TG4MM: Time-Varying Gaussian Splatting for 3D Motion Magnification

Zheng Zhang†, Jiabao Guo†, Fei Wang, Jinyang Huang, Zhi Liu*, Dan Guo*

***Abstract*—3D motion magnification aims to enable us to visualize subtle, imperceptible motions by integrating eulerian video magnification with novel view synthesis. Existing method extracts the variation of feature embeddings using Neural Radiance Fields (NeRF) over time. However, this volume rendering technique suffers from two shortcomings for 3D motion magnification: (1) When reconstructing time-varying scenes through volume rendering, spatial-temporal operations between static and dynamic representations often generate noticeable artifacts, leading to blurred magnified frames. (2) When processing high-resolution dynamic scenes, the intrinsically low rendering efficiency of these techniques causes excessive computational latency, preventing real-time visualization. In this work, instead of NeRF, we propose a novel *Time-varying Gaussian Splatting for 3D Motion Magnification* (TG4MM) that is capable of achieving real-time rendering while effectively handling blurred magnified frames in dynamic 3D motion magnification scenes. Specifically, we propose a motion-space decoupled triplane modeling approach. The space triplane captures major spatial structures from the first frame, while the motion triplane captures subtle motion information from subsequent frames. Furthermore, we develop a phase-based motion magnification module that enhances subtle motions by applying filters within the embedding space and subtle motion triplane. Experimental results demonstrate the effectiveness of our method, showing that it outperforms existing 3D motion magnification techniques and achieves a speed up to 126 FPS.**

***Index Terms*—3D Motion Magnification, Gaussian Splatting.**

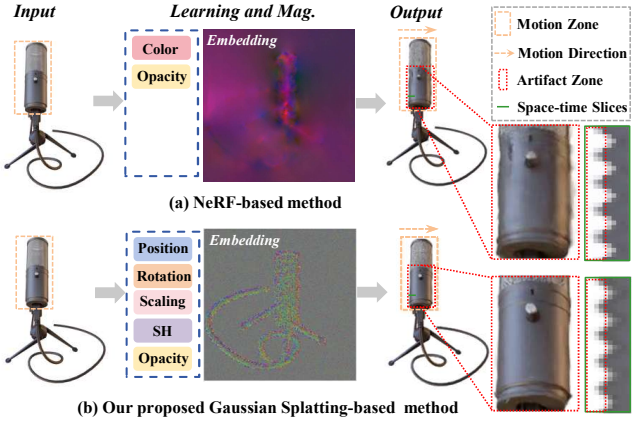


Fig. 1. **Comparison with existing 3D motion magnification method.** We propose a Gaussian Splatting-based approach for 3D motion magnification and compare it against existing NeRF-based methods [23]. (a) NeRF-based method [23] encodes point embeddings using only color and opacity, which limits its ability to capture complete 3D motion information and often results in blurriness and artifacts. In the artifact zone, the NeRF-based method causes a visible dent on the upper-left side and introduces blur and artifact accumulation on the right. In the space-time slices, the artifacts generated by this method tend to concentrate near the left boundary; (b) Our method leverages Gaussian Splatting to extract multi-dimensional attributes such as position, rotation, scale, and spherical harmonics, enabling clearer representation of shapes and contours in the embedding, and thereby allowing more effective representation and magnification of subtle motions in 3D space.

I. Introduction

THE world we live in is filled with subtle motions that contain rich physical and physiological information, and these signals have been applied across numerous fields, such as micro-expression recognition [1]–[3], arterial pulse detection [4], ultrasonic imaging analysis [5], and material property estimation [6]–[8]. Motion magnification techniques can further amplify subtle motion signals, enabling human visual perception of motion variations that would otherwise require sophisticated physical instrumentation for detection. However, existing methods [9]–[22] are confined to 2D images and thus cannot fully capture the subtle deformations and motions present in 3D space. In view of the inherently 3D nature of the real world, 3D motion magnification techniques have emerged. 3D motion magnification aims to visually enhance imperceptible subtle motions in the real world through novel view synthesis and Eulerian video magnification, while it supports observation of the amplified motion details from arbitrary viewpoints [23].

Admittedly, efficient and novel viewpoint rendering remains a major challenge in 3D motion magnification. Although Neural Radiance Fields (NeRF)-based methods have demonstrated effective solutions, it is observed that their point embeddings are constrained to color and opacity attributes, which are inadequate for comprehensively capturing 3D motion information. Consequently, the embedding representations frequently appear blurred, with indistinct object contours. In contrast, Gaussian Splatting incorporates additional attributes such as position, rotation, scale, spherical harmonics, and opacity as a richer representation for subtle motion. These attributes allow for accurate encoding of object shapes and contours in the embedding space. As illustrated in Fig. 1, NeRF-based point embeddings often do not represent the shape of the object

This work is supported by National Key R&D Program of China (NO.2024YFB3311600), Natural Science Foundation of China (62272144), the Anhui Provincial Natural Science Foundation (2408085J040), and the Major Project of Anhui Provincial Science and Technology Breakthrough Program (202423k09020001), and the Fundamental Research Funds for the Central Universities (JZ2024HGTG0309, JZ2024AHST0337).

Zheng Zhang† and Jiabao Guo† contributed equally to this work. Dan Guo* and Zhi Liu* are the corresponding authors. Zheng Zhang, Jiabao Guo, Fei Wang, Jinyang Huang and Dan Guo are with the Anhui Province Key Laboratory of Affective Computing and Advanced Intelligence Machine and School of Computer and Information, Hefei University of Technology, Hefei, 230601, China (e-mail: 2024110476@mail.hfut.edu.cn, garbo_guo@hfut.edu.cn, jiafei127@gmail.com, hjy@hfut.edu.cn, guodan@hfut.edu.cn). Zhi Liu is with the University of Electro-Communications, 182-8585, Japan (email: liuzhi@uec.ac.jp).

clearly, resulting in blurred appearance and visual artifacts in the magnified images. In contrast, Gaussian Splatting-based point embeddings better preserve the shape and outline of objects, reducing blur and suppressing artifacts effectively.

In this paper, we propose TG4MM, a 3D motion magnification method based on Gaussian Splatting. Our method captures rich representations including the position, rotation, scale, spherical harmonics coefficients, and opacity of Gaussians in 3D space, which provide sufficient information for perceiving and amplifying subtle motions. Inspired by the success of dynamic 3D Gaussian Splatting [24]–[29] in novel view synthesis (NVS), we recognized its potential for encoding 3D motion. Through empirical analysis, we observed that Gaussian embeddings offer more accurate representations of object shapes and contours. As a result, our method not only improves the visual quality of motion magnification but also achieves an order-of-magnitude speedup in rendering. Specifically, we decouple the spatial structure and subtle motion information into two dedicated tri-plane representations, enabling effective modeling of dynamic variations across timestamps. During the magnification phase, we design a phase-based motion magnification module that operates on the embedded representation in Gaussian Splatting. This module extracts high- and low-frequency components from the embedding and outputs them directly to preserve spatial structures and texture details. It then applies frequency filtering to isolate the target motion frequency, amplifies it with a user-defined factor, and fuses it with the original frequency components to generate the magnified embedding. Finally, we introduce a specialized decoder head that accurately reconstructs the updated embedding into Gaussian rasterization parameters, enabling motion magnification while maintaining spatial consistency and rendering quality.

Overall, our main contributions are as follows:

- We propose the first 3D motion magnification approach based on Gaussian Splatting named TG4MM. It leverages the multidimensional properties of Gaussians to realize real-time rendering while effectively handling blurred magnified frames in dynamic 3D motion magnification scenes.
- We propose a motion-space decoupled triplane modeling module. The space triplane captures major spatial structures from the first frame, while the motion triplane captures subtle motion information from subsequent frames.
- We design a phase-based motion magnification module that can be embedded into the Gaussian Splatting representation, effectively amplifying the subtle motions captured by the motion-space decoupled tri-plane module.
- Extensive experiments show that TG4MM avoids artifacts and blurring in qualitative evaluation, achieves promising performance in quantitative evaluation. It significantly reduces model complexity and training time, boosting rendering speed up to 126 FPS.

II. Related Work

A. Video motion magnification

Existing 2D motion magnification algorithms can be broadly categorized into Lagrangian motion magnification [9]–[11], Eulerian motion magnification [12]–[17], and learning-based motion magnification [18]–[22].

Lagrangian motion magnification tracks the motion trajectory of individual particles, achieving subtle motion amplification through optical flow. These methodologies were initially developed by Liu *et al.* [9] through the implementation of global video registration techniques to mitigate camera shake, subsequently applying explicit optical flow calculations to enhance the motion signals. The Global Lagrangian Motion Magnification (GLMM) [10] employs Principal Component Analysis (PCA) to isolate dominant displacement components while suppressing noise. Continuous Amplitude Modulation-based Lagrangian framework [11] utilizes dense optical flow estimation to model motion field, enabling precise extraction and enhancement of subtle motions. For conducting motion magnification in 3D scenes, we first need to reconstruct the scene in 3D with Gaussian points. Subsequently, to amplify the 3D motion of Gaussian points, we need to process the motion features of these points. However, traditional Lagrangian methods based on 2D optical flow cannot directly handle motion features of Gaussian points. To address the incompatibility with traditional Lagrangian methods, we need to encode Gaussian points into embedding features and process their motion by phase.

Eulerian motion magnification analyzes changes at fixed pixel locations, further divided into Eulerian linear motion magnification [12] and Eulerian phase-based motion magnification [13], [14] based on the type of change. Linear motion magnification methods begin by decomposing the image sequence into a Gaussian pyramid, enabling color variations to be amplified at multiple spatial scales [12], [13], [15]. In contrast, phase-based motion magnification algorithms decompose images using a complex steerable pyramid or Riesz pyramid [13]–[16], followed by temporal filtering of the phase signals at each scale and spatial location to detect and amplify subtle periodic or aperiodic motions. To enhance user interactivity and enable manual selection of regions where subtle variations should be amplified, several layer-based motion magnification approaches have been proposed. Verma *et al.* [30] developed a framework combining kernel K-means clustering with automated scribble generation via superpixels and Bézier curves, followed by alpha matting and Eulerian magnification. Elgharib *et al.* [31] introduced a temporal alignment approach for selective amplification of user-specified layers, while Kooij *et al.* [32] proposed a depth-aware bilateral filter enabling region selection within consistent depth planes. The embedding features for motion magnification in TG4MM contain multidimensional information such as 3D position, rotation, and color. However, Lagrangian motion magnification relies solely on 2D optical flow [9]–[11], leaving it unable to process this multidimensional information. In contrast, phase-based motion magnification can directly perceive motion information across different frequency bands, making it more suitable for the technical requirements of this study. Overall, our proposed TG4MM is based on a Lagrangian perspective to amplify the trajectories of 3D Gaussian points. Aiming at the inherent limitation of traditional Lagrangian methods relying on 2D optical flow, we have further designed TG4MM that

combines embedding features and phase.

Learning-based motion magnification employs deep neural networks to automatically learn motion decomposition and amplification filters from data, overcoming limitations of handcrafted approaches. Oh *et al.* [18] pioneered an encoder-manipulator-decoder architecture trained on synthetic data with two-frame inputs and regularization to disentangle shape and texture. Singh *et al.* introduced lightweight real-time implementations [19] and a multi-domain framework combining frequency-domain motion simulation with spatial denoising through inter-frame phase difference analysis [20]. Recent advancements include EulerMormer [21], the first Transformer-based model featuring sparse dynamic attention and multi-scale gating for noise suppression, and FD4MM [22] with its frequency-decoupled multi-level isomorphic architecture that captures both high-frequency details and stable low-frequency motion patterns. These developments demonstrate a trend toward more sophisticated models balancing computational efficiency with multi-domain processing capabilities.

Conventional 2D motion magnification methods are inherently limited to planar motion analysis within image space, while geometrically-aware 3D approaches overcome these constraints through explicit scene reconstruction and multi-view geometric modeling. By incorporating camera geometry or multi-view information, 3D motion magnification enables three-dimensional scene reconstruction and spatially consistent motion amplification [23]. This methodology naturally aligns with point-based rendering paradigms like Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) through temporal particle tracking. However, extending Lagrangian motion analysis to 3D requires dense 3D optical flow estimation - a computationally demanding task with ongoing research challenges in accuracy and efficiency [16], [33]–[35]. As a learning-based framework, 3D motion magnification reconstructs complete 3D scenes from limited input images and camera parameters, enabling spatially coherent amplification of subtle motions in three-dimensional space.

By introducing NeRF, the first 3D motion magnification method [23] integrates novel view synthesis with Eulerian phase-based motion magnification by constructing a temporally varying point embedding function within the NeRF framework. Specifically, a static NeRF is first trained to capture the scene's geometry and appearance, with its MLP parameters fixed. For each dynamic observation at a given time step, the embedding function is fine-tuned to capture subtle motions, ensuring that only dynamic content is encoded—excluding camera motion and static structures [23]. In the magnification phase, temporal embedding sequences of 3D points undergo frequency-domain analysis; target frequencies are extracted using band-pass filtering and amplified via linear or phase-based modulation. While this approach demonstrates the feasibility of NeRF-based 3D motion magnification, it suffers from slow rendering and blurred magnified frames. To address these limitations, we adopt the state-of-the-art 3D Gaussian Splatting [36] in place of NeRF, achieving an 134× speed-up in rendering and significantly improving magnification accuracy.

B. Novel View Synthesis

Novel View Synthesis (NVS) refers to generating images of a scene from unseen viewpoints, given a set of images and their corresponding camera poses. Current methods for novel view rendering primarily include NeRF [37] and 3DGS [36]. NeRF encodes image and pose information through implicit neural networks, while 3DGS explicitly represents the scene as a set of 3D Gaussian points, offering an explicit approach to modeling 3D scene geometry. This technique enables high-fidelity, real-time rendering of scenes captured from multiple viewpoints by discretizing the scene into a collection of 3D Gaussian ellipsoids, each characterized by a covariance matrix derived from rotation and scaling transformations.

Recent advances in dynamic 3D Gaussian Splatting (3DGS) have been categorized based on input requirements where multi-view video inputs provide dense spatial supervision but face limitations in scene coverage while monocular or sparse multi-view inputs introduce greater reconstruction complexity due to limited viewpoints [38]. For multi-view video sequences early approaches [24], [25] optimized dynamic 3DGS through frame-wise reconstruction with physical priors such as local rigidity and rotational consistency but remained constrained to regions visible in initial frames. Subsequent methods like SWinGS [26] addressed this issue by introducing flow-guided sliding window partitioning combined with dynamic MLP optimization to enforce temporal consistency across 4DGS representations [23]. Optical flow-based supervision [39] has also emerged as a promising direction to refine motion estimation under multi-view settings. In contrast monocular or sparse multi-view approaches [27], [28] which better align with consumer device capabilities typically follow a two-stage pipeline first reconstructing static 3DGS foundations before modeling temporal deformations. While some methods employ MLPs to predict position, rotation, and scale changes others like 4D-GS [29] leverage HexPlane [40], [41] representations for joint spatial-temporal encoding. Further improvements include ST-4DGS [42] which introduces spatio-temporal regularization mechanisms and 4K4D [43] that enhances rendering quality through differentiable depth peeling algorithms. Alternative strategies [44], [45] for sparse inputs include polynomial trajectory fitting and Gaussian-Flow's Dual-Domain Deformation Model [46] which explicitly models attribute deformations using Fourier series. Despite these methodological advances both paradigms face common challenges due to the lack of standardized multiview datasets capturing subtle motions which hampers progress in 3D motion magnification evaluation and development. LoopGaussian [47] generates loop videos in 3D scenes by combining Eulerian motion fields.

Building upon 3D Gaussian Splatting, our method extends the framework to the temporal domain through a HexPlane-based spatiotemporal representation. We employ six orthogonal planes to explicitly encode spatial structures and subtle motion patterns, enabling effective 3D motion magnification by leveraging temporal coherence in dynamic deformations. This framework achieves amplified motion visualization while maintaining geometric consistency over time.

III. Method

3D subtle motion magnification is defined as a technique designed to visualize subtle motions that goes beyond the limitations of prior 2D motion magnification methods, and it can magnify subtle motions from scenes taken by a moving camera while supporting novel view rendering. Inputs of 3D motion magnification are static spatial image $I_{GT}(p, 0)$, where p is spatial position (x, y, z) and 0 is the initial static moment. The image of subtle motion $I_{GT}(p, t)$, where t is the time dimension and $t > 0$ corresponds to moments with subtle motions. Model outputs the motion-amplified image $I_m(p, t)$ under any view. The goal is to balance the motion amplification effect and the quality of the image for different magnification factors α. Fig. 2 shows the overall pipeline of our proposed TG4MM, which reconstructs subtle motions through space reconstruction and subtle motion learning. Initially, sparse point clouds were recovered from multiview images using Structure-from-Motion (SfM) [48]. In space reconstruction, these points are projected onto a tri-plane representation, where bilinear interpolation generates three spatially consistent 2D feature planes. The concatenated feature planes form an embedding vector $\delta_r(p, 0)$. The Gaussian decoder outputs the spatial representation $\psi_m(p, 0)$, which serves as input to the differentiable rasterizer for rendering. In the subtle motion learning, the framework incorporates two tri-plane modules. The first module, with parameters obtained from the static spatial reconstruction stage and kept frozen, provides stable spatial structure information $\delta_r(p, 0)$. The second module is trained during subtle motion learning to capture subtle temporal variations $\delta_r(p, t)$ between adjacent frames. Through a residual connection structure, the model effectively learns and represents subtle motion information $\phi_e(p, t)$. Then a phase-based motion magnification block is used to extract and amplify the phase associated with subtle motions, enabling a more intuitive visualization and interpretation of subtle motion embedding $\phi_m(p, t)$. The Gaussian decoder outputs the subtle motion representation $\psi_m(p, t)$. We adopt the rasterizer of 3DGS [36] to render images $I_m(p, t)$. The loss $\mathcal{L}$ between the rendered images $I_m(p, t)$ and ground truth images $I_{GT}(p, t)$ is utilized for backpropagation to optimize the parameters of the tri-plane and decoder.

A. Preliminaries

In 3D Gaussian Splatting, a scene is explicitly represented as a set of 3D Gaussian points, which model the spatial structure and enable high-quality, real-time rendering from images captured at multiple viewpoints. 3D Gaussian point G is parameterized by a set of attributes: spatial position P, rotation quaternion Q, axis-aligned scaling S, opacity O, and spherical harmonic (SH) coefficients C for encoding anisotropic reflectance properties. Each Gaussian point G_i is defined as

$$G(x) = \exp\left(-\frac{1}{2}(x)^T \Sigma^{-1}(x)\right) \tag{1}$$

The shape of the Gaussian is governed by its covariance matrix Σ, which is decomposed as:

$$\Sigma = RSS^T R^T \tag{2}$$

where R and S represent the rotation and scaling matrices, respectively. S is a diagonal matrix, corresponding to the axis-aligned scaling. R is a rotation matrix derived from the learnable quaternion Q.

During rendering, all 3D Gaussian splats in the scene are first projected onto the 2D image plane, with their colors computed from the associated spherical harmonic (SH) parameters C. Then, for each 16×16 pixel tile of the final image, the projected Gaussians intersecting the tile are sorted by depth. For each pixel within the tile, the final color is computed via alpha compositing, where the contributing Gaussians are blended in a front-to-back order based on their depth. Specifically, the color is accumulated as:

$$C = \sum_{i \in N_{\text{cov}}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{3}$$

Here, N_{cov} denotes the set of Gaussians covering the pixel, c_i is the color of the i-th Gaussian, and α_i is its opacity scaled by the value of the projected 2D Gaussian density at the pixel location.

In addition to the formulation and rendering of Gaussian primitives, 3DGS relies on effective regulation of primitive quantity to achieve a balance between rendering fidelity and computational overhead. Initially, a sparse point cloud is generated by Structure-from-Motion (SfM) [48]. However, this initial set may be insufficient for high-fidelity reconstruction. To address this, 3DGS employs an adaptive strategy that monitors the view-space gradient of each primitive's position. This gradient information is used to assess whether a region is underrepresented or excessively represented. Based on this evaluation, the framework either duplicates or subdivides selected primitives to enrich scene representation. To further enhance training stability and avoid the accumulation of visual artifacts, all opacity values are periodically reinitialized. This adaptive refinement allows 3DGS to begin with a minimal number of Gaussians and dynamically grow the set during optimization, eliminating the need for dense point clouds typically required by earlier differentiable rendering pipelines.

During the training process, 3DGS performs iterative optimization by rendering images and minimizing the discrepancy between the synthesized and ground truth views. Due to the inherent ambiguity in projecting 3D structures into 2D observations, optimization must be capable of dynamically adjusting the scene geometry by adding, removing, or repositioning Gaussian primitives. The precision of covariance parameters in the 3D Gaussians is essential for compact yet expressive scene representations, particularly in modeling large homogeneous regions using anisotropic Gaussians.

B. Space Reconstruction and Subtle Motion Learning

Recent works increasingly adopt tri-plane optimization for both static and dynamic 3D Gaussian Splatting (3DGS) [29],

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2026.3664849

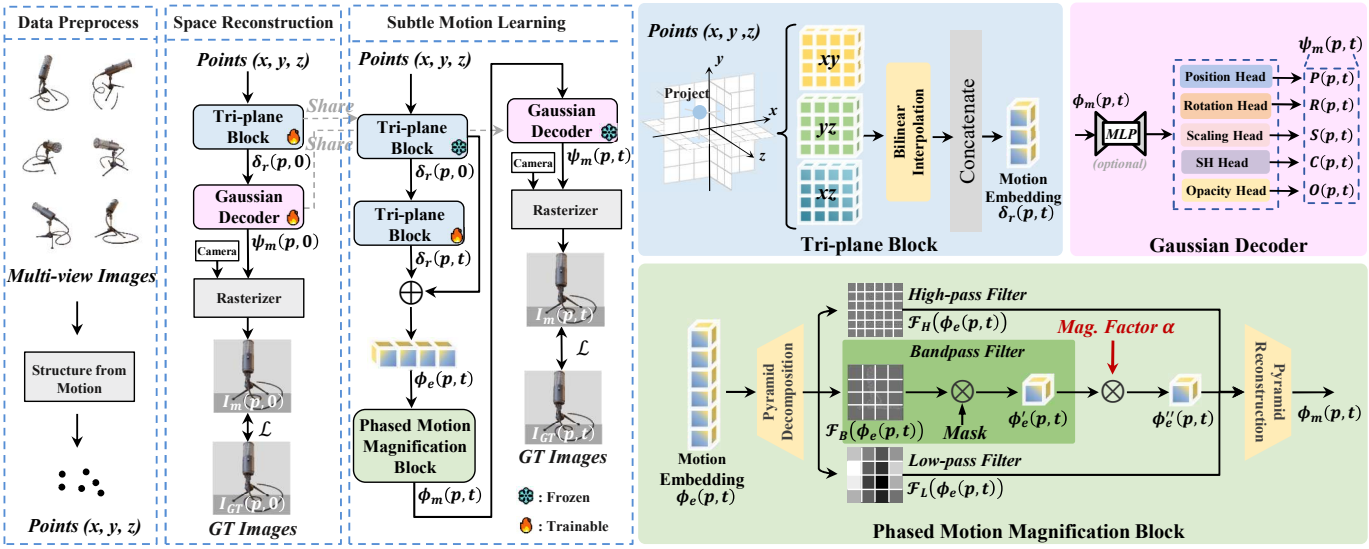


Fig. 2. **Overall pipeline of the proposed TG4MM.** In Space Reconstruction, these 3D points are projected onto three orthogonal planes (xy, yz, xz) via a trainable Tri-plane Block, followed by bilinear interpolation and concatenation to form a motion embedding. This embedding is decoded by a trainable Gaussian Decoder to produce dynamic parameters (position, rotation, scaling, SH, and opacity), which are rasterized into an image. Concurrently, Subtle Motion Learning processes the same 3D points using both a pre-trained and trainable Tri-plane Block. The combined motion embedding enters a Phased Motion Magnification Block, where steerable pyramid decomposition isolates high-frequency motion components. These components are scaled by a magnification factor α to amplify subtle movements before reconstruction.

[42], [43], [49], [50]. As a specific case of the more general K-plane representation, tri-plane modeling encodes spatial features across three orthogonal 2D planes to efficiently represent 3D structures. Our space reconstruction block is built upon this tri-plane representation and incorporates a Gaussian-based decoder tailored for 3DGS rendering. As a spatial representation paradigm, K-Plane [41] provides an efficient, interpretable, and cross-dimensionally unified representation method for high-dimensional spaces such as 3D static scenes, 4D dynamic scenes, and multi-appearance scenes through planar factorization and explicit feature modeling. Its core idea is to decompose complex high-dimensional spaces into combinations of low-dimensional planes, capture spatial structures and dynamic changes through feature interactions between planes, and enhance model performance through regularization and multi-scale design.

We adopt a tri-plane factorization scheme to represent the 3D volumetric features of a scene. Specifically, three orthogonal 2D spatial planes are defined as P_{xy}, P_{xz}, and P_{yz}. Each plane is represented as a feature tensor $P_k \in \mathbb{R}^{H \times W \times C}$, where $k \in \{xy, xz, yz\}$, H and W denote the spatial resolution of the planes, and C denotes the dimensionality of the planes.

Given a 3D point $p = (x, y, z)$, we first normalize it to the coordinate range $[0, 1)$, and then project it onto each of the three canonical planes using projection functions $\pi_k(\cdot)$, where $k \in \{xy, xz, yz\}$. The projected feature is obtained by bilinear interpolation $\psi(\cdot)$ on the corresponding 2D grid as follows:

$$f(p)_k = \psi\left(P_k, \pi_k(p)\right), \tag{4}$$

where $f(p)_k \in \mathbb{R}^C$ represents the interpolated feature vector from the k-th plane.

To generate the final spatial embedding, the features from all planes are combined using concatenation:

$$\delta_r(p, 0) = \text{concat}_{k \in \{xy, xz, yz\}}\left(f(p)_k\right) \tag{5}$$

For spatial reconstruction, the input feature $\delta_r(p, 0)$ is decoded by a Gaussian decoder to predict a set of multidimensional attributes: spatial position P, rotation R, scaling S, opacity O, and spherical harmonic (SH) coefficients C. These attributes are collectively denoted as $\psi_m(p, 0)$. Subsequently, $\psi_m(p, 0)$ and the camera parameters are fed into a differentiable rasterization pipeline to produce the rendered image $I_m(p, 0)$. The rendered output is then supervised using the ground-truth image $I_{\text{GT}}(p, 0)$ through a reconstruction loss. Gradients derived from this loss are utilized to update the parameters of both the tri-plane representation and the Gaussian decoder. Details of the Gaussian decoder architecture and loss formulation will be discussed in the following subsections. To model subtle motion over time, we first extract the spatial representation $\delta_r(p, 0)$ from a frozen tri-plane encoder, which shares parameters with the spatial reconstruction module. Then, a motion-specific tri-plane encoder is employed to learn temporal variations, producing $\delta_r(p, t)$. These two feature embeddings are combined via residual connection to form a motion-aware embedding:

$$\phi_e(p, t) = \delta_r(p, 0) + \delta_r(p, t) \tag{6}$$

This enables the system to model subtle temporal motion that may not be perceptible to the human eye.

To amplify subtle motions for visualization, we introduce a phase-based motion magnification module. Specifically, the embedding $\phi_e(p, t)$ is decomposed into different components using three filters: high-frequency filter $\mathcal{F}_H(\phi_e(p, t))$, low-frequency filter $\mathcal{F}_L(\phi_e(p, t))$, and a band-pass filter $\mathcal{F}_B(\phi_e(p, t))$ centered at the target motion frequency. The

band-passed signal is further masked and scaled by a magnification factor α, yielding the amplified signal $\phi_e''(p,t)$:

$$\phi_e(p,t) = \delta_r(p,0) + \delta_r(p,t) \tag{7}$$

The three components $\mathcal{F}_H(\cdot)$, $\phi_e''(p,t)$, and $\mathcal{F}_L(\cdot)$ are then fused through a pyramid-based reconstruction module to produce the final magnified embedding $\phi_m'(p,t)$.

Following this, similar to spatial reconstruction, the frozen spatial feature $\delta_r(p,0)$ and camera parameters are input into a Gaussian decoder to estimate multi-dimensional attributes, including spatial position P, rotation R, axis-aligned scaling S, opacity O, and spherical harmonic (SH) coefficients C. We denote the set of these attributes as $\psi_m(p,t)$.

Finally, $\psi_m(p,t)$ and the camera parameters are passed into a differentiable rasterization pipeline to render the output image $I_m(p,t)$. A reconstruction loss is computed between $I_m(p,t)$ and the ground-truth image $I_{\mathrm{GT}}(p,t)$, and gradients from this loss are used to update both the tri-plane encoders and the Gaussian decoder.

Details of the Gaussian decoder block, phased motion magnification block and the loss function will be discussed in subsequent subsections.

C. Gaussian Decoder

In this work, the decoder architecture is configured based on parameters required during the rasterization process, aiming to support precise geometry and appearance reconstruction. For novel view synthesis tasks, spherical harmonics (SH) were originally introduced to model anisotropic color variation arising from view-dependent effects [51]–[53]. Building upon these advancements, we incorporate SH representations into our decoder to enhance the expressiveness of appearance modeling. To further improve the quality of upsampling and compositing, we explicitly model opacity as an additional decoding target. Overall, our decoder comprises five parallel heads that respectively predict the spatial position P, rotation quaternion R, axis-aligned scaling S, opacity O, and spherical harmonic (SH) coefficients C, thereby jointly capturing the geometric and radiometric properties of each primitive.

$$\psi_m(p,t) = \mathrm{MLP}(\phi_e(p,t)) = (P,R,S,O,C). \tag{8}$$

In dynamic 3D Gaussian Splatting with triplane decomposition, the decoder is typically composed of two components. The first component is a multi-layer perceptron (MLP) consisting of 8 layers with a width of 256 [29], following the standard configuration of NeRF [37]. The second component is a multi-head decoder designed to match the shape and requirements of rasterization parameters, ensuring accurate rendering. All decoder parameters are jointly optimized via backpropagation. Through empirical evaluation, we observed that retaining only the second component—namely, a multi-head decoder composed of a GELU activation followed by a linear projection—yields the best performance in our setting. This architectural simplification was further validated through ablation studies shown in Tab. VI, demonstrating both accuracy and efficiency benefits.

D. Phased Motion Magnification

Phase-based motion magnification is a subclass of Eulerian motion magnification, and it follows the Eulerian principle of analyzing pixel intensity variations over time at fixed spatial positions.

Let $I(p,t)$ represent the pixel intensity at spatial location p and time t. Assuming the initial frame satisfies $I(p,0) = f(p)$ and subsequent frames are modeled as $I(p,t) = f(p + \delta_r(p,t))$, where $\delta_r(p,t)$ denotes a displacement field over time. The objective of video motion magnification is to generate a motion-magnified output $I_m(p,t)$ under a given magnification factor α. The target formulation is defined as:

$$I_m(p,t) = f\left(p + (1+\alpha)\delta_r(p,t)\right) \tag{9}$$

Assuming the motion is small, the first-order Taylor expansion of the original intensity $I(p,t)$ yields an approximation:

$$I(p,t) \approx f(p) + \delta_r(p,t)\frac{\partial f(p)}{\partial p} \tag{10}$$

By applying the same first-order Taylor expansion to the target definition in Eq. (9), we obtain the magnified intensity as:

$$I_m(p,t) \approx f(p) + (1+\alpha)\delta_r(p,t)\frac{\partial f(p)}{\partial p} \tag{11}$$

We adopt a phase-based approach using the complex steerable pyramid, which enables both motion estimation and reconstruction. To illustrate the principle, we consider a 1D signal undergoing small temporal displacement $\delta_r(t)$, represented as:

$$f(p + \delta_r(t)) = \sum_{\omega=-\infty}^{\infty} A_\omega e^{i\omega(p+\delta(t))} \tag{12}$$

Here, ω denotes the spatial frequency. The sinusoidal component $e^{i\omega(p+\delta_r(t))}$ encodes motion information in its phase term. To extract this motion, we apply a temporal band-pass filter to $\phi_m'(p,t)$, resulting in:

$$\mathcal{M}_\omega(p,t) = e^{i\omega\delta_r(t)} \tag{13}$$

We then multiply this filtered component with the base phase signal $\mathcal{M}_\omega(\cdot)$, and combine across frequencies to produce the magnified signal

$$\phi_e''(p,t) = \alpha \cdot \phi_e'(p,t) \tag{14}$$

where α is the magnification factor. This operation amplifies small motions and makes them visible in the synthesized sequence.

In summary, as illustrated in Fig. 2, we design a Phased Motion Magnification Block that leverages steerable pyramid decomposition to enhance subtle temporal motion. By isolating mid-frequency components and applying spatial masking, the target motion is extracted and magnified as:

$$\phi_m(p,t) = \alpha \cdot \mathcal{M}_\omega(\phi_e(p,t)) \tag{15}$$

where $\mathcal{M}_\omega(p,t)$ represents the band-pass filtered and masked motion embedding. The magnified signal is then reconstructed via inverse pyramid synthesis.

Fig. 3. **Visualization examples on synthetic dataset.** Space-time slices comparison between the proposed TG4MM and the existing 3DMM [23] method on the scene from the Blender dataset. Motion magnification results are shown under amplification factors of 5, 10, 20, 50, and 100.

E. Loss Function

Consistent with 3D Gaussian Splatting [36] and prior dynamic 3D reconstruction methods [29], [54], [55], we adopt the pixel-wise loss to supervise the training process. Given the rendered image I_m and the corresponding ground truth image I_{GT}, the reconstruction loss is defined as:

$$\mathcal{L} = \|I_{GT}(p,t) - I_m(p,t)\|_1 \tag{16}$$

IV. Experiments

A. Datasets

The synthetic dataset originally introduced in NeRF [37] was designed for benchmarking quantitative performance in static 3D scene reconstruction. An extended version of this dataset was later developed by synthesizing subtle motions into each scene using Blender. In this paper, we utilize the extended dataset [23], where each scene is rendered at 30 frames per second (FPS), and the subtle motions follow periodic patterns at either 3 Hz or 5 Hz. This configuration enables controlled evaluation of model performance in scenarios characterized by subtle motions over time. The real-world dataset is originally

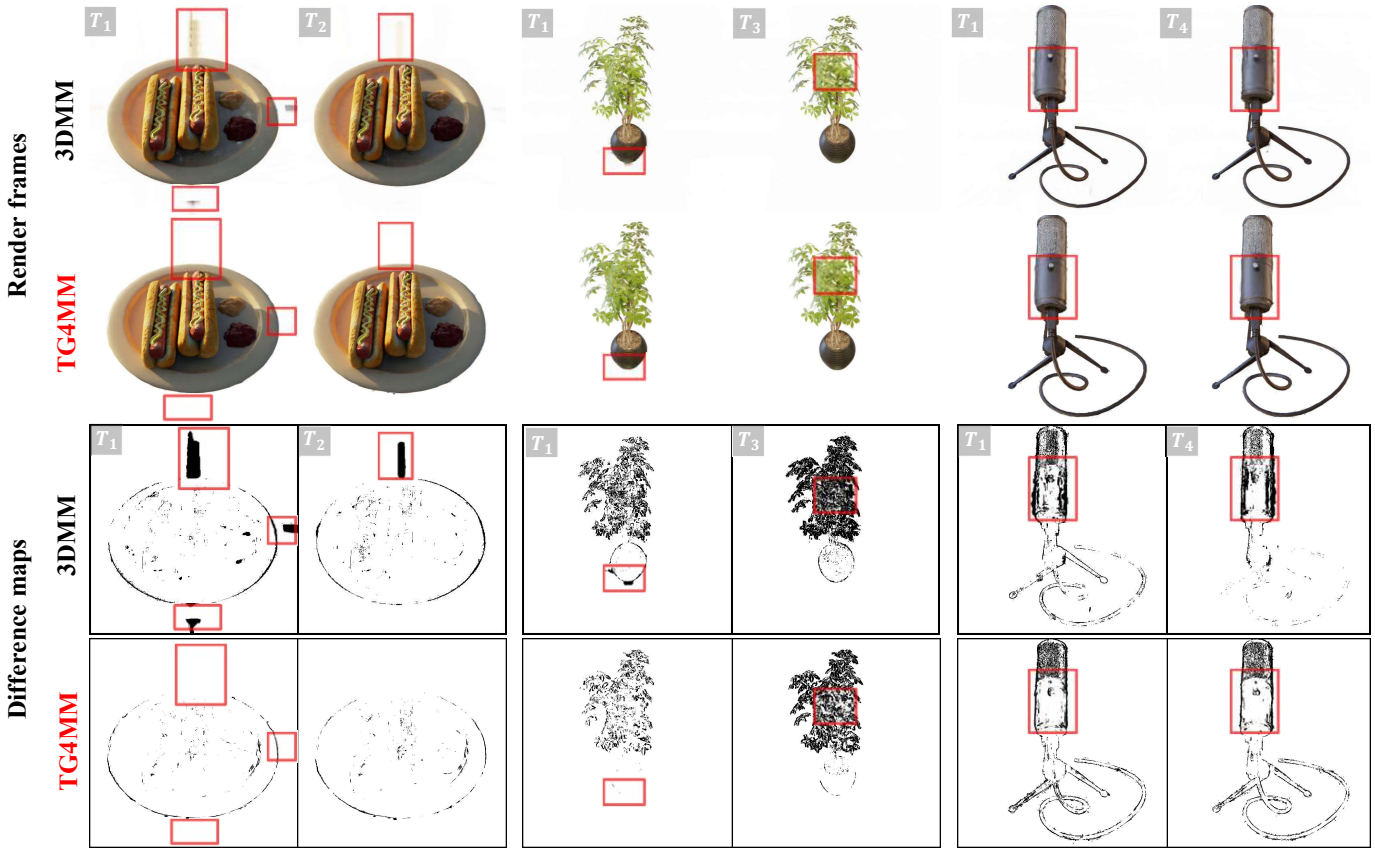


Fig. 4. **Visualization of render frames and difference maps.** The figure illustrates per-pixel absolute differences between consecutive frames I_t and I_{t+1} over one motion cycle. The top rows show results from the baseline method 3DMM [23], while the bottom rows present results from our proposed TG4MM. 3DMM exhibits pronounced incorrect magnification and artifacts in the regions highlighted by red boxes, resulting in significant black artifacts in the temporal difference maps. In contrast, our TG4MM method produces correct rendering outputs with substantially reduced incorrect magnification and artifacts in the corresponding temporal difference analysis.

derived from HumanNeRF [56]. 3DMM [23] extracts a short-time sequence on a single-camera setup from it, within the sequence, the human is relatively static but still has subtle body movements. We follow 3DMM and use the same dataset for our real-world experiments.

B. Implementation Details

Network Architecture and Training. For both the synthetic dataset and the real-world dataset, we adopted the same TG4MM architecture and training configurations, with all experiments implemented on an NVIDIA A800 GPU using the PyTorch framework [57]. For the TG4MM architecture, tri-plane features were configured with 32 channels and spatial dimensions of 512×512. The Gaussian decoder incorporated an MLP network with a depth of one layer and a width of 256 neurons per layer. The training pipeline operated in two sequential phases: an initial spatial representation learning phase involving 20,000 optimization iterations, followed by a subtle motion modeling phase with 7,000 iterations. Both stages employed the Adam optimizer [58] with a fixed learning rate of 1×10^{-4}. For the synthetic dataset, the final video outputs were rendered at a resolution of 512×512 pixels with a frame rate of 30 FPS, while the rendering resolution of the real-world dataset was 1920×1080 pixels.

TABLE I
SYNTHETIC DATASET AVERAGE RESULTS. TG4MM CONSISTENTLY OUTPERFORMS THE BASELINE ACROSS ALL MAGNIFICATION FACTORS, DEMONSTRATING SUPERIOR RESULTS. BOLD DENOTES THE BETTER PERFORMANCE.

Mag. Factors	3DMM [23]		TG4MM(Ours)	
	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓
5	0.9527	0.0344	**0.9621**	**0.0288**
10	0.9498	0.0353	**0.9582**	**0.0299**
20	0.9428	0.0375	**0.9483**	**0.0350**
50	0.9239	0.0427	**0.9258**	**0.0404**

Temporal Filter and Phase Module. To extract target motion frequencies while minimizing spectral leakage, we employ a standard Hamming-windowed FIR bandpass filter. The window function is applied with a length of $N = 30$, corresponding to a 1-second duration at 30 FPS. Filtering is performed efficiently in the frequency domain via FFT. Specifically, we generate the filter coefficients, compute the frequency response, mask the input phase signal in the frequency domain, and reconstruct the signal via Inverse FFT. For the phase module, we utilize a Complex Steerable Pyramid to separate amplitude and phase. It is decomposed into 8

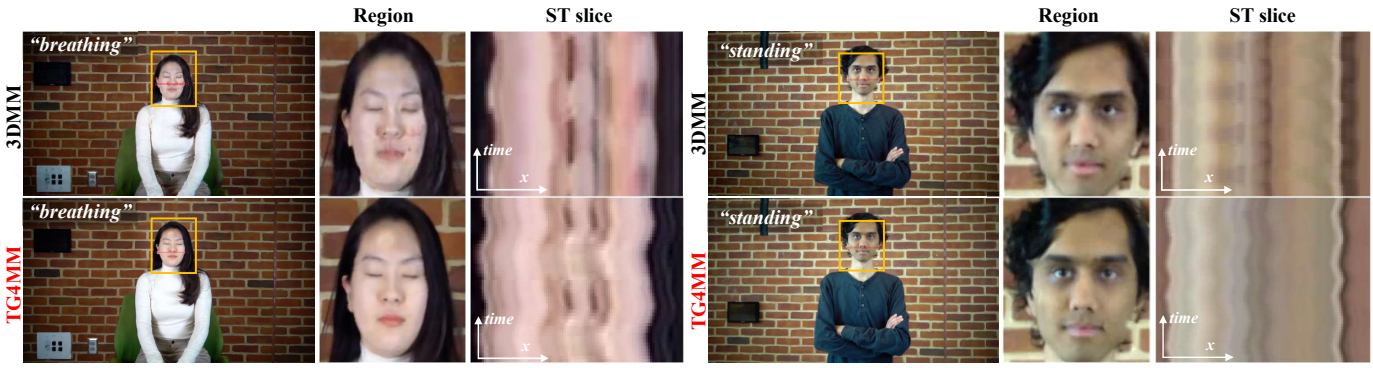


Fig. 5. **Visualization examples on real-world dataset.** Qualitative results on the real-world dataset show that TG4MM outperforms 3DMM in regional details, artifacts and ST slice quality.

orientation sub-bands to precisely capture motion in different directions. We employ an adaptive scale strategy that maximizes pyramid levels based on the input resolution, ensuring coverage from coarse structures to fine textures. To ensure smooth transitions between bands and minimize aliasing, the transition width is set to 0.75.

Frequency Band Settings. The sampling rate is set to 30 Hz. To ensure a fair comparison, we strictly follow the passband configurations of 3DMM [23], which target higher frequencies for rigid vibrations and lower bands for physiological signals. Specifically, scenes like *Mic*, *Chair*, and *Hotdog* use 3.5–6.5 Hz, while *Ficus* and *Ship* use 1.5–4.5 Hz. For real-world inputs, the *Breathing* scene employs 0.8–1.2 Hz and the *Standing* scene uses 0.5–1.5 Hz.

C. Qualitative Evaluation

Synthetic Dataset Results. The proposed TG4MM is evaluated for subtle motion magnification on Blender scenes, with qualitative comparisons presented in Fig. 3. This figure contrasts the baseline 3DMM method [23] (upper rows) against TG4MM (lower rows) across multiple scenarios (*hotdog*, *chair*, *ship*, *ficus*, *mic*) and magnification factors (5×, 10×, 20×, 50×, 100×). For each scene, the reference frame (leftmost column) defines the region of interest (red rectangle), with space-time slices extracted along the horizontal temporal axis (x-axis vs. time). At a 5× magnification factor in the *ficus* scenario, the 3DMM exhibits significant spatial artifacts, including blurred edges and texture distortions along the x-axis, reflecting inherent instability in modeling subtle motions through color and opacity representations. In contrast, TG4MM maintains sharp structural boundaries and effectively suppresses spatial inconsistencies across all test cases. In particular, *ficus* is a high-frequency scene characterized by dense interleaving of foliage and empty regions. In the *ship* scenario, when the magnification factor is set to 10, the ST slices generated by 3DMM show obvious blur and white snow-like artifacts. In contrast, TG4MM not only effectively magnifies subtle motions but also successfully avoids such blur issues. In the *hotdog* scenario, when the magnification factor is set to 100, 3DMM shows tearing. NeRF-based methods use color and opacity modeling, which limits their ability to represent such high-frequency content. In contrast, our approach employs multidimensional attribute modeling to more effectively capture these variations. This performance gap widens at higher amplification scales (e.g., 100×), where TG4MM preserves anatomical integrity while amplifying subpixel motions, whereas 3DMM introduces severe geometric deformations. These results validate TG4MM's superior capability in disentangling and magnifying subtle motions with enhanced spatial-temporal coherence.

TABLE II
COMPLEXITY AND TRAINING TIME COMPARISON OF 3DMM AND TG4MM. COMPLEXITY AND TRAINING TIME ARE LOWER-IS-BETTER METRICS. BOLD DENOTES THE BETTER PERFORMANCE.

Method	Phase 1		Phase 2	
	FLOPs↓	Time↓	FLOPs↓	Time↓
3DMM [23]	13.3G	17m51s	0.05G	3m41s
TG4MM(Ours)	**2.93G**	**5m12s**	**0.03G**	**2m12s**

Moreover, we discuss *Rendered Result and Difference Maps* of synthetic dataset. While the space-time slices provide valuable information on motion continuity and structural stability, their limited spatial coverage restricts the visibility of broader artifact patterns and inaccurately magnified regions. To quantitatively assess these limitations, we propose employing pixel-wise difference maps computed over one complete motion cycle. Specifically, six temporal difference maps are generated from the first seven consecutive frames using the following formulation. For each frame pair at temporal position t, the pixel-wise absolute difference is calculated as:

$$D_t(p) = |I_m(p, t+1) - I_m(p, t)| \tag{17}$$

This temporal differencing approach effectively highlights both localized artifacts and spatially propagated magnification errors across the entire motion sequence. As demonstrated in Fig. 4, the 3DMM [23] exhibits prominent spatiotemporal distortion artifacts in the "*hotdog*" scene. Difference map analysis reveals excessive deformation amplification in critical regions (red bounding boxes) between temporal keyframes T_1 *vs.* T_2, resulting in structural distortions manifesting as anomalous black artifacts. In contrast, TG4MM maintains anatomically consistent magnification characteristics across the temporal

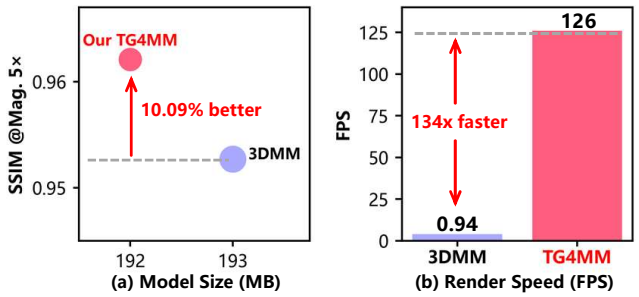


Fig. 6. **Comparison of Model Size and Rendering Speed.** Our proposed TG4MM maintains a compact model size of 192 MB, comparable to 3DMM's 193 MB, while significantly improving rendering speed to 126 FPS, achieving up to 134× speedup over 3DMM. These results validate the superiority of TG4MM in achieving real-time performance without compromising model compactness.

sequence. In the "*ficus*" scene, the 3DMM exhibits severe distortion in leaf vein structures, while TG4MM preserves anatomical details consistently across temporal phases. Similarly, in the "*mic*" scene, 3DMM introduces non-physical elongation artifacts in rigid material regions, whereas TG4MM maintains realistic vibration patterns without geometric aberrations. These comparisons demonstrate TG4MM's robustness in handling both organic and rigid material deformations across diverse scenarios. Quantitative assessment confirms TG4MM's superiority in temporal coherence, achieving lower deformation error energy density compared to conventional approaches while maintaining sub-surface motion amplification fidelity.

Real-world Dataset Results. The qualitative results on the real-world dataset are shown in Fig. 5. This figure compares the baseline method 3DMM (upper row) with our TG4MM (lower row). In the *breathing* scenario, obvious color artifacts appear on the face region marked by yellow boxes generated by 3DMM, and such blurriness and artifacts are also reflected in the ST slice. In contrast, the images generated by TG4MM are clearer and more natural in details like the facial area. In the *standing* scenario, TG4MM achieves better detail restoration for the human region such as finer reconstruction of nostrils and fewer artifacts in the ST slice.

D. Quantitative Evaluation

Synthetic Dataset Results. To quantitatively evaluate the effectiveness of our method, we adopt the 3DMM [23] as the baseline for comparison. Following the same evaluation protocol, we employ the Structural Similarity Index Measure (SSIM) [59] and the Learned Perceptual Image Patch Similarity (LPIPS) [60], with AlexNet [61] as the backbone network, to compare the reconstructed results against the corresponding ground-truth magnified frames. As shown in Tab. I, TG4MM consistently outperforms 3DMM across all magnification factors, increasing SSIM scores by 0.0094 to 0.0019 from 5× to 50× amplification while reducing LPIPS values by 0.0043 to 0.0067. Structural fidelity improvements grow more pronounced at higher magnifications, where TG4MM maintains SSIM advantages above 0.9483 even at 50×, whereas 3DMM drops below 0.924. Perceptual alignment gains remain stable across the amplification spectrum, with TG4MM consistently achieving lower LPIPS below 0.0404 compared to 3DMM's 0.0427 baseline at maximum magnification. Therefore, our method consistently outperforms the baseline across both metrics, demonstrating superior result.

TABLE III
PERFORMANCE COMPARISON OF 3DMM AND TG4MM ON VQA METRICS. MD-VQA AND FAST-VQA ARE HIGHER-IS-BETTER METRICS. BOLD DENOTES THE BETTER PERFORMANCE.

Scenario	MD-VQA↑		Fast-VQA↑	
	3DMM [23]	TG4MM	3DMM [23]	TG4MM
breathing	79.38	**81.04**	0.1922	**0.2089**
standing	74.13	**74.68**	0.1291	**0.1487**

TABLE IV
ABLATIONS ON SPARSE POINT CLOUDS INITIALIZATION. THE RESULTS SHOW THAT OUR SFM OUTPERFORMS VGGT IN SSIM AND LPIPS ACROSS MOTION MAGNIFICATION FACTORS.

Mag. Factors	VGGT [62]		SfM(Ours)	
	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓
5	0.9556	0.0761	**0.9621**	**0.0288**
10	0.9564	0.0696	**0.9582**	**0.0299**
20	0.9403	0.0435	**0.9483**	**0.0350**
50	0.9156	0.0760	**0.9258**	**0.0404**

Real-world Dataset Results. We adopted no-reference metrics including MD-VQA [63] and Fast-VQA [64]–[66] for quantitative evaluation. MD-VQA assesses video quality from three dimensions, namely semantic, distortion, and motion. Fast-VQA assesses video quality through the coordination of local and global aspects. The results are shown in Tab. III, where TG4MM outperforms 3DMM in both MD-VQA and Fast-VQA metrics. We use these no-reference metrics because the reference-based metrics SSIM and LPIPS can only be used when ground truth exists. However, ground truth is only available for the synthetic dataset. Therefore, 3DMM [23] relies on qualitative analysis for the real-world dataset. Considering the lack of a reference standard in real-world scenarios, we use MD-VQA and Fast-VQA for quantitative evaluation.

Model Complexity Analysis. *(1) Model Size.* Fig. 6 presents a comparison of model size between our TG4MM and the only existing method 3DMM [23]. TG4MM has a model size of 192 MB, which is comparable to the 193 MB of 3DMM. *(2) Training time.* Table II presents the computational complexity measured by training time and FLOPs. Both FLOPs and time are low-is-better metrics. Compared with 3DMM, TG4MM reduces the FLOPs in Phase 1 space reconstruction from 13.3G to 2.93G and shortens the training time from 17m'51s to 5m'12s. In Phase 2 subtle motion learning, the FLOPs decreases from 0.05G to 0.03G, and the training time is reduced from 3m'41s to 2m'12s, demonstrating the obvious advantage of our method in computational efficiency. *(3) Rendering Speed.* TG4MM achieves a rendering speed of 126 FPS, significantly outperforming the 0.94 FPS of 3DMM. These results demonstrate the effectiveness of TG4MM in

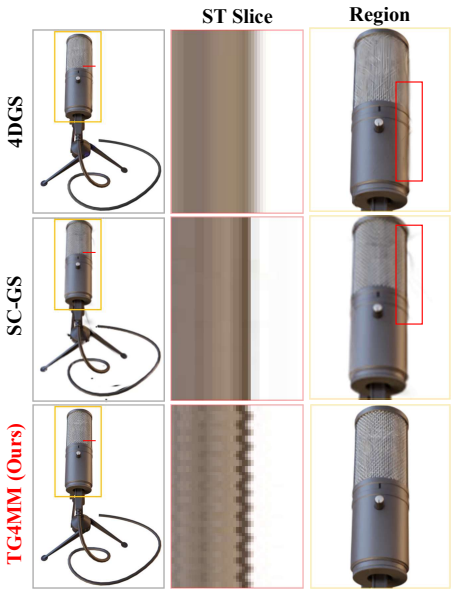


Fig. 7. **Qualitative comparison with Dynamic 3DGS.** 4DGS [29] and SC-GS [67] tend to over-smooth subtle motions, resulting in blurred edges. In contrast, TG4MM effectively captures and magnifies subtle motions.

TABLE V
QUANTITATIVE COMPARISON WITH DYNAMIC 3DGS FRAMEWORKS. TG4MM SIGNIFICANTLY OUTPERFORMS SOTA DYNAMIC METHODS IN RECONSTRUCTION QUALITY FOR SUBTLE MOTION SCENES.

Method	SSIM↑	LPIPS↓
4DGS [29]	0.9698	0.0220
SC-GS [67]	0.9517	0.0712
TG4MM (Ours)	**0.9734**	**0.0095**

TABLE VI
ABLATION STUDY ON DECODER DESIGN. THE SIMPLIFIED DESIGN USING ONLY THE DECODER HEAD ACHIEVES BETTER PERFORMANCE IN CAPTURING SUBTLE MOTIONS. BOLD DENOTES THE BETTER PERFORMANCE.

Mag. Factors	MLP + Decoder		Decoder Only	
	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓
5	**0.9622**	0.0289	0.9621	**0.0288**
10	0.9579	0.0300	**0.9582**	**0.0299**
20	0.9478	0.0356	**0.9483**	**0.0350**
50	0.9230	0.0440	**0.9258**	**0.0404**

delivering high efficiency while maintaining a compact model size.

E. Comparison with Dynamic 3DGS Frameworks

We further compare TG4MM with state-of-the-art dynamic 3DGS frameworks, specifically 4DGS [29] and SC-GS [67]. These methods prioritize large-scale motion reconstruction. As illustrated in Fig. 7, 4DGS and SC-GS fail to capture subtle motions. The ST slices appear smoothed and lack characteristic vibrational patterns. Standard dynamic 3DGS optimization employs temporal smoothness priors to ensure consistency in large-scale dynamics. This mechanism treats subtle high-frequency variations as noise and suppresses them. In contrast, TG4MM effectively preserves and magnifies these imperceptible dynamics. Tab. V confirms that our method yields higher SSIM and lower LPIPS scores. This demonstrates that a standard dynamic 3DGS backbone is insufficient. Our Motion-Space Decoupled Tri-plane and Residual Learning architecture separates static geometry from dynamic residuals. This design effectively prevents the suppression of subtle motion signals.

F. Ablation Studies

Point Initialization Ablation. In the data preprocess of Figure 2 , we use Structure from Motion (SfM) module for point cloud initialization. Recently, VGGT [62] has been applied to point cloud initialization. To verify the effectiveness of the point initialization module in our method, we conducted experiments to compare the performance of our module with VGGT initialization. The results in Table IV demonstrate that our SfM initialization outperforms VGGT initialization. VGGT excellent performance in 3D reconstruction is attributed to its training on large datasets [62], [68], [69], enabling it to provide acceptable baseline performance. However, due to the limited number of samples in our dataset, even with fine-tuning, its subtle motion modeling capability is inferior to our proposed method. In addition, we analyzed the training progress of different methods for point clouds initialization. The training progress is shown in Fig. 9, where we used PSNR (Peak Signal-to-Noise Ratio) and $\mathcal{L}_1$ Loss for quantitative analysis. The training performance in Fig 9 shows that SfM is more suitable for point clouds than VGGT in our method. These results further validate the robustness of our method in 3D motion magnification.

Residual-Structured Tri-plane Ablation. To evaluate the effectiveness of the proposed residual structure in learning subtle motion, we conducted an ablation study comparing two architectural variants. The ”w/ res.” incorporates a learned motion residual from the subsequent timestamp into the current spatial encoding, generating a scene representation embedded with temporal coherence. The ”w/o res.” eliminates this residual connection, instead learning spatial encodings at two time steps independently through separate triplane representations. As shown in Fig. 8, the proposed residual structure enables accurate modeling of subtle motion, evident in both the rendered sequences and spatio-temporal slices. In contrast, the approach without residual learning fails to capture temporal variations, producing static patterns in the space-time analysis. This comparison confirms the residual structure’s critical role in preserving and amplifying sub-surface motion cues. Under the condition of no residual structure, if we subsequently adopt the phase-based motion magnification module, the motion embeddings remain unchanged, and the final output results will remain static as shown in Fig. 8.

Decoder Ablation. For methods based on tri-plane representations, the design of the decoder plays a crucial role in balancing rendering quality and task performance. In conven-

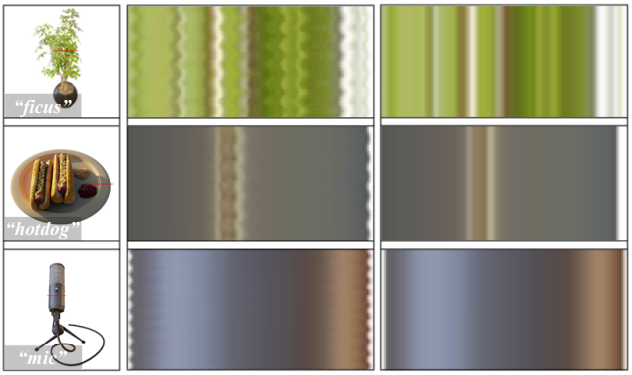


Fig. 8. **Ablation study on residual structure.** Comparison of space-time slices across three representative scenes. The approach without residual structure ("w/o res.") fails to capture temporal variations, producing static patterns in the space-time analysis. This comparison confirms the residual structure's critical role in preserving and magnifying subtle motions.

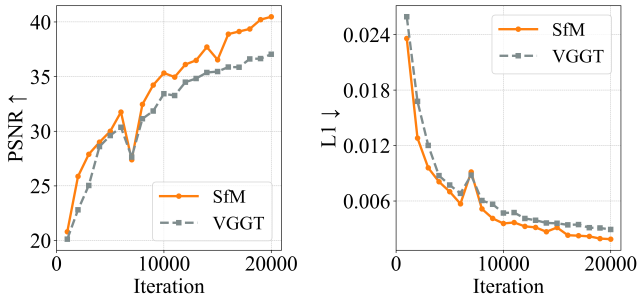


Fig. 9. **Ablation study on point initialization.** Training curves of PSNR (left) and $\mathcal{L}_1$ Loss (right) demonstrate SfM initialization (Ours) outperforms typical point cloud VGGT [62] initialization in both metrics during training.

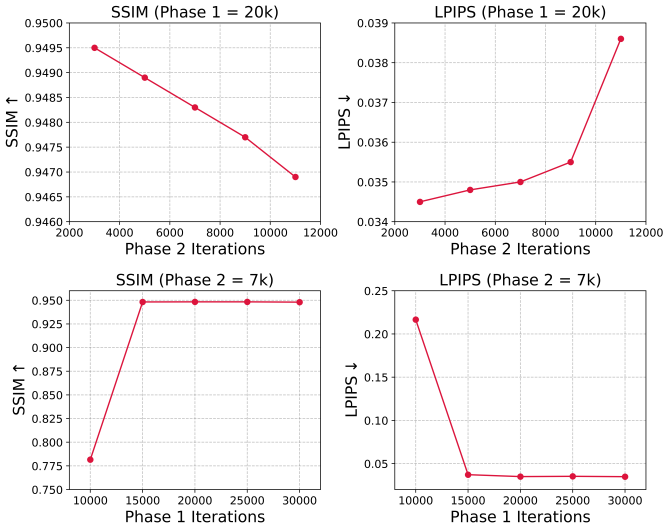


Fig. 10. **Analysis of training iteration counts on rendering quality.** Experiments of how reducing or increasing training iterations in Phase 1 and Phase 2 affects rendering quality, evaluated using SSIM and LPIPS.

tional novel view synthesis tasks, a simple decoder with an activation function followed by a linear layer often suffices for good rendering results [41]. 3D motion magnification, as a downstream task of novel view synthesis, is more sensitive to the decoder architecture. To investigate the impact of decoder complexity on motion magnification performance, we conduct an ablation study comparing two configurations: one with an MLP followed by a decoder head, and another with only the decoder head. As shown in Tab. VI, the simplified design using only the decoder head achieves better performance in capturing subtle motions.

Training Iteration Ablation. To clarify the impact of training iterations on rendering quality, we conducted an experiment: we fixed the iteration count of one phase while adjusting that of the other to observe changes. These results are shown in Fig. 10. In the Phase 2 experiment (with the iteration count of Phase 1 fixed at 20,000), we found that when the iteration count of Phase 2 was reduced from 7,000 to 3,000, the rendering quality actually improved. SSIM increased from 0.9483 to 0.9495, and LPIPS decreased from 0.0350 to 0.0345. This phenomenon occurs because the SSIM and LPIPS metrics tend to reflect static spatial information. However, the core goal of Phase 2 is to learn subtle motions. If we only pursue higher values of these two metrics, the accuracy of motion representation may be compromised. Therefore, for balancing spatial information and subtle motion, 7000 iterations is an appropriate choice. In the Phase 1 experiment (with the iteration count of Phase 2 fixed at 7,000), when the iteration count of Phase 1 was less than 15,000, the spatial information had not yet converged. When the iteration count of Phase 1 exceeded 15,000, the quality gain from further increasing iterations became insignificant, with SSIM stabilizing around 0.948 and LPIPS remaining around 0.035. Therefore, to avoid excessive fine-tuning for each scene, we chose 20,000 iterations to ensure sufficient convergence of spatial information across all scenes.

V. Conclusion

In this paper, we propose TG4MM, a novel 3D motion magnification method based on time-varying Gaussian Splatting. By jointly learning spatial structures and subtle motions, TG4MM effectively amplifies subtle motion in 3D space and supports efficient novel view rendering. Experimental results demonstrate that TG4MM achieves rendering quality comparable to or better than existing approaches across various magnification factors, while achieving a 134× speed-up compared to NeRF-based methods. This work introduces a new paradigm for high-quality and real-time 3D motion magnification that advances the state of the art in the field.

References

[1] D. Guo, K. Li, B. Hu, Y. Zhang, and M. Wang, "Benchmarking micro-action recognition: Dataset, methods, and applications," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 34, no. 7, pp. 6238–6252, 2024.

[2] W. Cai, J. Zhao, R. Yi, M. Yu, F. Duan, Z. Pan, and Y.-J. Liu, "Mfdan: Multi-level flow-driven attention network for micro-expression recognition," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2024.

[3] X.-B. Nguyen, C. N. Duong, X. Li, S. Gauch, H.-S. Seo, and K. Luu, "Micron-bert: Bert-based facial micro-expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1482–1492.

[4] F. Abnousi, G. Kang, J. Giacomini, A. Yeung, S. Zarafshar, N. Vesom, E. Ashley, R. Harrington, and C. Yong, "A novel noninvasive method for remote heart failure monitoring: the eulerian video magnification applications in heart failure study (amplify)," *NPJ digital medicine*, vol. 2, no. 1, p. 80, 2019.

[5] D. Huang, Y. Bi, N. Navab, and Z. Jiang, "Motion magnification in robotic sonography: Enabling pulsation-aware artery segmentation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 6565–6570.

[6] A. Davis, K. Bouman, J. Chen, M. Rubinstein, O. Buyukozturk, F. Durand, and W. Freeman, "Visual vibrometry: Estimating material properties from small motions in video." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 4, pp. 732–745, 2016.

[7] D. Zhang, A. Zhu, X. Gong, Y. Wang, J. Guo, and X. Zhang, "Hybrid eulerian–lagrangian framework for structural full-field vibration quantification and modal shape visualization," *Measurement*, vol. 219, p. 113270, 2023.

[8] M. Eitner, B. Miller, J. Sirohi, and C. Tinney, "Effect of broad-band phase-based motion magnification on modal parameter estimation," *Mechanical Systems and Signal Processing*, vol. 146, p. 106995, 2021.

[9] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson, "Motion magnification," *ACM transactions on graphics (ToG)*, vol. 24, no. 3, pp. 519–526, 2005.

[10] A. C. Le Ngo, A. Johnston, R. C.-W. Phan, and J. See, "Micro-expression motion magnification: Global lagrangian vs. local eulerian approaches," in *IEEE international conference on automatic face & gesture recognition (FG)*, 2018, pp. 650–656.

[11] P. Flotho, M. J. Bhamborae, L. Haab, and D. J. Strauss, "Lagrangian motion magnification revisited: Continuous, magnitude driven motion scaling for psychophysiological experiments," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 3586–3589.

[12] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM transactions on graphics (ToG)*, vol. 31, no. 4, pp. 1–8, 2012.

[13] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, "Phase-based video motion processing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 4, pp. 1–10, 2013.

[14] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, "Riesz pyramids for fast phase-based video magnification," in *IEEE International Conference on Computational Photography (ICCP)*, 2014, pp. 1–10.

[15] S. Takeda, K. Okami, D. Mikami, M. Isogai, and H. Kimata, "Jerk-aware video acceleration magnification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1769–1777.

[16] S. Takeda, Y. Akagi, K. Okami, M. Isogai, and H. Kimata, "Video magnification in the wild using fractional anisotropy in temporal distribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1614–1622.

[17] S. Takeda, K. Niwa, M. Isogawa, S. Shimizu, K. Okami, and Y. Aono, "Bilateral video magnification filter," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 369–17 378.

[18] T.-H. Oh, R. Jaroensri, C. Kim, M. Elgharib, F. Durand, W. T. Freeman, and W. Matusik, "Learning-based video motion magnification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 633–648.

[19] J. Singh, S. Murala, and G. Kosuru, "Lightweight network for video motion magnification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 2041–2050.

[20] J. Singh, S. Murala, and G. S. R. Kosuru, "Multi domain learning for motion magnification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 914–13 923.

[21] F. Wang, D. Guo, K. Li, and M. Wang, "Eulermormer: Robust eulerian motion magnification via dynamic filtering within transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 38, no. 6, 2024, pp. 5345–5353.

[22] F. Wang, D. Guo, K. Li, Z. Zhong, and M. Wang, "Frequency decoupling for motion magnification via multi-level isomorphic architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 984–18 994.

[23] B. Y. Feng, H. Alzayer, M. Rubinstein, W. T. Freeman, and J.-B. Huang, "3d motion magnification: Visualizing subtle motions from time-varying radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 9837–9846.

[24] J. Sun, H. Jiao, G. Li, Z. Zhang, L. Zhao, and W. Xing, "3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 675–20 685.

[25] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," in *International Conference on 3D Vision (3DV)*, 2024, pp. 800–809.

[26] R. Shaw, M. Nazarczuk, J. Song, A. Moreau, S. Catley-Chandar, H. Dhamo, and E. Pérez-Pellitero, "Swings: sliding windows for dynamic 3d gaussian splatting," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 37–54.

[27] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. 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)*, 2024, pp. 20 331–20 341.

[28] Y. Liang, N. Khan, Z. Li, T. 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 (WACV)*, 2025, pp. 2642–2652.

[29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 310–20 320.

[30] M. Verma and S. Raman, "Interest region based motion magnification," in *Image Analysis and Processing (ICIAP)*, 2017, pp. 27–39.

[31] M. Elgharib, M. Hefeeda, F. Durand, and W. T. Freeman, "Video magnification in presence of large motions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4119–4127.

[32] J. F. Kooij and J. C. van Gemert, "Depth-aware motion magnification," in *European Conferenceon Computer Vision (ECCV)*, 2016, pp. 467–482.

[33] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8934–8943.

[34] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," in *European Conferenceon Computer Vision (ECCV)*, 2016, pp. 471–488.

[35] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1164–1172.

[36] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (ToG)*, vol. 42, no. 4, pp. 1–14, 2023.

[37] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[38] Y. Bao, T. Ding, J. Huo, Y. Liu, Y. Li, W. Li, Y. Gao, and J. Luo, "3d gaussian splatting: Survey, technologies, challenges, and opportunities," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2025.

[39] Z. Guo, W. Zhou, L. Li, M. Wang, and H. Li, "Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2024.

[40] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 130–141.

[41] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 479–12 488.

[42] D. Li, S.-S. Huang, Z. Lu, X. Duan, and H. Huang, "St-4dgs: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering," in *ACM SIGGRAPH Conference*, 2024, pp. 1–11.

[43] Z. Xu, S. Peng, H. Lin, G. He, J. Sun, Y. Shen, H. Bao, and X. Zhou, "4k4d: Real-time 4d view synthesis at 4k resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 029–20 040.

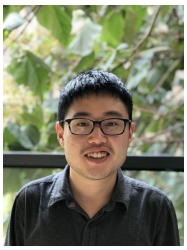
[44] A. Kratimenos, J. Lei, and K. Daniilidis, "Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian

splatting," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 252–269.
[45] Z. Li, Z. Chen, Z. Li, and Y. Xu, "Spacetime gaussian feature splatting for real-time dynamic view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 8508–8520.
[46] Y. Lin, Z. Dai, S. Zhu, and Y. Yao, "Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 21 136–21 145.
[47] J. Li, L. Cheng, Z. Wang, T. Mu, and J. He, "Loopgaussian: creating 3d cinemagraph with multi-view images via eulerian motion field," in *Proceedings of the 32nd ACM International Conference on Multimedia (ACM MM)*, 2024, pp. 476–485.
[48] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM SIGGRAPH*, 2006, pp. 835–846.
[49] J. Yan, R. Peng, L. Tang, and R. 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 (ACM MM)*, 2024, pp. 7871–7880.
[50] R. Hu, X. Wang, Y. Yan, and C. Zhao, "Tgavatar: Reconstructing 3d gaussian avatars with transformer-based tri-plane," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2025.
[51] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5501–5510.
[52] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 5752–5761.
[53] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 333–350.
[54] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia Conference*, 2022, pp. 1–9.
[55] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 318–10 327.
[56] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "HumanNeRF: Free-viewpoint rendering of moving people from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 210–16 220.
[57] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[59] 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 (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.
[60] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.
[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NeuraIPS)*, vol. 25, 2012.
[62] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025, pp. 5294–5306.
[63] Z. Zhang, W. Wu, W. Sun, D. Tu, W. Lu, X. Min, Y. Chen, and G. Zhai, "Md-vqa: Multi-dimensional quality assessment for ugc live videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1746–1755.
[64] H. Wu, C. Chen, L. Liao, J. Hou, W. Sun, Q. Yan, J. Gu, and W. Lin, "Neighbourhood representative sampling for efficient end-to-end video quality assessment," 2022.
[65] H. Wu, C. Chen, J. Hou, L. Liao, A. Wang, W. Sun, Q. Yan, and W. Lin, "Fast-vqa: Efficient end-to-end video quality assessment with fragment sampling," *Proceedings of European Conference of Computer Vision (ECCV)*, 2022.
[66] H. Wu, "Open source deep end-to-end video quality assessment toolbox," 2022. [Online]. Available: http://github.com/timothyhtimothy/fast-vqa
[67] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2024, pp. 4220–4230.
[68] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 697–20 709.
[69] J. Yang, A. Sax, K. J. Liang, M. Henaff, H. Tang, A. Cao, J. Chai, F. Meier, and M. Feiszli, "Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025, pp. 21 924–21 935.

Zheng Zhang received the B.E. degree in Computer Science and Technology from Anhui University of Finance and Economics, China, in 2024. He is currently pursuing the B.S. degree in Computer Science and Technology from Hefei University of Technology. His current research interests include computer vision, 3D vision, neural rendering, and motion magnification.

Jiabao Guo received the Ph.D. degree at the School of Cyber Science and Engineering, Wuhan University. She is currently pursuing postdoctoral research in the School of Computer and Information Science at Hefei University of Technology. Her current research interests include computer vision and multimedia security.

Fei Wang is currently pursuing the Ph.D. degree in Engineering with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His research interests include computer vision and multimodal affective computing. He has published six papers at top international conferences, including CVPR, AAAI, IJCAI, and WWW, and received seven competition awards from ACM MM and IJCAI. He regularly serves as a PC Member for top-tier conferences in multimedia and artificial intelligence, like ICLR, IJCAI, and ACM MM.



Jinyang Huang is a lecturer at the School of Computer Science and Information Engineering, Hefei University of Technology (HFUT) and the Secretary-General of the Anhui Province Key Laboratory of Affective Computing and Advanced Intelligence Machine (led by Prof. Meng Wang (IEEE Fellow)). He obtained his Ph.D. in Computer Science and Technology from the School of Cyberspace Security, University of Science and Technology of China (USTC) in 2022. He was sponsored by China Scholarship Council (CSC) (from 2020.12.1 to 2021.11.31) for a joint Ph.D. study supervised by Assoc. Prof. Lu Su from Purdue University and Chang Wen Chen from University at Buffalo, USA. His research interests include Multimodal Perception, Human-computer Interaction, Wireless Security, and Signal Processing. In this area, he has published 33 papers in international peer-reviewed journals and conferences, including ToN, TMC, TIFS, TAFFC, THMS, TVT, IOTJ, MobiCom, Infocom, ACM MM, and ECCV. He has served as a TPC member for conferences, including ACM MM, IEEE ICME, and Globecom, and has the honor of becoming ACM MM 2024 Outstanding Reviewers. He is a Guest Editor for Applied science. He is the recipient of the Young Scientist of Anhui Computer Federation and IEEE HITC Distinguished PhD Dissertation Award.

Zhi Liu (S'11-M'14-SM'19) received the Ph.D. degree in informatics in National Institute of Informatics. He is currently an Associate Professor at The University of Electro-Communications. His research interest includes video network transmission and mobile edge computing. He is now an editorial board member of Springer wireless networks and IEEE Open Journal of the Computer Society. He is a senior member of IEEE.

Dan Guo received the B.E. degree in computer science and technology from Yangtze University, China, in 2004, and the Ph.D. degree in system analysis and integration from Huazhong University of Science and Technology, China, in 2010. She is currently a Professor with the School of Computer Science and Information Engineering, Hefei University of Technology, China. Her research interests include computer vision, machine learning, and intelligent multimedia content analysis. She regularly serves as a PC Member and for top-tier conferences and prestigious journals in multimedia and artificial intelligence, like ACM Multimedia, IJCAI, AAAI, CVPR and ECCV. She also serves as a SPC Member for IJCAI 2021.