{aligned}(\Psi_r \circ \Psi_s)(x, t) &= \Psi_r\,(x + A[\psi(t + s) - \psi(t)],\ t + s) \\ &= \Big(x + A[\psi(t + s) - \psi(t)] \\ &\qquad + A[\psi((t + s) + r) - \psi(t + s)],\ (t + s) + r\Big). \\ &= \Big(x + A[\psi(t + s) - \psi(t) + \psi(t + s + r) - \psi(t + s)],\ t + s + r\Big) \\ &= \Big(x + A[\psi(t + (s + r)) - \psi(t)],\ t + (s + r)\Big)\end{aligned} \tag{35}$$

we obtain

$$(\Psi_r \circ \Psi_s)(x, t) = \Psi_{r+s}(x, t). \tag{36}$$

Because (x, t) was arbitrary,

$$\Psi_r \circ \Psi_s = \Psi_{r+s}. \tag{37}$$

B.5 Closure

Lemma 4 (Closure under composition)**.** *For any $\Psi_r, \Psi_s \in \mathcal{S}$,*

$$\Psi_r \circ \Psi_s \in \mathcal{S}. \tag{38}$$

Proof. By definition of $\mathcal{S}$,

$$r, s \geq 0. \tag{39}$$

Therefore,

$$r + s \geq 0. \tag{40}$$

By the composition law established in Lemma 3,

$$\Psi_r \circ \Psi_s = \Psi_{r+s}. \in \mathcal{S}. \tag{41}$$

Hence,

$$\Psi_r, \Psi_s \in \mathcal{S} \quad \Longrightarrow \quad \Psi_r \circ \Psi_s \in \mathcal{S}. \tag{42}$$

Thus $\mathcal{S}$ is closed under composition.

B.6 Associativity

Lemma 5 (Associativity)**.** *For any $\Psi_r, \Psi_s, \Psi_q \in \mathcal{S}$,*

$$(\Psi_q \circ \Psi_r) \circ \Psi_s = \Psi_q \circ (\Psi_r \circ \Psi_s). \tag{43}$$

Proof.

$$\begin{aligned}(\Psi_q \circ \Psi_r) \circ \Psi_s &= \Psi_{q+r} \circ \Psi_s \\ &= \Psi_{q+r+s},\end{aligned} \tag{44}$$

while

$$\begin{aligned}\Psi_q \circ (\Psi_r \circ \Psi_s) &= \Psi_q \circ \Psi_{r+s} \\ &= \Psi_{q+r+s}.\end{aligned} \tag{45}$$

Hence,

$$(\Psi_q \circ \Psi_r) \circ \Psi_s = \Psi_q \circ (\Psi_r \circ \Psi_s). \tag{46}$$

B.7 One-Parameter Monoid Structure

Theorem 1 (DCT positional one-parameter monoid)**.** *The family*

$$\mathcal{S} = \{\Psi_s \mid s \geq 0\} \tag{47}$$

equipped with function composition $\circ$ forms a monoid. More specifically,

$$\Psi_r \circ \Psi_s = \Psi_{r+s}, \qquad r, s \geq 0, \tag{48}$$

and

$$\Psi_0 = \mathrm{Id}_{\mathcal{X}}. \tag{49}$$

Proof. To establish that $(\mathcal{S}, \circ)$ is a monoid, we verify the three required properties.

1. Closure. By Lemma 4, for any $\Psi_r, \Psi_s \in \mathcal{S}$,

$$\Psi_r \circ \Psi_s = \Psi_{r+s} \in \mathcal{S}. \tag{50}$$

2. Associativity. By Lemma 5, composition is associative:

$$(\Psi_q \circ \Psi_r) \circ \Psi_s = \Psi_q \circ (\Psi_r \circ \Psi_s). \tag{51}$$

3. Identity. By Lemma 2,

$$\Psi_0 = \mathrm{Id}_{\mathcal{X}}. \tag{52}$$

Furthermore, for every $s \geq 0$,

$$\Psi_0 \circ \Psi_s = \Psi_{0+s} = \Psi_s, \tag{53}$$

and

$$\Psi_s \circ \Psi_0 = \Psi_{s+0} = \Psi_s. \tag{54}$$

Thus Ψ_0 is a two-sided identity element of $(\mathcal{S}, \circ)$.

Therefore, $(\mathcal{S}, \circ)$ is a monoid.

B.8 Commutativity

Proposition 1 (Commutativity of the temporal evolution family)**.** *For any $r, s \geq 0$,*

$$\Psi_r \circ \Psi_s = \Psi_s \circ \Psi_r. \tag{55}$$

Consequently, $(\mathcal{S}, \circ)$ is a commutative monoid.

Proof. By the composition law,

$$\Psi_r \circ \Psi_s = \Psi_{r+s}. \tag{56}$$

Similarly,

$$\Psi_s \circ \Psi_r = \Psi_{s+r}. \tag{57}$$

Since addition over $\mathbb{R}_{\geq 0}$ is commutative,

$$r + s = s + r. \tag{58}$$

Therefore,

$$\Psi_r \circ \Psi_s = \Psi_{r+s} = \Psi_{s+r} = \Psi_s \circ \Psi_r. \tag{59}$$

Hence $(\mathcal{S}, \circ)$ is a commutative monoid.

B.9 Equivalence Between Direct Evaluation and Segmented Accumulation

The semigroup property directly implies that evaluating a temporal displacement over a long interval is equivalent to applying a sequence of shorter temporal evolution operators.

Theorem 2 (Segmented temporal accumulation)**.** *Let*

$$0 = s_0 < s_1 < \cdots < s_N = s \tag{60}$$

be any partition of a non-negative elapsed time s. Then

$$\Psi_s = \Psi_{s_N - s_{N-1}} \circ \cdots \circ \Psi_{s_2 - s_1} \circ \Psi_{s_1 - s_0}. \tag{61}$$

Equivalently, for any initial state (x, t),

$$\begin{aligned} &\Psi_s(x,t) \\ &= \Psi_{s_N - s_{N-1}} \left(\cdots \Psi_{s_2 - s_1} \left(\Psi_{s_1 - s_0}(x,t) \right) \cdots \right). \end{aligned} \tag{62}$$

Proof. Since

$$s = (s_N - s_{N-1}) + \cdots + (s_2 - s_1) + (s_1 - s_0), \tag{63}$$

repeated application of the composition law

$$\Psi_r \circ \Psi_q = \Psi_{r+q} \tag{64}$$

gives

$$\begin{aligned} &\Psi_{s_N - s_{N-1}} \circ \cdots \circ \Psi_{s_1 - s_0} \\ &= \Psi_{(s_N - s_{N-1}) + \cdots + (s_1 - s_0)} \\ &= \Psi_s. \end{aligned} \tag{65}$$

Therefore,

$$\Psi_s = \Psi_{s_N - s_{N-1}} \circ \cdots \circ \Psi_{s_1 - s_0}. \tag{66}$$

B.10 Equivalence at the Positional Level

The previous theorem can also be expressed directly in terms of the positional increments.

[Additive positional accumulation] For any $t \in \mathbb{R}$, $s \geq 0$, and any partition

$$t = t_0 < t_1 < \cdots < t_N = t + s, \tag{67}$$

the total positional increment satisfies

$$L(t, t+s) = \sum_{i=0}^{N-1} L(t_i, t_{i+1}). \tag{68}$$

Proof. By definition,

$$\begin{aligned} \sum_{i=0}^{N-1} L(t_i, t_{i+1}) &= \sum_{i=0}^{N-1} A[\psi(t_{i+1}) - \psi(t_i)] \\ &= A \sum_{i=0}^{N-1} [\psi(t_{i+1}) - \psi(t_i)]. \end{aligned} \tag{69}$$

The summation telescopes:

$$\begin{aligned} &[\psi(t_1) - \psi(t_0)] + [\psi(t_2) - \psi(t_1)] + \cdots + [\psi(t_N) - \psi(t_{N-1})] \\ &= \psi(t_N) - \psi(t_0). \end{aligned} \tag{70}$$

Hence,

$$\begin{aligned} \sum_{i=0}^{N-1} L(t_i, t_{i+1}) &= A[\psi(t_N) - \psi(t_0)] \\ &= A[\psi(t+s) - \psi(t)] \\ &= L(t, t+s). \end{aligned} \tag{71}$$

B.11 Extension to a Group

The one-parameter monoid can be extended to a group by allowing both positive and negative elapsed times.

Let

$$\mathcal{S}_{\mathbb{R}} = \{\Psi_s \mid s \in \mathbb{R}\}. \tag{72}$$

Proposition 2 (Inverse evolution)**.** *For every $s \in \mathbb{R}$,*

$$\Psi_s^{-1} = \Psi_{-s}. \tag{73}$$

Proof. Using the composition law,

$$\Psi_{-s} \circ \Psi_s = \Psi_{-s+s} = \Psi_0 = \mathrm{Id}_{\mathcal{X}}, \tag{74}$$

and similarly,

$$\Psi_s \circ \Psi_{-s} = \Psi_{s-s} = \Psi_0 = \mathrm{Id}_{\mathcal{X}}. \tag{75}$$

Therefore,

$$\Psi_s^{-1} = \Psi_{-s}. \tag{76}$$

Theorem 3 (DCT positional one-parameter group)**.** *The family*

$$\mathcal{S}_{\mathbb{R}} = \{\Psi_s \mid s \in \mathbb{R}\} \tag{77}$$

equipped with function composition forms a commutative group.

Proof. Closure follows from

$$r + s \in \mathbb{R} \tag{78}$$

for all $r, s \in \mathbb{R}$, together with

$$\Psi_r \circ \Psi_s = \Psi_{r+s}. \tag{79}$$

Associativity follows from associativity of function composition. The identity element is Ψ_0. Finally, Proposition 2 establishes that every Ψ_s has inverse Ψ_{-s}. Therefore, $(\mathcal{S}_{\mathbb{R}}, \circ)$ is a group.

Furthermore,

$$\Psi_r \circ \Psi_s = \Psi_{r+s} = \Psi_{s+r} = \Psi_s \circ \Psi_r, \tag{80}$$

so the group is commutative.

B.12 Relation to the Normalized Observation Interval

Although the normalized observation interval in the proposed method is $[0, 1]$, the semigroup construction above is defined on $t \in \mathbb{R}$. This distinction is necessary for a strict semigroup formulation.

If the time variable were restricted to $t \in [0, 1]$, then for a given t and arbitrary $s \geq 0$, the state $t + s$ could lie outside $[0, 1]$. Consequently, the operator Ψ_s would not be defined for every state and every $s \geq 0$, and the closure required by a global one-parameter semigroup would be lost.

Therefore, we regard $[0, 1]$ as the normalized observation or training interval, while the DCT basis and the mathematical evolution operator are defined on the larger domain $\mathbb{R}$. Within the observed interval, the semigroup property applies to every composition for which the intermediate times remain in the observation interval.

B.13 Summary of the Algebraic Structure

The above results establish the following hierarchy.

First, the DCT positional representation

$$p(t) = A\psi(t) \tag{81}$$

induces the temporal increment

$$L(a, b) = p(b) - p(a), \tag{82}$$

which satisfies the additive cocycle relation

$$L(a, c) = L(a, b) + L(b, c). \tag{83}$$

Second, by augmenting the state with absolute time,

$$X = (x, t), \tag{84}$$

the temporal increment induces the one-parameter evolution operator

$$\Psi_s(x, t) = (x + A[\psi(t + s) - \psi(t)], t + s)\,. \tag{85}$$

Third, the evolution operators satisfy the classical one-parameter semigroup law

$$\Psi_r \circ \Psi_s = \Psi_{r+s}, \qquad r, s \geq 0, \tag{86}$$

together with

$$\Psi_0 = \mathrm{Id}. \tag{87}$$

Therefore,

$$(\mathcal{S}, \circ) \text{ is a commutative monoid, and hence a semigroup.} \tag{88}$$

If negative elapsed times are additionally admitted, the family extends to a one-parameter commutative group:

$$\Psi_s^{-1} = \Psi_{-s}. \tag{89}$$

Importantly, the semigroup property does not rely on a special addition formula of the DCT basis. Rather, it follows from the endpoint-difference structure

$$L(a, b) = A[\psi(b) - \psi(a)] \tag{90}$$

and the explicit inclusion of absolute time in the state. The DCT basis provides the temporal positional parameterization, while the augmented state construction converts its interval-additivity into a strict one-parameter semigroup of evolution operators.