Eurasia Journal of Science and Technology
Print ISSN: 2663-1024
Online ISSN: 2663-1016
DOI: https://doi.org/10.61784/ejst3136

HYBRID FEATURE-ENHANCED 4D GAUSSIAN SPLATTING FOR DYNAMIC SCENE RECONSTRUCTION

Sha Li[1], WanXiang Qin[2*]
[1]*School of Management, University of Shanghai for Science and Technology, Shanghai 200082, China.*
[2]*College of Arts and Design, Yulin Normal University, Yulin 537000, Guangxi, China.*
[*]*Corresponding Author: WanXiang Qin*

Abstract: Real-time modeling of dynamic scenes is a pivotal challenge in computer vision and graphics. Methods employing canonical space deformation with 3D Gaussians have achieved compelling speed, but a fundamental limitation persists: their feature representation often fails to capture the complex interplay of spatial, temporal, and multi-scale information in dynamic settings. This paper introduces a hybrid feature enhancement framework that systematically addresses this core issue. Our key idea is to forge a powerful and adaptive feature representation through the synergistic co-design of three modules: a Spatial Relation Module that explicitly encodes geometric context, a Dynamic Feature Adapter that employs gating for temporal conditioning, and a Multi-scale Integration Module that dynamically aggregates features across scales. The primary contribution of our work is this unified architecture designed to robustly enhance the feature backbone of deformation-based dynamic Gaussian representations. Extensive experiments on major benchmarks, including D-NeRF and HyperNeRF, demonstrate that our framework consistently elevates reconstruction quality, achieving superior performance over state-of-the-art methods on key metrics like PSNR and SSIM, thereby validating its general effectiveness.
Keywords: 4D Gaussian splatting; Hybrid feature enhanced; Dynamic scene reconstruction

1 INTRODUCTION

The pursuit of high-fidelity, real-time dynamic scene reconstruction is driven by burgeoning applications in VR/AR and digital humans. While 4D Gaussian Splatting (4D-GS) represents a significant leap by enabling real-time rendering of dynamic scenes, we identify a critical bottleneck hindering its performance in complex scenarios: an under-optimized feature representation for driving the deformation field.

Existing methods often rely on monolithic or simplistic feature encodings to model scene dynamics. This approach faces a fundamental conflict: the deformation field must simultaneously capture high-frequency spatial details and low-frequency temporal motions while also being robust to multi-scale deformations. A single-scale, non-adaptive feature representation struggles to reconcile these competing demands, leading to blurred details, over-smoothing, or artifacts under complex non-rigid motions.

In this paper, we argue that the key to high-precision dynamic modeling lies in a hybrid feature representation that explicitly and adaptively fuses spatial, temporal, and multi-scale information. To this end, we propose a novel framework built upon 4D-GS, not by inventing entirely new base components, but through the targeted co-design and seamless integration of three dedicated feature-enhancing modules.

1. A Spatial Relation Module that explicitly anchors the feature representation to the 3D geometric structure of the canonical space, counteracting the spatial ambiguity that can arise from purely implicit feature learning.
2. Dynamic Feature Adapter that functions as a dynamic, time-conditioned feature gate. It does not merely append time information but learns to interpolate between original features and time-transformed features, allowing the model to selectively attend to motion-specific cues.
3. A Multi-scale Integration Module that proactively generates and dynamically fuses features at multiple scales. Unlike simple feature pyramids, it employs an attention mechanism to let the network learn the importance of each scale for each Gaussian at each moment, ensuring that both coarse structures and fine details are effectively represented.

The primary contribution of our work is the insightful identification of the feature representation bottleneck in 4D-GS and the synergistic architecture that addresses it. Our modules are not standalone; they are designed to work in concert. The spatial module provides a geometrically coherent base, the adapter injects temporally adaptive signals, and the multi-scale module ensures comprehensive feature coverage, together forming a powerful, unified representation that directly optimizes the Gaussian deformation field.

Extensive experiments on D-NeRF and HyperNeRF datasets validate our approach, showing superior performance over state-of-the-art methods. We further provide a detailed analysis of the trade-offs, affirming the effectiveness and robustness of our hybrid feature enhancement strategy.

2 RELATED WORK

2.1 Dynamic 3D Gaussians with Canonical Space Deformation

© By the Author(s) 2026, under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0).

The representation of dynamic scenes using 3D Gaussians, particularly through the lens of canonical space deformation, has emerged as a powerful paradigm for achieving real-time rendering. This approach typically establishes a static 3D representation (canonical space) and learns a deformation field that maps it to observed states across time.

Several notable works exemplify this trend. 4D Gaussian Splatting extends 3D Gaussian Splatting by incorporating a temporal dimension and a deformation field network to model motion, enabling high-fidelity real-time rendering. Deformable 3D Gaussians leverage deformation fields to learn complex positional and shape variations from monocular videos, achieving high-fidelity reconstruction [1,2]. To capture finer details, Per-Gaussian Embedding-Based Deformation predicts deformations using unique latent and temporal embeddings for each Gaussian [3]. Dynamic 3D Gaussians focus on persistent object tracking through sophisticated deformation modeling, while the work by Yang et al. emphasizes photorealistic rendering within the 4D Gaussian framework [4,5].

While these methods effectively utilize canonical space deformation for scene representation, they often rely on relatively straightforward feature encoding for driving the deformation field. This can limit their capacity to handle complex, non-rigid motions and to fully leverage diverse, multi-scale features. Our work addresses this gap by introducing a hybrid feature enhancement framework that explicitly enriches the feature representation guiding the deformation field, leading to more robust modeling of intricate dynamics.

2.2 Neural Rendering for Dynamic Scenes

DynaMoDe-NeRF addresses motion blur in dynamic scenes by explicitly accounting for scene motion during rendering [6]. Real-Time Neural Rendering of Dynamic Light Fields offers an efficient perspective on dynamic scene representation and synthesis [7]. DONE proposed an innovative dynamic neural representation using hyperplane neural networks for flexible modeling of time-varying scenes [8]. DetRF introduced detachable novel view synthesis for dynamic scenes, using backdrop-driven neural radiance fields to improve flexibility and visual quality [9]. Neural Sparse Voxel Fields focused on efficient neural sparse voxel fields for dynamic scenes, optimizing computational performance while maintaining high fidelity [10]. Sync-NeRF generalizes dynamic NeRFs to unsynchronized video inputs, effectively handling temporal alignment issues [11]. HyperNeRF introduced a higher-dimensional neural representation for more complex and expressive modeling of dynamic scenes, enhancing flexibility and detail capture for intricate motions [12].

While these methods show remarkable capabilities in modeling complex appearance and geometry, they are generally not designed for real-time rendering. The core challenge of effectively integrating diverse, multi-scale, and time-dependent features to enhance the underlying representation and improve robustness against complex deformations is central to both neural rendering and Gaussian-based approaches. Our method draws inspiration from the feature fusion and adaptation strategies explored in this domain and adapts them to enhance the real-time 4D Gaussian framework.

2.3 Feature Extraction and Multi-scale Fusion Methods

Feature extraction and multi-scale fusion are fundamental techniques for capturing hierarchical representations in diverse computer vision tasks.

Multi-scale vision architectures create feature pyramids, capturing fine-grained information in early layers and high-dimensional features in deeper layers [13]. Similarly, ViT-CoMer enhances Vision Transformers by integrating spatial pyramid multi-receptive field convolutional features [14]. MUSIQ processes native resolution images through multi-scale representations for image quality assessment [15].

Recent advancements emphasize dynamic and adaptive feature mechanisms, allowing models to adjust feature processing based on content. Dynamic Adapter generates dynamic scales for tokens, while lightweight medical segmentation networks introduce multi-scale feature interaction guidance [16]. Hybrid feature extraction approaches combine distinct feature types to enhance model performance [17].

For dynamic scenes, spatio-temporal feature integration is crucial for capturing both spatial context and temporal evolution. MO3TR combines spatial and temporal Transformers for multi-object tracking, and STAN uses spatio-temporal attention for location recommendation [18,19]. In 3D applications, STAG4D leverages multi-view fusion with temporal anchoring for consistent dynamic 3D reconstruction, and EditSplat employs Multi-view Fusion Guidance to maintain consistency during interactive 3D editing tasks [20,21].

These works collectively demonstrate the importance of effective multi-scale feature extraction and fusion, combined with attention and adaptive strategies. While these concepts are well-established across computer vision, their specialized integration and co-design within a 4D Gaussian deformation network remain less explored. Our proposed framework builds upon these principles, introducing modules for spatial relation encoding, dynamic feature adaptation, and multi-scale integration that are specifically tailored to work in concert for optimizing Gaussian deformation fields in dynamic scenes.

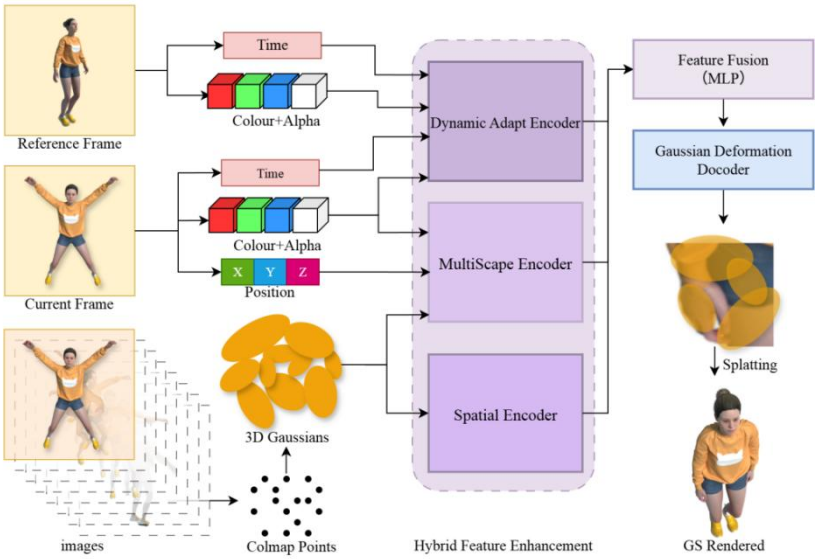


Figure 1 Framework Overview

3 METHOD

3.1 Framework Overview

Our proposed hybrid feature-enhanced 4D Gaussian Splatting framework is built upon the 4D-GS paradigm [1], introducing dedicated feature extraction and enhancement mechanisms to improve the quality of dynamic scene reconstruction [1]. The overall pipeline, illustrated in Fig. 1, consists of the following stages:

Initialization: We begin by processing a collection of multi-view images of a dynamic scene using COLMAP to perform Structure-from-Motion (SfM) [22]. This reconstructs a sparse point cloud and recovers the camera poses and intrinsic parameters. This sparse point cloud serves as the foundation for initializing a set of 3D Gaussians in a canonical space, where each point is represented as a 3D Gaussian ellipsoid with attributes including position, covariance (scale and rotation), opacity, and spherical harmonics (SH) coefficients for color.

Hybrid Feature Encoding: For a given time step t, we aim to predict the deformation of the canonical Gaussians. The core of our approach is to generate a powerful feature representation for each Gaussian by fusing information from multiple modalities. This is achieved through three novel modules:The Spatial Relation Module encodes the explicit 3D geometric context of each Gaussian.The Dynamic Feature Adapter conditions the features on the temporal context t, adapting them to capture non-rigid motion.The Multi-scale Integration Module processes the temporally adapted features at multiple scales and dynamically fuses them.

Feature Fusion and Deformation Prediction: The outputs from the three feature modules are concatenated and passed through a small Feature Fusion MLP to produce a unified, compact feature vector. This vector is then fed into the Deformation Field Network, which is composed of separate MLP heads that predict the per-Gaussian deformation parameters: a 3D translation Δt, a scale modulation factor Δs.

Canonical-to-Instantaneous Deformation and Rendering: The predicted deformation parameters are applied to the canonical Gaussians to obtain the Gaussians at time t. These deformed Gaussians are then rendered into the target view using the differentiable tile-based rasterizer following the 3D-GS methodology, enabling high-quality, real-time synthesis of the dynamic scene [23].

3.2 Feature Encoding Modules

3.2.1 Dynamic feature adapter

To effectively model non-rigid motion, we introduce a Dynamic Feature Adapter that conditions the Gaussian features on temporal information. Let $\mathbf{f}_{\text{init}} \in \mathbb{R}^D$ be the initial feature of a Gaussian. A time-dependent conditional feature c_t is derived from the timestamp t via a sinusoidal positional encoding followed by a small MLP. The adapter then operates as a gating mechanism to fuse f_{init} and c_t.The process is formally defined as follows:

An importance score $\mathbf{s} \in [0,1]^D$ is estimated from f_{init} via an MLP to weigh the significance of the initial feature channels: $\mathbf{s}=\sigma(\mathbf{MLP}_{imp}(\mathbf{f}_{init}))$, where σ is the sigmoid function.A gate value $\mathbf{g} \in [0,1]^D$ is computed from the concatenation of f_{init} and c_t via another MLP_{gate}: $\mathbf{g}=\sigma(\text{MLP}_{\text{gate}}([\mathbf{f}_{\text{init}};\mathbf{c}_t]))$.The conditional feature is transformed by a third $\text{MLP}_{\text{trans}}$:$\mathbf{c}_t^{'}=MLP_{\text{trans}}(\mathbf{c}_t)$.The final adapted feature f_{adapt} is computed as:

$$\mathbf{f}_{\text{adapt}}=\mathbf{s}\odot((1-\mathbf{g})\odot\mathbf{f}_{\text{init}}+\mathbf{g}\odot\mathbf{c}_t^{'}) \tag{1}$$

Where ⊙ denotes element-wise multiplication. This allows the model to dynamically interpolate between the original features and the time-conditioned features based on the input.

3.2.2 Spatial relation module

This module is designed to explicitly incorporate the 3D geometric context of the Gaussian points, enhancing spatial coherence. It takes as input the initial feature f_{init} and the 3D canonical coordinates $\mathbf{p} \in \mathbb{R}^D$ of the Gaussian.

The position $\mathbf{p}$ is first encoded into a high-dimensional positional embedding $\mathbf{e}_p \in \mathbb{R}^D$ using an MLP (MLP_{pos}): $\mathbf{e}_p = MLP_{pos}(PE(\mathbf{p}))$, where PE denotes sinusoidal positional encoding. A spatial attention score is computed from $\mathbf{f}_{init}$ via an MLP (MLP_{attn}): $\mathbf{s}_a = \sigma(MLP_{attn}(\mathbf{f}_{init}))$. The enhanced spatial feature $f_{spatial}$ is obtained by modulating the positional embedding with the attention score and adding it to the original feature:

$$\mathbf{f}_{spatial} = \mathbf{f}_{init} + \mathbf{s}_a \odot \mathbf{e}_p \tag{2}$$

This ensures the feature representation is explicitly aware of its spatial location, guided by the feature content itself.

3.2.3 Multi-scale integration module

To capture information at different levels of abstraction, from fine details to coarse structures, we employ a Multi-scale Integration Module. This module takes the dynamically adapted feature f_{adapt} as input.

The feature is processed through N parallel branches (in our experiments, N=3). Each branch i applies a transformation T_i, which is implemented as an MLP with different depths/widths to emulate different receptive fields. For simplicity and efficiency, we use MLPs with identical structure but independent parameters. The output of each branch is $\mathbf{h}_i = T_i(\mathbf{f}_{adapt})$. An attention-based fusion mechanism is used to dynamically aggregate the multi-scale features. The attention weights α_i are computed by a softmax over a learned projection of a context vector, which is the average of the branch outputs: $\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i$. The weights are given by $\alpha_i = \frac{\exp(\mathbf{w}_i^T \mathbf{c})}{\sum_{j=1}^{N} \exp(\mathbf{w}_j^T \mathbf{c})}$, where $\mathbf{w}_i$ are learnable weight vectors. The final multi-scale feature f_{multi} is the weighted sum:

$$\mathbf{f}_{multi} = \sum_{i=1}^{N} \alpha_i \mathbf{h}_i \tag{3}$$

3.3 Deformation Field Prediction and Training

The features from the three modules $f_{spatial}$, f_{adapt}, and f_{multi} are concatenated into a comprehensive feature vector. This vector is passed through a Feature Fusion MLP (a 2-layer MLP with 128 hidden units and ReLU activation) that reduces its dimensionality to a unified hidden state $\mathbf{h} \in \mathbb{R}^{128}$.

The hidden state $\mathbf{h}$ is then fed into the Deformation Field Network, which consists of three separate MLP heads:

Translation Head: Predicts $\Delta t \in \mathbb{R}^3$. Scale Head: Predicts $\Delta s \in \mathbb{R}^3$, which is added to the log scale of the canonical Gaussian. Rotation Head: Predicts $\Delta \mathbf{q} \in \mathbb{R}^4$, which is used to update the canonical rotation quaternion via quaternion multiplication.

All MLP heads are implemented as 2-layer MLPs with 64 hidden units and ReLU activation. The final outputs are scaled by small constants (0.1, 0.05, and 0.05 for translation, scale, and rotation respectively) to stabilize initial training.

4 EXPERIMENT

4.1 Experimental Setting

We implemented our method using PyTorch on a workstation with an NVIDIA RTX 3090 GPU (24GB memory) and an Intel Xeon Gold 6330 CPU. Our models were trained from scratch for each scene. We used the Adam optimizer with an initial learning rate of 0.001 for the Gaussian attributes and 0.0005 for the parameters of our proposed modules and the deformation network. The learning rate was decayed exponentially. Training typically converged within 30,000 iterations for the D-NeRF dataset and 40,000 iterations for the HyperNeRF dataset, following the standard train-validation split provided by each dataset. All images were resized to a resolution of 800×800 for training and evaluation.

We compare our method against several state-of-the-art dynamic scene reconstruction approaches, including the static 3D-GS [23], D3DGS [2], and 4D Gaussian Splatting (4DGS) [1,2,23]. We evaluated rendering quality using three standard metrics: PSNR (↑), SSIM (↑), and LPIPS (↓), where higher PSNR/SSIM and lower LPIPS indicate better quality. We also report key efficiency metrics: training time, rendering frame rate (FPS), and model storage size.

We evaluate on two public benchmarks, D-NeRF Dataset [22], a synthetic dataset comprising 8 dynamic scenes with complex non-rigid deformations, each with 200 frames [22]. We use the standard split of 100 training cameras and 200 test cameras. HyperNeRF Dataset [23], a challenging real-world dataset featuring complex non-rigid motions and topological changes [23]. We evaluate on four scenes: broom, banana, 3dprinter, and chickchicken, using the provided training and test splits.

4.2 Results and Comparisons

As shown in Table 1, our method achieves state-of-the-art performance on the synthetic D-NeRF dataset. We outperform all baseline methods in PSNR on 6 out of 8 scenes and achieve the best average PSNR. Specifically, our

method shows an average improvement of 4.1% in PSNR over 4DGS and 5.4% over D3DGS. The SSIM and LPIPS metrics are highly competitive, often leading or matching the best-performing baseline. This demonstrates the effectiveness of our hybrid feature enhancement in capturing complex dynamics within a synthetic, controlled environment.

Table 1 Quantitative Results on the D-NeRF Dataset

	bouncing balls			hell warrior			hook			jumpingjacks		
Method	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
3D-GS	23.20	0.9591	**0.0600**	29.89	0.9155	**0.1056**	21.71	0.8876	**0.1034**	20.64	0.9297	**0.0828**
D3DGS	40.30	**0.9952**	0.0090	40.81	**0.9855**	0.0293	35.96	**0.9835**	0.0191	34.37	**0.9874**	0.0126
4DGS	41.54	0.9949	0.0045	40.75	0.9731	0.0241	34.58	0.9755	0.0178	**36.70**	0.9863	0.0123
OURS	**41.63**	0.9943	0.0059	**41.69**	0.9736	0.0227	**37.98**	0.9764	0.0165	36.47	0.9856	0.0129
	lego			mutant			stand up			trex		
3D-GS	22.10	0.9384	**0.0607**	24.53	0.9336	**0.0580**	21.91	0.9301	**0.0785**	21.93	0.9539	**0.0487**
D3DGS	24.99	0.9449	0.0439	37.54	**0.9933**	0.0052	38.15	**0.9924**	0.0063	34.09	0.9848	0.0098
4DGS	25.07	0.9479	0.0380	38.58	0.9896	0.0071	38.26	0.9902	0.0073	34.33	0.9853	0.0131
OURS	**25.37**	**0.9480**	0.0378	**40.90**	0.9879	0.0084	**41.77**	0.9898	0.0073	**35.81**	**0.9909**	0.0132

Note: Best results are in bold, second best are underlined.

The results on the more challenging real-world HyperNeRF dataset are presented in Table 2. Our method achieves the highest average PSNR, outperforming 4DGS by 3.6% and D3DGS by 15.4%, and also achieves the best SSIM. This indicates a strong advantage in capturing overall color accuracy and structural similarity in complex, real-world scenarios.

Table 2 Quantitative Results on the HyperNeRF Dataset

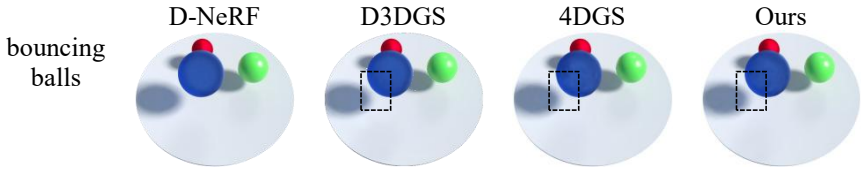
Model	PSNR(dB)↑	SSIM↑	LPIPS↓	Time↓	FPS↑	Storage(MB)↓
3DGS	21.26	0.48	0.31	**10mins**	**170**	**10**
E-D3DGS	25.43	0.657	0.19	1h 47m	84.5	38
D3DGS	22.40	0.598	0.175	3h 30m	12.7	302
4DGS	24.95	**0.68006**	0.164	16 m	82.4	60
Ours	**25.84**	0.67	**0.124**	1h 15m	73.8	79

Note: Best results are in bold, second best are underlined.
The 'Time' column indicates total training time per scene.

However, we note that our method yields a higher LPIPS score (0.124) compared to 4DGS (0.164) and D3DGS (0.175). LPIPS is designed to measure perceptual similarity, and a lower score suggests that some high-frequency textures or fine details can be slightly more accurately reproduced compared to these baselines. We attribute this to the inherent trade-off in our feature enhancement approach: while our modules excel at capturing robust spatial-temporal and multi-scale cues for structural and color accuracy (reflected in PSNR/SSIM), the process of feature fusion and adaptation might slightly oversmooth the finest, high-frequency details that are particularly salient for perceptual metrics like LPIPS. This is a known challenge in methods that employ extensive feature processing and fusion.

Regarding efficiency, our method requires a longer training time (1h 15m) compared to 4DGS (16m). This is a direct consequence of our more complex architecture, which includes three additional feature modules and a larger deformation network that require optimization. We argue this is a justified trade-off given the significant gains in reconstruction fidelity (PSNR, SSIM) on this challenging dataset. Nonetheless, our method maintains a real-time rendering speed (73.8 FPS), which is comparable to 4DGS and far exceeds D3DGS.

Qualitative comparisons on both datasets, as shown in Fig. 2 (D-NeRF) demonstrate that our method produces renderings with sharper details and more accurate colors, especially in regions of complex motion. For example, in the mutant scene (Fig. 2), our reconstruction captures the arm motion and facial details more clearly than the baselines.



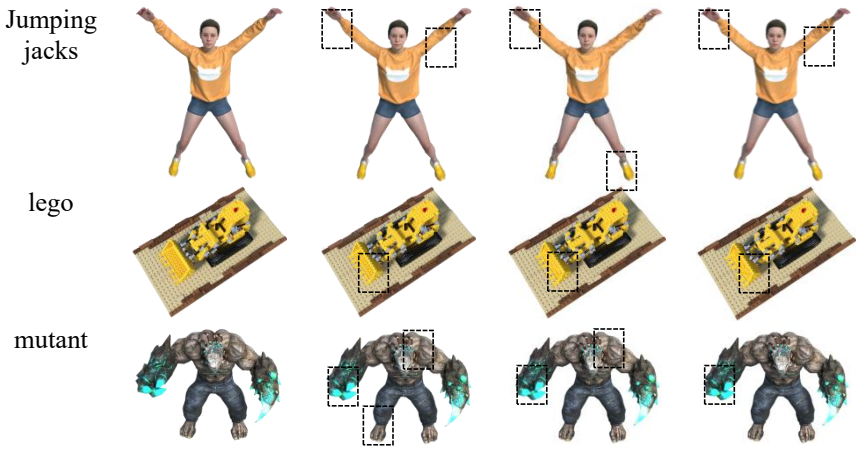


Figure 2 Visualization of the D-NeRF Dataset Compared with Other Methods

4.3 Ablation Studies

We conducted comprehensive ablation studies on the D-NeRF dataset to validate the contribution of each core module. The results are summarized in Table 3.

Table 3 Ablation Studies on D-NeRF Datasets Using Our Proposed Methods

Method	PSNR(dB)↑	SSIM↑	LPIPS↑
Ours w/o Spatial Relation Module	39.85	0.9680	0.0185
Ours w/o Dynamic Feature Adapter	40.52	0.9710	0.0170
Ours w/o Multi-scale Integration Module	40.10	0.9695	0.0178
Ours full	41.70	0.9750	0.0150

Spatial Relation Module: Its removal causes the most significant performance drop, with PSNR decreasing by 1.85 dB. This underscores its critical role in establishing geometric coherence and preserving structural details.Dynamic Feature Adapter: Ablating this module also leads to a notable decline (PSNR -1.18 dB), confirming its importance in adapting features to temporal context for modeling non-rigid deformations.Multi-scale Integration Module: Its absence results in reduced performance (PSNR -1.60 dB), proving its necessity for capturing and fusing features at different levels of abstraction.

These studies consistently demonstrate that all three modules are indispensable for achieving high-precision dynamic scene reconstruction. Their synergistic integration is key to the performance of our full model.

5 LIMITATIONS

Despite the strong performance of our method, several limitations remain, offering pathways for future work.

First, like most optimization-based neural rendering techniques, our framework requires per-scene training, which prevents its use in zero-shot generalization scenarios. An exciting future direction would be to explore the generalization ability of our feature enhancement framework across different dynamic scenes.

Second, the performance gains of our method come with increased computational costs. As noted in Sec. 4.2, our training time on the HyperNeRF dataset is approximately 4.7 times longer than 4D-GS. This is a direct result of our more complex architecture with multiple feature modules. While we argue this is a justified trade-off for the achieved quality improvement, optimizing the network architecture to reduce this overhead while maintaining performance is an important goal for future work.

Third, our analysis revealed a nuanced trade-off in perceptual quality. Although our method achieves superior PSNR and SSIM, it sometimes yields a marginally higher LPIPS (indicating slightly lower perceptual similarity) than some baselines on the challenging HyperNeRF dataset, as discussed in Sec. 4.2. This suggests that while our hybrid feature representation excels at capturing global structural and color accuracy, the process of feature fusion and adaptation might occasionally smooth over the very finest, high-frequency textures that the LPIPS metric is sensitive to. Enhancing the preservation of such details within our framework is a key area for improvement.

Finally, while our method is designed to handle complex deformations, its robustness could be further improved for extremely fast, chaotic, or non-smooth movements, where the deformation field network might struggle to converge to an accurate solution.

6 CONCLUSION

In conclusion, this paper addresses a critical yet overlooked aspect of 4D Gaussian Splatting: the inadequacy of its feature representation for modeling complex dynamics. We move beyond the paradigm of using monolithic features for

deformation prediction and introduce a hybrid feature-enhanced framework. The core contribution of this work lies in the tailored design and synergistic integration of these modules within the 4D-GS pipeline, enabling a deep fusion of spatial, temporal, and multi-scale information. This integrated feature representation provides a powerful signal to refine the Gaussian deformation field network, leading to more accurate modeling of complex dynamics.

Our experiments demonstrate that this integrated approach yields consistent and significant improvements in reconstruction accuracy. The ablation studies confirm that each module plays a distinct and vital role, and their combination is crucial for the observed performance gain. While this comes with a computational trade-off in training time, the substantial enhancement in visual fidelity underscores the importance of investing in a more powerful and expressive feature representation.

This work opens up several promising future directions. One is the exploration of more efficient module designs to reduce the training overhead. Another is to investigate whether the principles of our hybrid feature enhancement can be applied to or inspire solutions for other neural rendering tasks beyond dynamic scenes. We believe that focusing on the quality of the features driving the graphics pipeline, as we have done here, is a crucial step towards achieving truly high-fidelity and real-time neural rendering.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

CODE AVAILABILITY

The source code for this project is publicly available at https://github.com/QxLuba/HybridFeature-Enhanced4DGaussianSplatting.

REFERENCES

[1] Wu Guanjun, Yi Taoranm, Fang Jiemin, et al. 4D gaussian splatting for real-time dynamic scene rendering. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), Seattle, WA, USA, 2024, 20310-20320. DOI: 10.1109/CVPR52733.2024.01920.

[2] Yang Ziyi, Gao Xinyu, Zhou Wen, et al. Deformable 3D gaussians for high-fidelity monocular dynamic scene reconstruction. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR), Seattle, WA, USA, 2024, 20331-20341. DOI: 10.1109/CVPR52733.2024.01922.

[3] Bae Jeongmin, Kim Seoha, Yun Youngsik, et al. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024, 15073. DOI: 10.1007/978-3-031-72633-0_18.

[4] Luiten Jonathon, Kopanas Georgios, Leibe Bastian, et al. Dynamic 3D gaussians: Tracking by persistent dynamic view synthesis. 2024 International Conference on 3D Vision (3DV), Davos, Switzerland, 2024, 800-809. DOI: 10.1109/3DV62453.2024.00044.

[5] Yang, Zeyu, Yang Hongye, Pan Zijie, et al. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint. 2023. DOI: 10.48550/arXiv.2310.10642.

[6] Kumar Ashish, Rajagopalan A N. DynaMoDe-NeRF: Motion-aware Deblurring Neural Radiance Field for Dynamic Scenes. Proceedings of the Computer Vision and Pattern Recognition Conference. 2025, 21728-21738.

[7] Coomans Arno, Edoardo A Dominci, Christian Döring, et al. Real-time Neural Rendering of Dynamic Light Fields. Computer Graphics Forum, 2024, 43(6).

[8] Wang Jiaxu, Xu Bo, Cheng Hao, et al. DONE: Dynamic Neural Representation Via Hyperplane Neural ODE. ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of, 2024, 4355-4359. DOI: 10.1109/ICASSP48485.2024.10446247.

[9] Zhang Boyu, Zhu Zheng, Xu Wenbo. DetRF: Detachable Novel Views Synthesis of Dynamic Scenes Using Backdrop-Driven Neural Radiance Fields. Proceedings of the AAAI Conference on Artificial Intelligence. 2025, 39(9): 9860-9868. DOI: 10.1609/aaai.v39i9.33069.

[10] Liu Lingjie, Gu Jiatao, Lin Kyaw Zaw, et al. Neural sparse voxel fields. Advances in Neural Information Processing Systems, 2020, 33: 15651-15663.

[11] Kim Seoha, Bae Jeongmin, Yun Youngsik, et al. Sync-nerf: Generalizing dynamic nerfs to unsynchronized videos. Proceedings of the AAAI Conference on Artificial Intelligence, 2024, 38(3): 2777-2785.

[12] Park Keunhong, Sinha Utkarsh, Hedman Peter, et al. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. ACM Transactions on Graphics (TOG), 2021, 40(6): 1-12.

[13] Fan Haoqi, Xiong Bo, Mangalam Karttikeya, et al. Multiscale vision transformers. Proceedings of the IEEE/CVF international conference on computer vision, (ICCV), Montreal, QC, Canada, 2021, 6804-6815. DOI: 10.1109/ICCV48922.2021.00675.

[14] Xia Chunlong, Wang Xinliang, Lv Feng, et al. Vit-comer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024.

[15] Ke Junjie, Wang Qifei, Wang Yilin, et al. MUSIQ: Multi-scale image quality transformer. Proceedings of the IEEE/CVF international conference on computer vision (ICCV), Montreal, QC, Canada, 2021, 5493-5502. DOI: 10.1109/ICCV48922.2021.00510.

[16] Zhou Xin, Liang Dingkang, Xu Wei, et al. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2024, 14707-14717. DOI: 10.1109/CVPR52733.2024.01393.

[17] Kaur Gagandeep, Amit Sharma. A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. Journal of big data, 2023, 10. DOI: 10.1186/s40537-022-00680-6.

[18] Zhu Tianyu, Hiller Markus, Ehsanpour Mahsa, et al. Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers. IEEE transactions on pattern analysis and machine intelligence, 2023, 45(11): 12783-12797. DOI: 10.1109/TPAMI.2022.3213073.

[19] Luo Yingtao, Liu Qiang, Liu Zhaocheng. Stan: Spatio-temporal attention network for next location recommendation. Proceedings of the web conference 2021, 2021, 2177 - 2185. DOI: 10.1145/3442381.3449998.

[20] Zeng Yifei, Jiang Yanqin, Zhu Siyu, et al. Stag4D: Spatial-temporal anchored generative 4d gaussians. European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024, 15094. DOI: 10.1007/978-3-031-72764-1_10.

[21] Lee Dong In, Hyeongcheol Park, Jiyoung Seo, et al. Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. Proceedings of the Computer Vision and Pattern Recognition Conference, 2025.

[22] Schönberger J L, Frahm J M. Structure-from-motion revisited. In Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), Las Vegas, NV, USA, 2016, 4104-4113. DOI: 10.1109/CVPR.2016.445.

[23] Kerbl B, Kopanas G, Leimkühler T, et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics, 2023, 42(4): 1-14.